FURB – BCC - TCC

Emulador de SGBD Orientado a Objetos

Fernando Colombo

Alexander R. Valdameri (Orientador)

Introdução (1)

Sistemas de Informação (SIs) são desenvolvidos para acessarem um SGBD, que possui papel crucial neste tipo de software. Em geral, desenvolvedores de SIs delegam ao SGBD as funcionalidades de acesso concorrente por vários usuários, recuperação em caso de falha, gerência de grandes volumes de dados, e até mesmo a integridade dos dados e aspectos relacionados à performance.

Portanto, a qualidade de um SI e a produtividade de seu desenvolvimento depende dos recursos de abstração da máquina e de representação da realidade que o SGBD oferece.

Introdução (2)

Infelizmente, os SGBDs mais populares e mais usados ainda são os SGBD Relacionais (SGBDRs). Embora eles desempenhem muito bem tarefas relacionadas a concorrência, grande volume de dados, recuperação, e performance, eles não oferecem recursos de abstração em suficiência para as necessidades de hoje, tendo em vista a crescente complexidade dos SIs.

Este trabalho desenvolve um software que permite que desenvolvedores de SIs programem como se o SGBD fosse orientado a objetos (SGBDOO), embora ele rode num SGBDR.

Objetivos

- **Implementar um Emulador de SGBD Orientado a Objetos (SGBDOO)**
- **O software deve expor todas as funcionalidades de um SGBDOO, conforme Atkinson et al (1989)**
- **Muitas funcionalidades são fornecidas por um SGBD Relacional, que é usado de forma transparente (daí o emprego da palavra emulador)**

Características de SGBDOO

Relacionadas a SGBDs:

- **Persistência**
- **Armaz. Secundário**
- **Concorrência**
- **Recuperação**
- **Consultas *ad hoc***

Relacionadas à OO:

- **Objetos complexos**
- **Identidade de objetos**
- **Encapsulamento**
- **Tipos (ou Classes)**
- **Herança**
- **Extensibilidade**
- **Completude Computacional**

Vertentes da Indústria

Devido à forte presença dos SGBDRs na indústria e no mercado, há duas correntes para trazer a orientação a objetos ao mundo dos SGBDs:

- **PURISTAS: conceituam um SGBD Orientado a Objetos de maneira independente dos SGBDRs – utilizam a sigla SGBDOO;**
- **CONSERVADORES: conceituam um SGBD Orientado a Objetos como um SGBDR que possui extensões para suporte à OO – utilizam a sigla SGBDOR (SGBD Objeto-Relacional).**

Modelos mais usados

- **PURISTAS: usam a especificação *The object data standard: ODMG 3.0* (Catell et al, 2000), ou simplesmente "ODMG 3". Foi desenvolvida por um consórcio composto pela maioria dos desenvolvedores de SGBDOO. ODMG significa *Object Data Management Group*.**
- **CONSERVADORES: usam a norma ISO/IEC 9075-*:1999, publicada pela ISO, e desenvolvida por um comitê subordinado a ela. É conhecida como SQL3, e trata-se da evolução da SQL92.**

Especificação ODMG 3

Principais características:

- **foco na portabilidade para usuários de SGBDOOs;**
- **suporta tipos complexos como listas, matrizes, e mapas;**
- **define as linguagens OQL (para consultas *ad hoc*) e ODL (para definição de objetos);**
- **obriga a utilização de uma linguagem entre C++, Java e Smalltalk, para completude computacional.**

ODMG 3 – Modelo de objetos

- **todo objeto tem um tipo, que define o comportamnto do objeto e/ou seus possíveis estados (atributos);**
- **o estado de qualquer objeto é definido pelo valor que possui em seus atributos;**
- **todo objeto possui uma identidade que o distingue dos demais;**
- **a identidade de qualquer objeto é um valor abstrato que independe do estado do mesmo.**

ODMG 3 – Suporte a herança

- **é suportada herança simples de estados e de comportamento;**
- **herança múltipla de estados não é suportada – somnte herança múltipla de comportamento;**
- **o suporte a herança é idêntico ao encontrado nas linguagens Smalltalk, Java e Delphi;**
- **o suporte a herança difere do encontrado nas linguagens C++ e Eiffel, nas quais herança múltipla de estados é suportada.**

ODMG 3 - Coleções

São definidos os seguintes tipos de coleção:

- **conjunto (não-ordenada, não permite duplicatas);**
- **sacola (não-ordenada, permite duplicatas);**
- **lista (ordenada, permite duplicatas, tamanho variável);**
- **matriz (ordenada, permite duplicatas, tamanho fixo);**
- **dicionário (coleção não-ordenada de pares chave-valor, não permite chaves duplicadas).**

ODMG 3 – Linguagem ODL

É a linguagem de definição de objetos (ODL = *Object Definition Language*)

```
module exemploODL {

  class Pessoa (extent Pessoas) {
    attribute string nome;
    attribute date datNas;
  };

  class Funcionario extends Pessoa (extent Funcionarios) {
    attribute float salario;
    relationship set<Dependente> deps inverse resp;
  };

  class Dependente extends Pessoa {
    relationship Funcionario resp inverse deps;
  };

};
```

ODMG 3 – Linguagem OQL (1)

- **é a linguagem de consulta de objetos (OQL = *Object Query Language*);**
- **desenvolvida com foco na simplicidade e na concisão;**
- **é semelhante à SQL, embora não seja compatível.**

ODMG 3 – Linguagem OQL (2)

Comandos válidos em OQL:

- `select nome, depto.nome from Funcionarios`
  (traz o nome de cada funcionário e seu departamento)
- `count(Funcionarios where salario > 1000)`
  (traz a quantidade de funcionários que ganham mais $ 1000)
- `avg(select idade from Funcionarios)`
  (traz a média de idade dos funcionários)

ODMG 3 - Miscelâneas

A especificação ainda define:

- **um modelo de acesso concorrente ao SGBDOO, incluindo travamentos;**
- **um modelo de transação e de recuperação em caso de falha;**
- **o ciclo de vida dos objetos, baseado em criação e destruição explícitas.**

Funcionalidades Adicionais

Esta trabalho propõe as seguintes funcionalidades ao SGBDOO:

- **destruição automática de objetos quando os mesmos se tornarem inatingíveis;**
- **coleções globais;**
- **referências fortes, suaves, e fracas.**

Mapeamento OR

- **Define meios de se salvar objetos num SGBDR. É composto plos seguintes aspectos:**
- **mapeamento da identidade dos objetos;**
- **mapeamento da herança;**
- **mapeamento das associações.**

Mapeamento OR - Identidades

- **cada objeto deve possuir um OID (*Object IDentifier*);**
- **devem haver colunas nas tabelas do SGBDR para manter o OID;**
- **o OID é um número inteiro de 64, 96 ou 128 bits;**
- **este número não deve ter qualquer significado para o usuário;**
- **uma vez que o valor do OID foi atribuido para um objeto, ele não deve mais mudar.**

Mapeamento OR - Herança

- **cada atributo deve mapear para exatamente uma coluna de uma tabela, no SGBDR;**
- **deve haver uma tabela por grupo de classes, ou uma tabela por classe;**
- **usa-se chaves estrangeiras para garantir a ligação com a tabela das classes-base.**

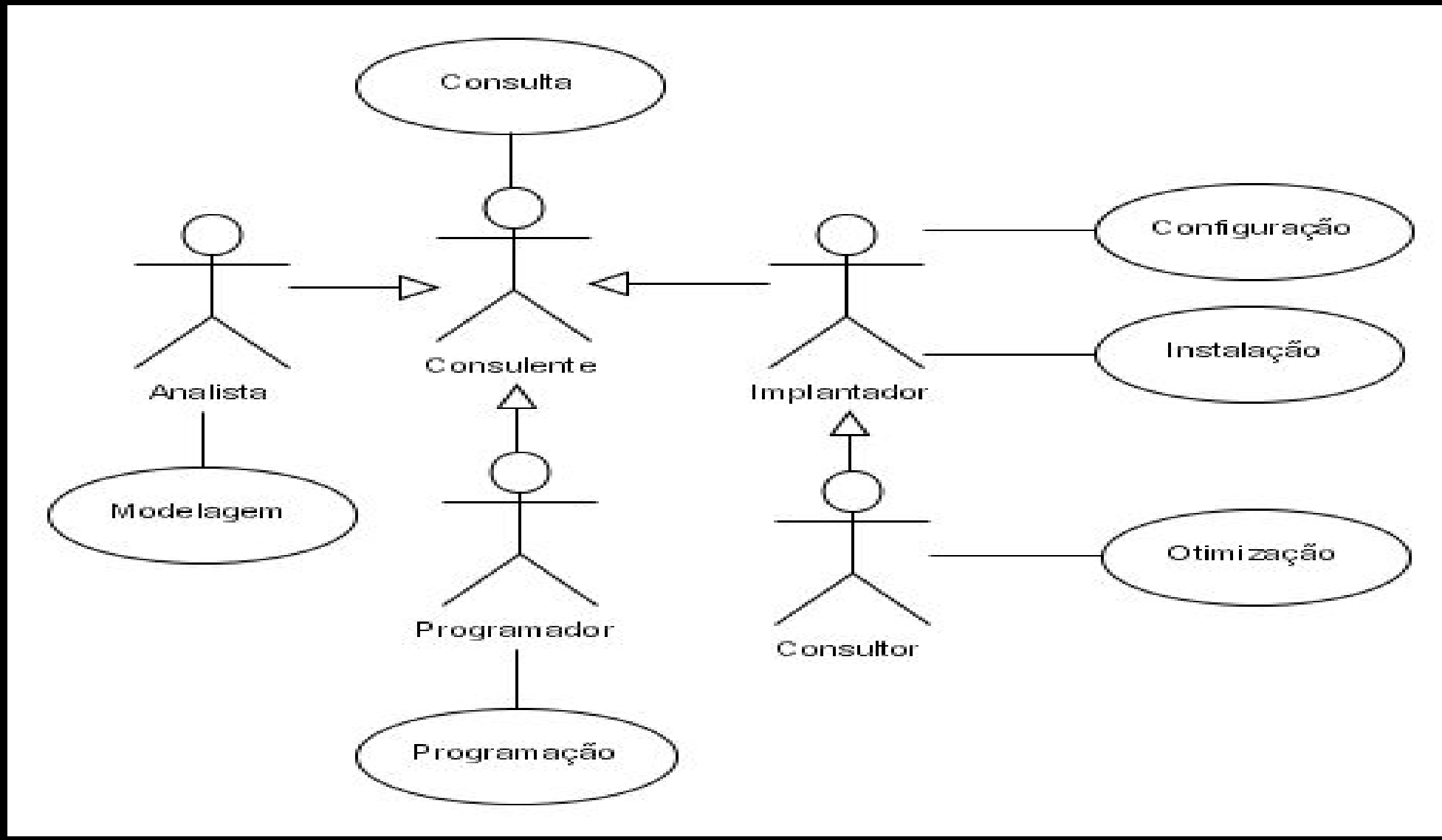
Mapeamento OR - Associações

- **associações 1-para-n são implementadas através de uma chave estrangeira na tabela do lado n, que referencia a tabela do lado 1;**
- **associações 1-para-1 são implementadas através de chaves estrangeiras nas tabelas dos dois lados, cada uma referenciando a tabela do outro lado;**
- **associações m-para-n são implementadas através de tabelas intermediárias.**

Especificação do Software

- **a especificação ODMG 3 é usada como base teórica, mas não há intenção de o software suportá-la;**
- **há suporte a herança múltipla no estilo do C++;**
- **há suporte a destruição automática de objetos;**
- **há suporte a coleções globais;**
- **há suporte a referências fortes, suaves e fracas;**
- **criação, destruição e alteração de objetos deve ser feito através de um programa escrito em Java.**

Papel dos Usuários



Representação de Objetos

- **para cada classe modelada há uma classe Java;**
- **todos os atributos e associações são mapeados para *getters* e *setters* na classe Java.**

```
void criaFuncionario(IObjectContext db) throws Exception {

    // Cria uma nova instância.
    Funcionario f =
      (Funcionario) db.newInstance(Funcionario.class);

    // Seta o valor dos atributos.
    f.setNome("Smith");
    f.setTitulo("Agent");
    f.setDatNas(GregorianDates.newDate(1968, 10, 2));

    // Persiste a instância.
    f.save();

}
```

Contexto de Objetos

- **representa uma conexão com o SGBDOO;**
- **trata-se de um objeto da classe IObjectContext;**
- **necessário para:**
  - **criar objetos;**
  - **efetuar consultas em OQL;**
  - **trabalhar com transações.**

Consultas Simples

Usa-se o método query(), de IObjectContext.

```
void imprimeAgentes(IObjectContext db) {

  Collection agents = db.query(
    "Funcionarios where titulo='Agent'");

  for (Iterator i = agents.iterator(); i.hasNext();) {

    Funcionario agente = (Funcionario) i.next();
    System.out.println(agente.getNome());

  }

}
```

Consultas Otimizadas

Consiste em preparar uma consulta para que ela seja executada várias vezes, com parâmetros diferentes.

```
void mostraFuncionarios(IObjectContext db, int[] empresas) {

  IQuery q = db.newQuery();
  q.setStmt("Funcionarios where empregador.codigo=:0");

  for (int i = 0; i < empresas.length; ++i) {

    System.out.println("Empresa " + empresas[i] + ":");
    q.setParam(0, new Integer(empresas[i]));

    for (Iterator j = q.run().iterator(); j.hasNext(); ++j) {
      Funcionario f = (Funcionario) j.next();
      System.out.println("  " + j.getNome());
    }

  }

}
```

Otimizações Permitidas

São suportadas as seguintes opções para otimização:

- **interferência no mapamento, na qual o usuário define quais tabelas são criadas no SGBDR para o armazenamento de objetos;**
- **interferência no script DDL, na qual o usuário cria índices e aloca espaço para as tabelas, de modo a otimizar os acessos.**

Ferramentas Utilitárias

As seguintes ferramentas fazem parte do software:

- **ferramenta de consultas interativas em OQL;**
- **gerador de arquivo de mapeamento OR;**
- **gerador de fontes Java para classes modeladas;**
- **gerador de script DDL para criação de tabelas no SGBDR.**

Técnicas Usadas

- **a metodologia orientada a objetos;**
- **a metodologia de desenvolvimento em espiral;**
- **a disciplina de compilação;**
- **a disciplina de banco de dados;**
- **a linguagem UML;**
- **a linguagem Java.**

Ferramentas Usadas

- **Poseidon for UML, versão 1.6 (modelagem UML);**
- **Eclipse IDE, versão 3.0 (ambiente Java);**
- **JavaCC, versão 2.1 (gerador de *parsers*);**
- **Velocity, versão 1.3 (gerador de texto a partir de modelos);**
- **MySQL, versão 4.0 (implementação de referência).**

Implementação de Referência

- **suporta o SGBDR MySQL;**
- **implementa 100% da especificação;**
- **serve para implementar o suporte a um SGBDR rapidamente.**

Execução das Consultas

Premissas para boa performance em consultas:

- **comandos OQL são convertidos em comandos SQL do SGBDR ao máximo possível;**
- **o comando SQL gerado utiliza ao máximo recursos específicos do SGBDR sendo usado.**

Limites Conhecidos

Os seguintes limites são bem conhecidos:

- **devido a leitura dos atributos ser sob demanda, não tem performance boa para aplicações de missão crítica;**
- **a instalação e configuração são tarefas trabalhosas;**
- **não há suporte para conversão do esquema do SGBDR, caso hajam mudanças na modelagem.**

Conclusão

- **o software expõe com sucesso as funcionalidades de um SGBDOO;**
- **certas funcionalidades são de inteira responsabilidade do SGBDR (como armazenamento secundário e recuperação) e da linguagem Java (como completude computacional);**
- **o software apresenta qualidade suficiente para um certo nicho de aplicações comerciais, mas não para aplicações de missão crítica.**

Fim desta apresentação