Comparação entre a modelagem orientada a objeto e a modelagem estruturada relacional

Marcos Leandro Figueiredo[1], Hélio Rubens Soares[1]

[1]Curso de especialização em banco de dados

Centro Universitário do Triângulo – UNITRI – Uberlândia – MG - Brasil

mlfigueiredo2004@yahoo.com.br, hrubens@unitri.edu.br

Abstract. *This article has as objective supplies an embasamento on the process and application of the diagram structured relacional and diagram of classes, more specifically on advantages and disadvantages of each modelling. The modelling Guided to Objects it is affirmed basically in concepts, structures and models based on the real world, such as: objects, subjects, inheritance, relationships, etc., the relacional structrured modelling already bases on concepts of entities and attributes, such as relationships, normalization, keys to avoid the redundancy and inconsistency. Due to that definitions of those modellings are presented, the cases in that are applied and also as it is the transition of a modelling for other.*

Resumo. *Este artigo tem como objetivo fornecer um embasamento sobre o processo e aplicação do diagrama estruturado relacional e diagrama de classes, mais especificamente sobre vantagens e desvantagens de cada modelagem. A modelagem Orientada a Objetos se afirma basicamente em conceitos, estruturas e modelos baseados no mundo real, tais como: objetos, assuntos, herança, relacionamentos, etc., já a modelagem estruturada relacional baseia-se em conceitos de entidades e atributos, tais como relacionamentos, normalização, chaves para evitar a redundância e inconsistência. Devido a isso são apresentadas definições dessas modelagens, os casos em que se aplicam e também como é a transição de uma modelagem para outra.*

1. Introdução

O Diagrama de Classes e o Diagrama Estruturado Relacional (DER) são modelos gráficos utilizados no desenvolvimento de sistemas para a representação de bancos de dados, onde cada modelo tem um método de se relacionar com o ambiente de negócios e de representar os dados, apesar de serem bastante parecidos.

A principal utilidade do Diagrama de Classes é a capacidade que a mesma possui em utilizar conceitos OO (orientado a objetos) onde ilustram alguns conceitos como classes, objetos, interfaces e relacionamentos entre elas [3, 4].

Os conceitos do Diagrama de Classes, que é um dos principais diagrama da UML, surgiram do resultado do desenvolvimento de diversos modelos anteriores e ferramentas de representação que adotaram esse paradigma, que descrevem os tipos de objetos no sistema [3].

O DER envolve a identificação do que é importante para a organização, sua característica e como eles estão relacionados uns aos outros. O DER é uma ferramenta de modelagem relacional usada para definir as informações necessárias ao modelo de entidade-relacionamento para o ambiente de negócios [6, 8].

A base das modelagens conceitual Orientada a Objetos e relacional é um pequeno número de conceitos que precisam ser bem entendidos para a correta representação de sistemas de informação [1, 4].

Este artigo destina-se a analistas e projetistas que estejam utilizando o Diagrama de Classes e o DER para a modelagem de sistemas e que irão servir-se de um banco de dados para armazenar os dados, onde será apresentados alguns conceitos, vantagens e desvantagens de cada metodologia e também uma transição do diagrama de classes para o DER, onde o leitor identificará qual modelo mais adapta ao seu ambiente de negócios. O leitor deve ter bons conhecimentos sobre o modelo estruturado relacional/entidade-relacionamento e algum conhecimento sobre o modelo de classes e objetos utilizado pela UML. O artigo compara algumas das técnicas propostas por diversos autores e adotadas em alguns projetos, ressaltando vantagens e desvantagens de cada modelo.

2. O diagrama de classes na orientação a objetos

O diagrama de classes é uma estrutura lógica estática que consta de um conjunto de elementos do sistema, representados como classes. Em comparação com o DER, um diagrama de classes mostra definições para entidades de softwares e conceitos do mundo real [10].

O diagrama de classes contém tipicamente um conjunto de pacotes, formando uma visão gráfica dos elementos arquitetônicos principais do sistema, nada mais é que um gráfico bidimensional de elementos de modelagem que pode conter tipos, relacionamentos, instâncias, objetos e conexão.

Depois de identificar os objetos necessários para o modelo, defini-se suas propriedades. Estas propriedades podem ser dados (atributos), operações e relacionamentos com suas devidas multiplicidades. Trata-se de uma estrutura lógica estática em uma superfície de duas dimensões, mostrando uma coleção de elementos declarativos do modelo [6]. A seguir será mostrado alguns conceitos para a construção do Diagrama de classes.

2.1. Classes e Objetos

Classe é um grupo de objetos semelhantes que compartilham atributos, comportamentos e relacionamentos semelhantes. Uma classe é uma descrição de um tipo de objeto.

Todos os objetos são exemplares das classes que descrevem as propriedades e comportamentos dos mesmos. A figura 1 mostra a representação de uma classe [2]. Cada objeto possui uma identidade própria e são distinguíveis, sendo que também possui um estado e pode mudar quando recebe uma mensagem, isto é, chamado de evento.

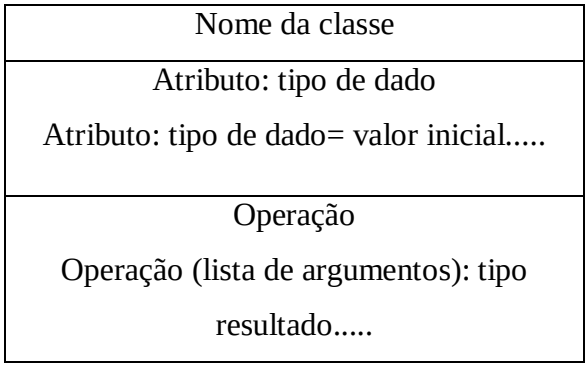


Figura1 – Representação de uma classe.

2.2. Relacionamentos

Relacionamentos são ligações de uma classe a outra, mostrando que entre elas há uma relação importante. Há quatro tipos de relacionamentos [7,9]:

Associação: É a relação que permite especificar que objetos de uma classe se comunicam com objetos de outra classe.

Generalização: Conhecido também como herança ou classificação, permite a criação de elementos especializados em outro. A generalização é um relacionamento entre uma classe geral e uma outra mais especifica (superclasse e subclasse). Existem também alguns tipos de generalizações que variam em sua utilização a partir da situação.

Agregação: Uma agregação é uma forma especial de associação utilizada para mostrar que um tipo de objeto é composto pelo menos em parte de outro em uma relação de todo-parte.

Dependência: Uma classe independente e outra classe dependente, onde uma mudança no elemento independente afetará o elemento dependente.

Multiplicidade: É o número de instâncias de uma classe relacionada com uma instância de outra classe. Para cada associação, há uma multiplicidade em cada direção. Uma multiplicidade é um subconjunto, possivelmente infinito de inteiros não negativos. A figura 2 mostra alguns exemplos de multiplicidade [3].

Multiplicidade	Significado
0..1	zero ou um
1	Somente 1
0...* ou 0...n	Maior ou igual a zero
* ou n	Maior ou igual a zero
1...* ou 1...n	Maior ou igual a 1
1...25	Intervalo de 1 a 25
1...5 , 10...18	Intervalo de 1 até 5 ou 10 até 18

Figura 2 – Exemplos de multiplicidade.

2.3. Atributos e Operações

Para escrever os atributos e operações de uma classe também existem algumas regras a serem seguidas [8]. A figura 3 mostra um exemplo de classe com os atributos e operações.

Os atributos são definidos: Visibilidade NomeDoAtributo: TipoDeExpressão= ValorInicial {Propriedade}.

As operações são definidas: VisibilidadeNomeOperação (parâmetro): ExpressãoTipoRetorno {Propriedade}.

A visibilidade citada pode ter três tipos:

Visibilidade pública (+): é o valor default. Todos têm acesso. O atributo pode ser acessado por outras operações em outras classes;

Visibilidade protegida (#): o atributo é acessado por todas as operações da mesma classe ou do mesmo pacote (grupo de classes que possuem os mesmos comportamentos);

Visibilidade privada (-): o atributo é acessado apenas pelas operações da classe.

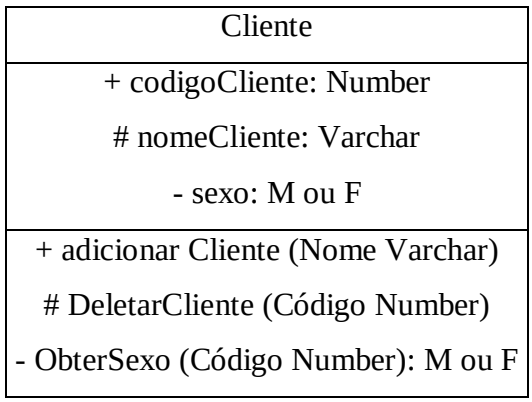


Figura 3- Exemplo de uma classe cliente.

2.4. Aplicação de Diagrama de Classes para modelagem de sistemas

Para mostrar a representação do modelo de dados utilizando o diagrama de classes, este artigo usará um modelo simples, baseado no funcionamento de

uma Imobiliária que tem sob sua responsabilidade a locação de imóveis de terceiros, conforme descrição sucinta, dada a seguir.

Proprietários deixam seus imóveis para serem alugados e é feito um contrato definindo as regras para esta prestação de serviços por parte da Imobiliária.

Alguns imóveis possuem mais de um proprietário com percentuais de propriedade que podem ser diferentes. Quando um imóvel é locado, é feito um contrato de locação que especifica os nomes dos proprietários, o nome do inquilino, os nomes dos fiadores, documentos de identidades das partes, duração do contrato e outros.

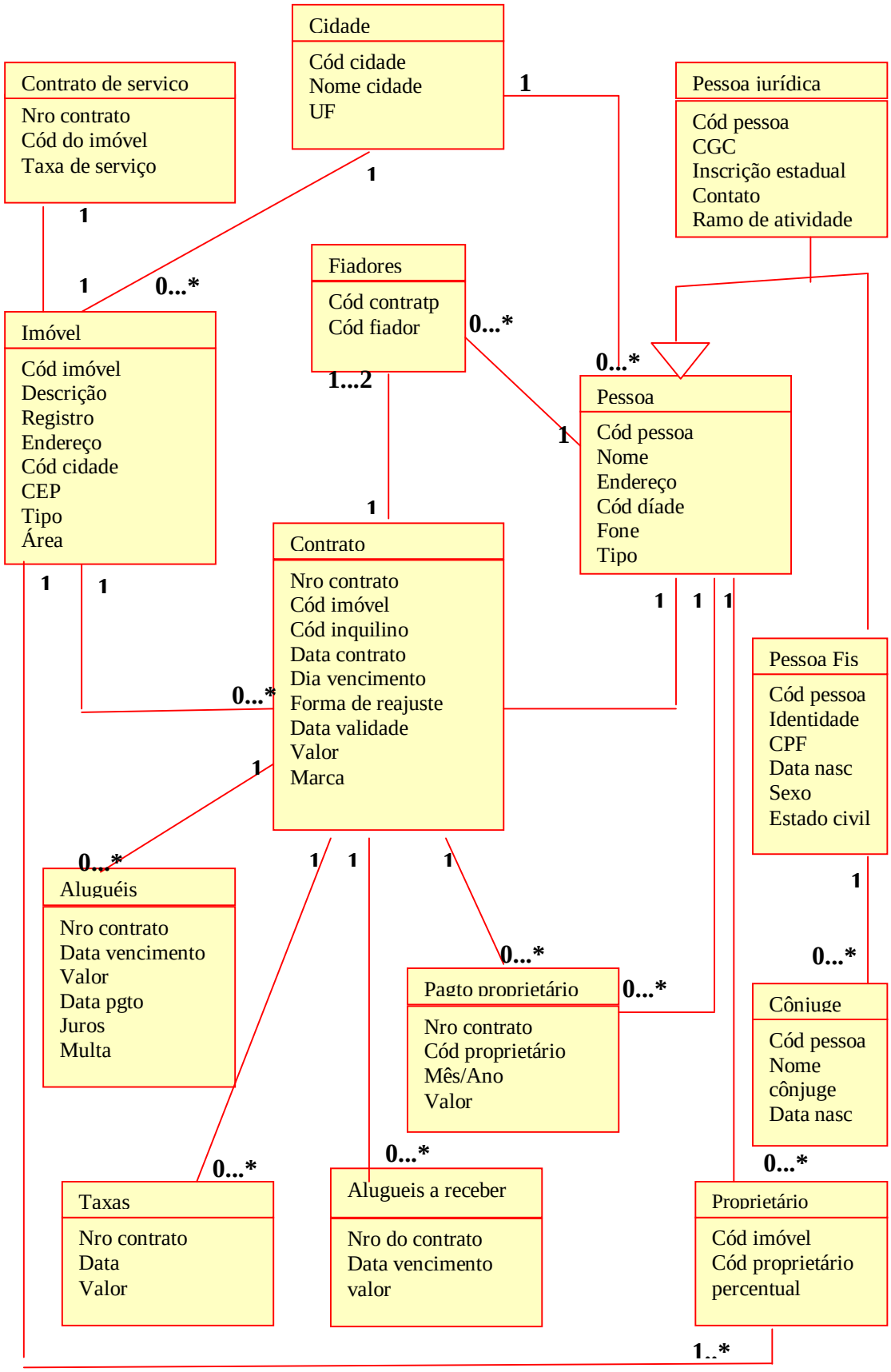
Todos os meses, até o dia do vencimento, o inquilino comparece na Imobiliária para efetuar o pagamento e então recebe o recibo de quitação. Do valor recebido pelo aluguel, a imobiliária desconta a taxa administrativa que é feita em uma listagem de valores a pagar aos proprietários, o valor devido a cada proprietário.

Anualmente, a Imobiliária deve fornecer aos proprietários uma declaração do que eles receberam, relativo a cada imóvel para fins de declaração de imposto de renda.

Com estas informações, foram criadas as Classes: Imóvel, Cidade, Proprietário, Pessoa, Ramo atividade, Contrato, Contrato serviço, Pagamento do Proprietário, Fiadores, Cônjuge, Taxas administração, Alugueis a Receber, Alugueis Recebidos e seus respectivos atributos e operações.

Com esta relação de atividades, serão feitos os diagramas necessários do sistema onde estes foram criados e modelados com auxilio da ferramenta Rational Rose.

Na figura 4 é apresentado o Diagrama de Classes, que será composto por quinze classes que se interagem entre si. São apresentados neste diagrama seus relacionamentos, por exemplo, ligações e herança, seus atributos e se necessário podem se colocados alguns serviços necessário à classe. O diagrama de classes dará enfoque a criação de outros diagramas necessários para a modelagem.


Cidade
Cód cidade
Nome cidade
UF
Contrato de servico
Nro contrato
Cód do imóvel
Taxa de serviço
Pessoa jurídica
Cód pessoa
CGC
Inscrição estadual
Contato
Ramo de atividade
Fiadores
Cód contratp
Cód fiador
Imóvel
Cód imóvel
Descrição
Registro
Endereço
Cód cidade
CEP
Tipo
Área
Pessoa
Cód pessoa
Nome
Endereço
Cód díade
Fone
Tipo
Contrato
Nro contrato
Cód imóvel
Cód inquilino
Data contrato
Dia vencimento
Forma de reajuste
Data validade
Valor
Marca
Pessoa Fis
Cód pessoa
Identidade
CPF
Data nasc
Sexo
Estado civil
Aluguéis
Nro contrato
Data vencimento
Valor
Data pgto
Juros
Multa
Pagto proprietário
Nro contrato
Cód proprietário
Mês/Ano
Valor
Cônjuge
Cód pessoa
Nome
cônjuge
Data nasc
Taxas
Nro contrato
Data
Valor
Alugueis a receber
Nro do contrato
Data vencimento
valor
Proprietário
Cód imóvel
Cód proprietário
percentual
1
0...*
1...2
1..*

3- O Diagrama Estruturado Relacional (DER)

O DER tem como objetivo identificar as entidades de importância na organização, as propriedades destas entidades (atributos) e como eles estão relacionados uns aos outros (relacionamentos).

Conseqüentemente, versões diferentes deste modelo estruturado relacional foram usadas para uma variedade de projetos de banco de dados, métodos [11].

O DER é uma ferramenta de modelagem usada para definir as informações necessárias ao modelo de entidade-relacionamento muito utilizado para banco de dados relacional, diferente do diagrama de classes que modela o mundo real.

Seu objetivo é de disponibilizar um meio simplificado de representar a armazenagem de dados complexos e grandes, sendo um dos mais populares métodos de modelagem de dados. O modelo resultante da informação é independente de qualquer armazenamento de dados ou método de acesso [4].

A idéia-chave é acrescentar um estágio intermediário ao projeto lógico de banco de dados. O projetista de banco de dados primeiro identifica as entidades e relacionamentos que são de interesse para a empresa. Nesse estágio, o projetista deve examinar os dados do ponto de vista da empresa como um todo (não a visão de um programador de aplicação específico). Portanto, usa-se chamar a descrição da visão da empresa quanto aos dados, de esquema da empresa.

O esquema da empresa deve ser uma representação pura do mundo real e deve ser independente de considerações sobre armazenamento e eficiência. O projetista de banco de dados primeiro projeta o esquema da empresa e então o traduz a um esquema do usuário para seu sistema de banco de dados.

O DER baseia-se nos seguintes elementos para sua construção [3, 6, 11]:

Entidades - são coisas reais ou abstratas que são relevantes para uma empresa, como clientes, produtos, faturas, etc. A entidade é algo que deve ser guardada no modelo de dados, onde cada entidade tem seus atributos naturais necessários para sua identificação.

Relacionamentos - são conexões entre duas ou mais entidades. Embora as entidades possam ser independentes, os relacionamentos somente podem ser definidos pela combinação entre entidades relacionadas.

Os relacionamentos no DER podem ser representados de formas diferentes para identificar qual tipo de relacionamento, existe entre as entidades.

Normalização e Desnormalização - é o processo de crítica e validação de um modelo de dados para garantir que este esteja limpo e não redundante, ou seja, que esteja correto. A normalização consiste em criticar o modelo frente às chamadas formas normais, que é um conjunto de regras que definem o que é um bom modelo de dados, para isso existem seis formas normais. Já a Desnormalização são violações intencionais (ou não) às formas normais citadas para melhorar o desempenho (velocidade, agilidade, tempo de resposta) do sistema a certas ações.

3.1 Aplicação do DER na modelagem

Para mostrar também a representação do modelo de dados utilizando o DER, este artigo usará um modelo simples baseado em uma empresa de armazenamento de produtos, conforme a descrição detalhada a seguir.

Uma empresa responsável pelo armazenamento de produtos em cais possui diversos armazéns na região. Cada armazém está climática e estruturalmente adaptado para armazenar apenas um tipo de produto (por exemplo: grãos, produtos químicos, minérios, etc).

Cada armazém possui diversos containeres, identificados por número, onde são alocados os produtos. Quando um cliente traz um produto, este produto ganha um código de identificação, seu tipo é determinado e de acordo com o tipo aloca-se este produto em um ou mais containeres em um ou mais armazéns.

Deve-se saber os containeres que estão disponíveis para um dado tipo de produto e onde estão.

Com esta relação de atividades, será feito o diagrama com auxílio da ferramenta Er-Win. Na figura 5 é apresentado o DER, com suas entidades, atributos, tipos de relacionamentos, migrações de chaves, etc.

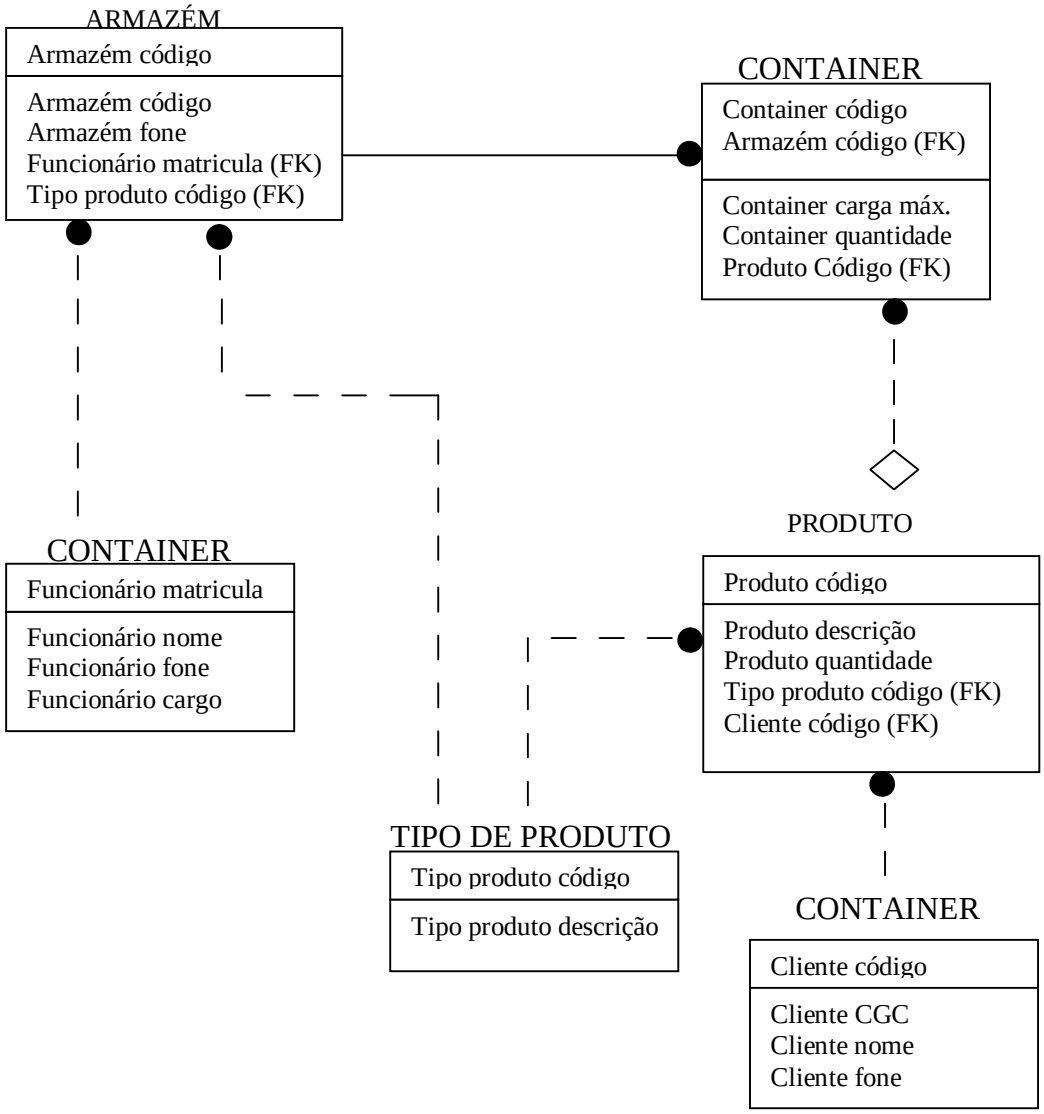


Figura 5 – Diagrama estruturado relacional

4- Vantagens e desvantagens entre diagrama de classe e o DER

A modelagem utilizando orientação a objetos apresenta uma série de vantagens. A principal é uma maior interação com o mundo real, enquanto no modelo estruturado relacional o analista se preocupa com tabelas, relacionamentos

e formas normais e na abordagem OO o foco é na fiel representação do mundo real, segundo os aspectos relevantes para o negócio em questão. Não há, por exemplo, discussões para decidir o que deve ou não ser entidade fraca, relacionamento ternário, chave primária ou candidata, formas de normalização e tipos de relacionamentos [5, 11].

Para a chamada “transição suave” da análise para o projeto e implementação orientada a objetos é necessário que, além de programar em uma linguagem OO (Orientado à objetos) com a modelagem OO, se armazene as informações em um banco de dados OO, por exemplo o Caché. No entanto, a maioria das organizações dispõe apenas de sistemas gerenciadores de bancos de dados relacionais, tais como Oracle, DB2, SQL Server, MySQL, Firebird, dentre outros. Esta grande gama de bancos de dados relacionais é uma vantagem de se trabalhar com a modelagem relacional.

O mapeamento pode ser feito de forma semi-automática por algumas ferramentas, porém, sempre há um custo tanto de desempenho quanto de esforço de programação [10]. Para minimizar estes custos, uma série de heurísticas é apresentada, juntamente com subsídios para que se entenda o cerne do problema.

O termo "descasamento de impedância" é usado para ressaltar as diferenças entre os paradigmas relacionais e orientado a objetos. O paradigma OO é baseado em princípios de programação e engenharia de software bem aceitos ao longo dos anos, tais como acoplamento, coesão e encapsulamento, polimorfismo, herança, mensagens.

O DER baseia-se em princípios matemáticos da teoria dos conjuntos junto com a análise estruturada do ambiente de negócios, este modelo é ainda o mais utilizado para desenvolvimento de sistemas [8].

O diagrama de classes tem várias vantagens para a concepção e a programação de software, como agregação, encapsulamento, agregamento de dados e funções e reutilização. Já o DER tem regras a serem definidas para evitar a redundância e inconsistência dos dados como a normalização, tipos de relacionamentos e outros.

Os dois fundamentos teóricos levam a diferentes pontos fortes e fracos. Além disso, o paradigma OO pressupõe administrar a complexidade dividindo o problema em classes que representam dados e comportamentos associados aos dados, ao passo que o DER propõe formas de se armazenar os dados evitando redundância e as chamadas anomalias de atualização – as formas normais são um bom exemplo de tais técnicas [6]. Com objetivos tão distintos em mente é razoável que se tenha conflitos.

O segredo do bom mapeamento ou escolha de um modelo é entender bem os dois paradigmas e as suas diferenças e fazer escolhas levando em conta a relação custo/benefício de cada decisão. Um exemplo é uma base de um sistema gerenciador de banco de dados puramente relacional, porque isto representa a situação da maioria das empresas. No entanto, é bom lembrar que alguns fabricantes estão estendendo o DER para o chamado modelo objeto-relacional por exemplo, o Oracle 9i já apresenta uma série de construções OO, tais como classes, herança e métodos, tornando a migração certamente mais fácil de um modelo para outro [10].

A adoção das metodologias de desenvolvimento Orientadas a Objetos como um padrão de mercado levou a uma mudança radical na estruturação e organização de sistemas de informação. Contudo, a utilização de modelos e bancos de dados relacionais ainda é uma prática comum e será mantida por um longo período de tempo. Graças à necessidade de se trabalhar com estas bases de dados relacionais para o armazenamento persistente de dados, é comum a adaptação dos modelos de objetos na tentativa de compatibilizá-los com o modelo estruturado relacional [9, 13].

5. Conclusão.

As duas abordagens oferecem interessantes direcionamentos para o desenvolvimento de sistemas, cada uma com um modo de pensamento, simplificando a complexidade do ambiente computacional.

Fundamentalmente, qualquer um que se proponha a compreender uma organização deverá estar atento às questões relacionadas a estratégia empresarial, às funções e aos dados, independente de qual modelagem será utilizada. A formação dos profissionais de negócio não tem sido suficiente nesses aspectos, a falta de compreensão atua como uma força poderosa que leva a organização a apresentar situações indesejáveis, sendo que o problema está deixando de ser o software e o hardware, cada dia mais poderosos [3, 4].

A modelagem é a base teórica para a estruturação da organização e suas respectivas áreas organizacionais. Busca a compreensão dos recursos gerenciados pelo negócio para que se estabeleçam funções e processos derivados de tratamento aos diversos eventos acionadores [8].

Com os estudos de caso citados, foi possível mostrar as vantagens e desvantagens de cada modelagem e como utilizá-las em termos de projeto. Desenvolver um sistema utilizando estes diagramas para modelagem de dados é considerado fácil, mas necessita do domínio do ambiente e do paradigma a ser utilizado.

6. Referências

[1] Roger S., Engenharia de Software – São Paulo, Editora Makron Books, 1995.

[2] Coleman, D. et al. Desenvolvimento Orientado a Objetos: O Método Fusion. Ed. Campos, Rio de Janeiro, 1996.

[3]Costa, R.M. Um método de Engenharia Reversa para a Manutenção de Software.
Dissertação de mestrado, ICMC–USP, São Carlos, 1997.

[4] Furlan, Davi. Modelagem de objetos através da UML. Editora Makron Books, 1998.

[5] Larman, Craig. Utilizando a UML e padrões – Uma introdução à análise e ao projeto orientado a objetos. 1ª Edição. Editora Bookman, 2000.

[6] Coad, Peter & Yordon, Edward. Análise baseada em objetos. 2ª ed. Editora Campus, 1992.

[7] Martin, James & Odell, James. Análise e projeto orientados a objetos. Editora Mackron Books.

[8]Diagrama de classes: http://www.mundooo.com.br\diagrama de classes. Acessado no dia 13 de abril 2004, em espanhol.

[9]Desenvolvimento do diagrama de relacional e diagrama de classes: http://www.rational.com.br - Página oficial da UML, acessado 13 de abril de 2004.

[10]Unified Modeling Language: http://www.rational.com/uml/index.jtmpl, acessado 23 de abril de 2005, em inglês.

[11]Desenvolvimento em banco de dados relacional e orientado a objetos: http://www.dourados.br/apostilas/diagrama estruturado relacional, acessado 15 de agosto de 2004.

[12]Banco de dados e diagrama estruturado relacional: http://www.genesis.ufrj.br/diagramarelacional, acessado 13 de outubro de 2004.

[13]Transições entre modelagem relacional e diagrama de classes: http://www.cefets.br/edu/sinergia/67pc.html, acessado 12 de agosto de 2004.