

UMA ABORDAGEM DINÂMICA SOBRE O MAPEAMENTO OBJETO-RELACIONAL

Daniel Costa Lima¹

Resumo. O Mapeamento Objeto-Relacional é um recurso utilizado para minimizar os problemas da engenharia de software orientada a objetos, que utiliza bancos de dados relacionais. No cenário atual, onde os desenvolvedores convivem com a realidade das tecnologias de orientação a objetos, aliada à utilização de válidas e consistentes bases de dados relacionais, cresce de importância o conhecimento de *frameworks* de mapeamento objeto-relacional ou simplesmente ORM (*“Object-Relational Mapping”*).

Palavras-chave: Mapeamento Objeto-Relacional. Modelo Relacional. Modelo de Objetos. Orientação a Objetos. Triângulo pessoas, processos e tecnologia. Mapeamento da herança.

Abstract. Object-Relational Mapping is a resource used to minimize the problems of software engineering oriented to objects that use relational database. In the current scenario, where the developers work with the reality of technology orientation to objects, in association to the use of valid and consistent relational database, which reflects the importance of framework knowledge for mapping object-relational or simply ORM (Object-Relational Mapping).

Keywords: Object-Relational Mapping. Relational Model. Object Models. Object Orientation. Triangle people, processes and technology. Legacy Mapping.

1 Introdução

Do final dos anos 90 até os dias atuais, as linguagens de programação orientadas a objetos foram difundidas em grande proporção, sendo crescente o desenvolvimento de softwares em linguagens que adotam esta tecnologia, configurando-se como uma importante tendência no campo da engenharia de software. Ao mesmo tempo, siste-

mas gerenciadores de bancos de dados orientados a objetos foram lançados no mercado. Entretanto, tais lançamentos não alcançaram o sucesso esperado em aplicações convencionais, talvez pelo fato de que os bancos relacionais possuem posição largamente consolidada, aliada à grande variedade e versatilidade das opções disponíveis no mercado; isto, sem mencionar-se o transtorno que seria causa-

¹ Especialização em Desenvolvimento de Aplicações World Wide Web. Universidade para o Desenvolvimento do Estado e da Região do Pantanal (UNIDERP), Campo Grande, Brasil. danclima_plus@hotmail.com .

do pelo não-aproveitamento das grandes bases de dados constituídas pelas organizações durante anos de utilização modelos relacionais.

O modelo orientado a objetos veio oferecer um leque de recursos consideráveis, trazendo soluções mais versáteis, fáceis de manter e evoluir. Contudo, o modelo relacional na implementação de bancos de dados é uma realidade ainda presente na maioria dos projetos de sistemas. A teoria relacional de Edgar Frank Codd continua imbatível, como a melhor forma de arranjar, armazenar e recuperar dados. O modelo de armazenamento de dados relacional de Codd preconiza o uso de tabelas, que consistem basicamente em linhas e colunas e que permitem a imposição de restrição de consistências, teoria esta ainda extremamente válida e utilizada em larga escala pelos bancos de dados existentes no mercado.

A orientação a objetos permitiu um aperfeiçoamento do modelo *procedural*, uma vez que *procedures* e *functions* são, na maioria dos casos, difíceis de evoluir e manter, devido à fraca consistência e coesão dos programas. O modelo de objetos veio justamente para aumentar esta coesão. Mesmo assim, a persistência de objetos para posterior recuperação, em virtude de uma necessidade de análise

futura, tornou, nesta situação particular, o modelo de objetos relativamente pobre em relação ao modelo relacional. A solução foi utilizar o melhor das duas tecnologias, permitindo que os objetos fossem responsáveis pelas regras de negócio, enquanto o modelo relacional ficasse responsável em garantir um estado persistente dos objetos, contribuindo para facilitar a busca e a análise das informações, nascendo, desta forma, o Mapeamento Objeto-Relacional, ou OMR (“*Object-Relational Mapping*”).

Passaremos agora a apresentar os principais aspectos da tecnologia do Mapeamento Objeto-Relacional no contexto da Orientação a Objetos, mostrando as principais semelhanças e diferenças existentes entre o Modelo de Entidade e Relacionamento e o Modelo Orientado a Objetos, apontando os principais benefícios desta nova tecnologia, avaliando o comportamento das pessoas e organizações, diante deste novo paradigma, destacando seu impacto no cenário atual e concluindo sobre o futuro das Técnicas Orientadas a Objetos.

2 A Análise orientada a Objetos e as técnicas estruturadas

A Análise Orientada a Objetos tomou força, como tendência, a partir da segunda metade da década de 1990. Abordando o conceito da Análise Baseada em Objetos, Yourdon e Coad dizem o seguinte:

A OOA – Análise Baseada em Objetos – utiliza os conceitos que aprendemos no jardim de infância: objetos e atributos, todos e partes, classes e membros. É difícil explicar por que demoramos tanto a aplicar estes conceitos à análise e especificação de sistemas de informações – talvez porque estivéssemos ocupados demais “seguindo a boiada” durante o auge da análise estruturada para imaginar que havia alternativas (COAD e YOURDON, 1997, p.1, grifos dos autores).

A Orientação a Objetos é um dos maiores avanços em tecnologia de desenvolvimento de software desde o final da década de 80. É uma forma mais natural e intuitiva de se analisar o mundo. Ela nos permite construir sistemas melhores e, além disso, de maneira mais fácil.

As técnicas estruturadas obtiveram grande aceitação, desde que foram lançadas no final dos anos 70. Contudo, à medida que se utilizavam tais técnicas, a decomposição funcional pas-

sou a mostrar-se inadequada em situações de sistemas complexos e principalmente para profissionais iniciantes. Os aperfeiçoamentos introduzidos em 1984, por Stephen M e John F Palmer e, mais tarde, complementados por Stephen Mellor e Paul Ward, para sistemas em *real-time*, ajudaram a análise estruturada a se tornar mais eficiente, mas os sistemas criados com as técnicas estruturadas ainda eram difíceis de serem incrementados com novas funções e as alterações em funções já existentes, muitas vezes, provocando sérios problemas em outras partes do software.

A maior dificuldade do cliente de um software é comunicar ao analista e ao projetista o que ele deseja, de modo claro, completo e preciso. Esta dificuldade é constatada, em muitos casos, apenas quando o sistema está concluído, e não apresenta os requisitos desejados, gerando insatisfação e frustração. O insucesso é tão intenso quanto as suas expectativas. O atendimento aos requisitos dos usuários de software não possui uma relação linear com a satisfação do cliente, aspectos subjetivos, requisitos não expressos e subentendidos podem comprometer o trabalho de desenvolvimento. A linguagem de comunicação entre o cliente e o analista deve ser clara para ambos. A análise baseada em objetos

supera este desafio, utilizando elementos do próprio domínio do cliente para representar o sistema. Um sistema baseado em objetos é a composição de elementos conhecidos do usuário, arranjados de modo a representar o sistema desejado com maior precisão, e com o mínimo de esforço.

3 As pessoas e as mudanças tecnológicas

Uma das melhores formas de se avaliar o processo de desenvolvimento de software é verificar se o triângulo “pessoas-processos-tecnologia” está sendo aplicado, conforme retrata a figura 1.

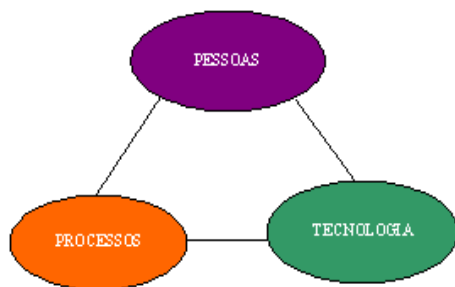


Figura 1 - Triângulo de desenvolvimento de software

Esta teoria não se aplica somente a projetos de software, sendo utilizada em qualquer processo de desenvolvimento na área da tecnologia. A tecnologia é uma das bases do triân-

gulo, pois possui vital importância no processo, representando inovações, criatividade, idéias novas, visando aumentar a eficácia e a eficiência da organização envolvida, através da automação de processos antes realizados manualmente ou com pouca utilização de recursos de informática.

Aplicando-se a tecnologia no contexto do mapeamento objeto-relacional e não se possuindo uma forma padronizada de persistir objetos, soluções caseiras e pouco eficientes foram adotadas, trazendo inúmeras limitações e graves problemas de performance, além de não garantir uma portabilidade multiplataforma entre os bancos de dados existentes no mercado. Surgiram, então, problemas relacionados à complexidade no tratamento das diferenças de impedância entre o modelo de objetos e o relacional.

As mudanças tecnológicas implicam nova postura das pessoas, o que nem sempre é um processo tranquilo, de vez que há uma tendência natural do ser humano em reagir às mudanças, principalmente em processos já consolidados. No contexto do desenvolvimento de software, ocorreu um grande impacto quando a orientação a objetos passou do mundo acadêmico para o mundo dos negócios. Quando este fato ocorreu, as pessoas envolvidas sentiram uma grande necessidade

de evoluírem sua maneira de pensar para abordarem este novo paradigma. Por outro lado a análise estruturada buscou enfatizar a centralização na arquitetura de componentes, através do *Rational Unified Process*.

A grande revolução tecnológica que promoveu uma profunda mudança de comportamento nas pessoas e nos processos utilizados foi o surgimento de *Integrated Development Environment* (IDE) totalmente orientadas a objetos. Sendo precursor desta tendência o *Smaltalk*, na década de 90. De uma forma geral, essas novas IDEs trouxeram uma nova gama de recursos que agilizaram em muito o processo de desenvolvimento, tais como *build* automáticos, integração contínua entre outros. Essas ferramentas abriram novos horizontes para os programadores, dando uma maior liberdade no processo de desenvolvimento de software, provando que as pessoas se adaptam rapidamente ao novo paradigma, a cada nova evolução tecnológica, revendo seus procedimentos e processos.

Um dos aspectos fundamentais para que o triângulo pessoas-processos-tecnologia agregue importante valor às organizações é a compreensão de que a chave para o sucesso está em investir, prioritariamente, nas pessoas e na tecnologia, principalmente em

atividades predominantemente criativas como a engenharia de software. Ainda persiste na política de muitas Organizações a idéia do investimento prioritário nos processos, em detrimento das pessoas e da tecnologia, cren-do-se que, evoluindo o processo, as pessoas e a tecnologia também evoluirão. Um conceito comprovadamente distorcido. Muitas organizações preferem gastar um milhão de reais em certificações a investirem cinco mil reais em treinamento de pessoal.

4 As características do Modelo Relacional e do Modelo de Objetos

O Modelo de Entidade e Relacionamento (MER) tem por base a percepção de que o mundo real é formado por um conjunto de objetos chamados de entidades e pelo conjunto de relacionamentos entre esses objetos. Foi desenvolvido para facilitar o projeto do banco de dados, permitindo a especificação do esquema da Organização que representa e toda a sua estrutura lógica. O MER é um dos modelos com maior capacidade semântica, no que se refere à tentativa de representar o significado dos dados.

Sendo um dos mais importantes instrumentos para representar a modelagem de um sistema estruturado, o

MER, criado por Peter Chen em 1976 mostra o nível de abstração visto pelo usuário do sistema. Esse modelo representa os dados estruturados em tabelas e seus relacionamentos, no universo de um banco de dados, forçando uma consistência restritiva. É um modelo, ao mesmo tempo, conceitual e técnico, sendo também conhecido como Modelo Conceitual de Dados ou, *Conceptual Data Model (CDM)*. Em sua representação geral, demons-

tra entidades de negócios e seus relacionamentos, como descrito na figura 2.

Outro importante instrumento para representar a modelagem de um sistema, neste caso, utilizando a técnica de orientação a objetos é o Diagrama de Classes. Esse diagrama representa classes que definem o comportamento de objetos, suas especificações, associações e dados, conforme se pode observar na figura 3.

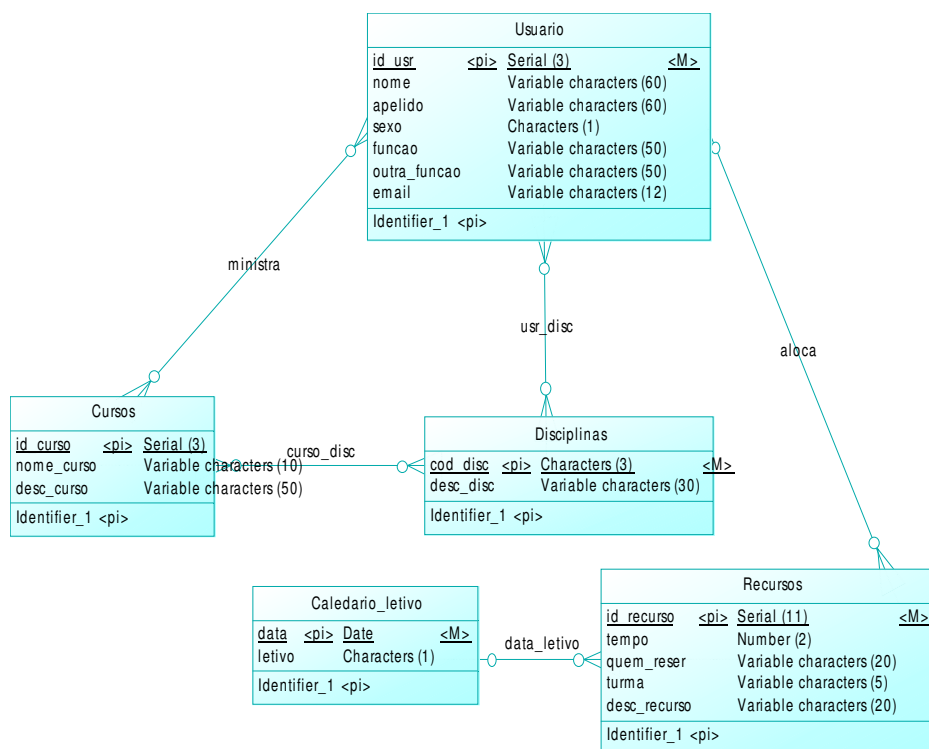


Figura 2 - Diagrama de entidades e relacionamentos de um sistema escolar

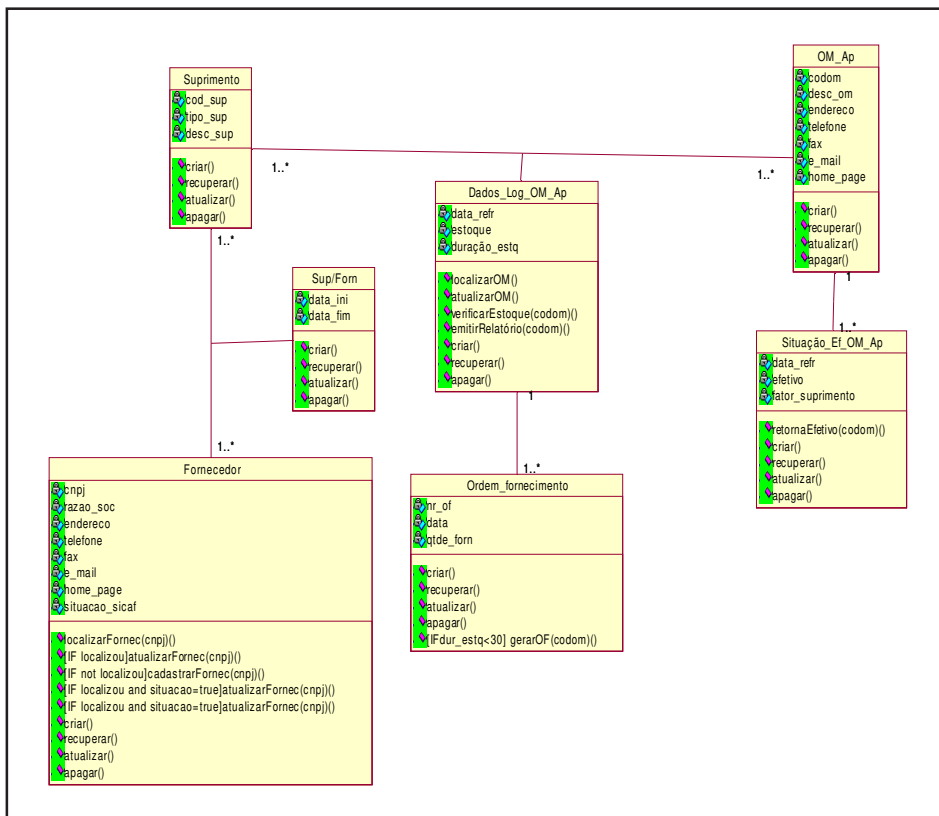


Figura 3 - Diagrama de Classes de um sistema militar de gerenciamento logístico

Posto que é um diagrama técnico-conceitual, como o Modelo de Entidade e Relacionamento, o Diagrama de Classes pode ser utilizado para apresentar ao usuário como os conceitos de negócio serão validados e compreendidos no contexto do sistema, mostrando como esses conceitos serão implementados em código. O objetivo principal de um diagrama de

classes é estabelecer uma idéia do domínio da aplicação, sob um determinado aspecto. As associações entre as classes no modelo de objetos são muito mais focadas no aspecto funcional que nos dados propriamente ditos.

Os conceitos existentes no Modelo de Entidade e Relacionamento são suficientes para representar vários esquemas da base de dados para apli-

cações comerciais. No entanto, com a evolução das tecnologias, novas aplicações de bases de dados surgiram, sendo que estas têm necessidades mais complexas do que as encontradas na maioria das aplicações tradicionais.

O Diagrama de Entidades e Relacionamentos e o Diagrama de Classes, modelos que à primeira vista se prestam a um mesmo objetivo, apresentam diferenças marcantes e também semelhanças, em se tratando de Análise e Projeto de Sistemas. As características peculiares de cada modelo apontam que as diferenças e semelhanças existentes entre ambos decorrem da natureza dos dois modelos, pois, enquanto o modelo relacional é fundamentado em princípios matemáticos comprovados, como a Álgebra Relacional, o modelo de objetos é fundamentado em boas práticas da engenharia de software, também comprovadas. Contudo, temos disponíveis estratégias, também comprovadas, para lidar com essas importantes peculiaridades de cada modelo.

5 Estratégias de Mapeamento da Herança para o Modelo de Dados

A principal diferença entre um Diagrama de Classes e um Diagrama de Entidade e Relacionamentos é que o modelo de tabelas não suporta a he-

rança. O Modelo Relacional representa a estrutura física e lógica dos dados das tuplas, não permitindo o estabelecimento de heranças (YOSHIMA, 2007, p.18).

O Modelo de Objetos possui uma riqueza de detalhes muito maior do que o modelo relacional, sendo muito mais simples compreender a natureza de uma classe, simplesmente observando-se as operações definidas para seus objetos, o que proporciona uma visão mais aprofundada das funcionalidades do sistema.

O Modelo de dados enfoca os dados em si, sem se preocupar com funcionalidades e coesão.

O Mapeamento da Herança foi um dos grandes problemas enfrentados no contexto do Mapeamento Objeto-Relacional, juntamente com as questões de controle transacional.

Para resolver o problema do Mapeamento da Herança para o modelo de dados surgiram diversas estratégias, como o mapeamento de uma tabela por hierarquia, com a criação de uma única tabela que concentrará todos os campos das estruturas das classes. Esta abordagem atende, de forma mais eficiente, estruturas de objetos simples, com poucas classes e atributos, sendo mais eficaz para buscas, uma vez que são dispensados *Joins* com outras tabelas, melhorando

sensivelmente o desempenho.

Outra estratégia adotada para resolver o Mapeamento da Herança é o mapeamento de uma tabela por classe, favorecendo um menor acoplamento e acomodando os dados das classes em tabelas separadas. Esta abordagem desfavorece o aspecto da performance do banco de dados. Sendo, a estrutura das classes e o volume de dados envolvidos muito grande, podendo-se concluir que fatalmente ocorrerão problemas com o sistema, sendo necessários alguns ajustes na estrutura de índices do banco de dados, bem como na estrutura das classes, no sentido de agilizar as consultas.

6 Conclusão

O presente trabalho apresentou uma abordagem sobre o Modelo Relacional e o Modelo Orientado a Objetos, no contexto do Mapeamento Objeto-Relacional, técnica que busca viabilizar o armazenamento de objetos em bancos de dados relacionais. O foco deste trabalho foi descrever aspectos desta nova tecnologia no contexto atual das Organizações, o comportamento das pessoas diante deste novo paradigma e as principais estratégias utilizadas para solucionar o problema do mapeamento da herança.

O principal problema do Mapeamento Objeto-Relacional está ligado à performance, pois a cada execução comandada, ocorrerá uma carga completa do banco de dados para a memória principal e, posteriormente, uma reconstrução completa da base. Tal procedimento pode se tornar extremamente oneroso para grandes bases de dados.

Verificou-se também, através do estudo da teoria do triângulo pessoas-processos-tecnologia, que há necessidade de uma mudança na postura das Organizações, no sentido de investirem, de forma mais efetiva, nas pessoas e na tecnologia e não somente nos processos, visando alcançarem resultados mais efetivos, diante de novos paradigmas do mundo tecnológico, como é o caso do Mapeamento Objeto-Relacional.

Por outro lado, considerando a tendência de crescimento das técnicas orientadas a objetos do cenário atual, o futuro aponta para uma adoção gradativa de bancos de dados orientados a objetos, permitindo a superação dos atuais óbices, que restringem sua utilização a situações muito particulares. Esta nova postura proporcionará a manipulação de estruturas, o gerenciamento de armazenamento, o tratamento de integridade e a persistência dos dados. A implementação de

um Sistema de Gerenciamento de Banco de Dados Orientado a Objetos (SGBDOO) permitirá, ainda, a execução de *Object Query Language* (OQLs) sobre a base de dados, propiciando a expressão de qualquer função computacional, a alocação dinâmica de novos tipos, a chamada de métodos e a utilização de expressões de caminho. Esta mudança de paradigma, certamente, provocará novas adaptações no comportamento das pessoas e das Organizações, buscando se ajustarem à nova realidade, com reflexos no mundo corporativo que só o porvir poderá mostrar.

Referências

- CATTEL, Raymond; MEHLER, Barry. **The Object Database Standard: ODMG 3.0**. San Francisco, USA: Morgan Kaufmann Publishers, 2000.
- COAD, Peter; YOURDON, Edward. **Análise Baseada em Objetos**. 2. ed. São Paulo: Editora Campos, 1997.
- CODD, Frank. **A Relational Model of Data for Large Shared Data Banks**. New York, USA: Association for Computing Machinery, 1970.
- CHEN, Peter. **The Entity Relationship Model**. New York, USA: Association for Computing Machinery, 1976.
- FERREIRA, João; TAKAI, Osvaldo. **Persistência de Objetos**. São Paulo: Instituto de Matemática e Estatística da Universidade de São Paulo, p.1-46, abr 2005. Disponível em: <<http://www.ime.usp.br/~jef/persistencia.pdf>>. Acesso em: 17 nov 2007.
- MURTA, Leonardo; VERONESE, Gustavo; WERNE, Cláudia. MOR: Uma ferramenta para o Mapeamento Objeto-Relacional em Java. In: XV Simpósio Brasileiro de Engenharia de Software, 2001, Rio de Janeiro. **Anais**. Rio de Janeiro: IME, out 2001, p.392-397.
- YOSHIMA, Rodrigo. Mapeamento Objeto-Relacional: Tecnologia, impedância, herança. **Mundo Java**, Curitiba, n.24, p.16-21, jul/ago 2007.