Abordagens para Reúso de Software – Visão Geral

Rosana T.Vaccare Braga

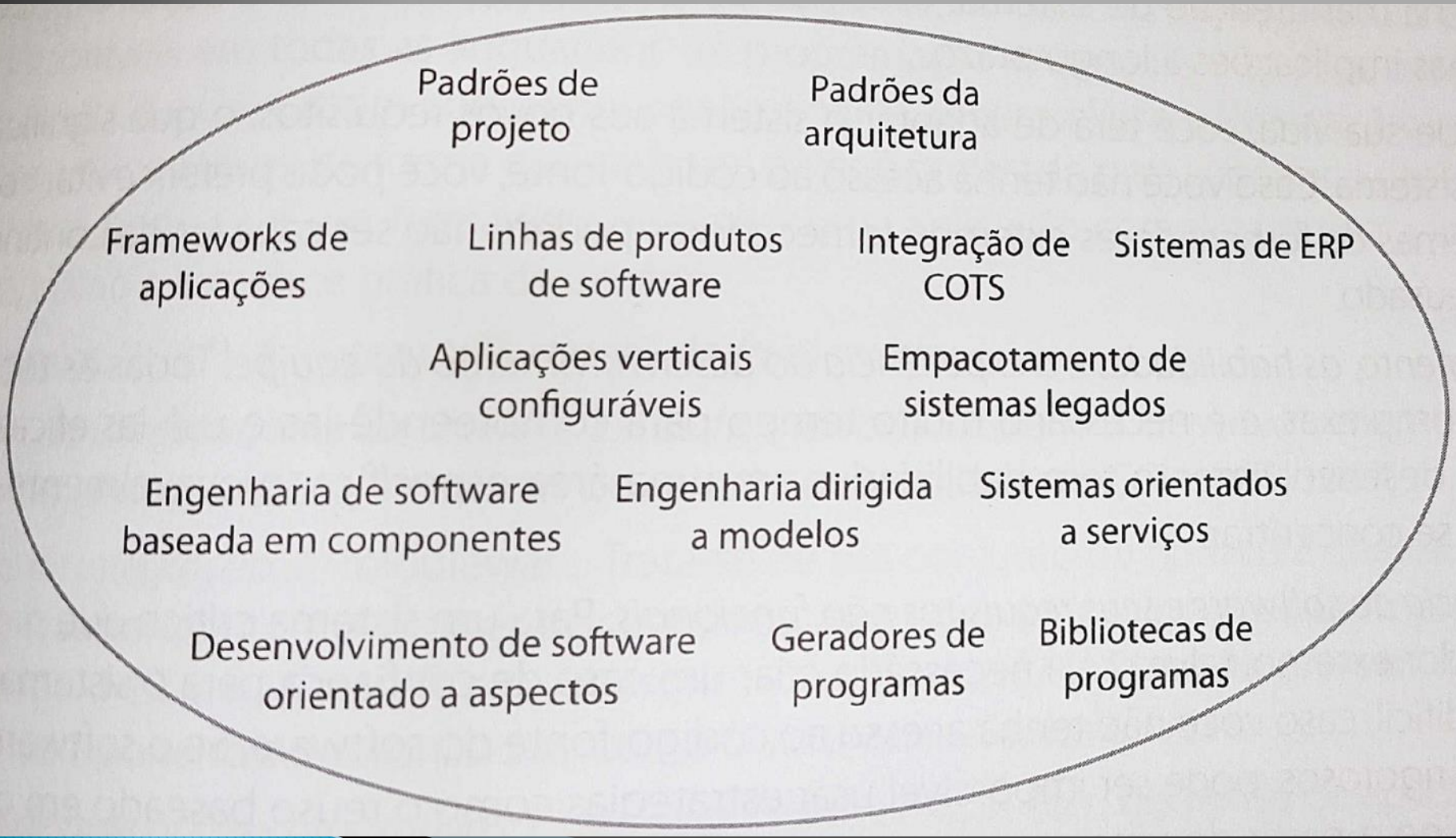
Introdução

- Relembrando: Reúso é uma abordagem de desenvolvimento que tenta maximizar a reutilização de software existente
  - Pontos positivos: Produtividade, qualidade, menor risco, uso de experiência de especialistas, conformidade com padrões
  - Pontos negativos: custos, faltam ferramentas, preconceito, dificuldade em procurar, compreender e adaptar ativos reusáveis

Reuso de Software

- replicação de qualquer tipo de conhecimento sobre um sistema em outros sistemas similares, com o objetivo de reduzir o esforço de desenvolvimento e a manutenção nesses novos sistemas *(Biggerstaff e Perlis, 1989)*.

O panorama de reúso (Sommerville)



Abordagens/ Tecnologias para Reúso

Abordagem	Descrição
Padrões de arquitetura	Padrões de arquitetura de software que oferecem suporte a tipos comuns de sistemas de aplicação são usados como base de aplicações. São descritos nos capítulos 6, 13 e 20.
Padrões de projeto	Abstrações genéricas que ocorrem em todas as aplicações são representadas como padrões de projeto, mostrando os objetos abstratos e concretos e as interações. São descritos no Capítulo 7.
Desenvolvimento baseado em componentes	Sistemas são desenvolvidos através da integração de componentes (coleções de objetos) que atendem aos padrões de modelos e componentes. São descritos no Capítulo 17.
Framework de aplicações	Coleções de classes abstratas e concretas são adaptadas e estendidas para criar sistemas de aplicação.
Empacotamento de sistemas legados	Sistemas legados (veja o Capítulo 9) são 'empacotados' pela definição de um conjunto de interfaces e acesso a esses sistemas legados por meio dessas interfaces.
Sistemas orientados a serviços	Sistemas são desenvolvidos pela ligação de serviços compartilhados, que podem ser fornecidos externamente. São descritos no Capítulo 19.
Linhas de produtos de software	Um tipo de aplicação é generalizado em torno de uma arquitetura comum para que esta possa ser adaptada para diferentes clientes.
Reúso de produto COTS	Sistemas são desenvolvidos pela configuração e integração de sistemas de aplicação existentes.

Abordagens/ Tecnologias para Reúso

Sistemas de ERP	Sistemas de grande porte que sintetizam a funcionalidade e as regras de negócios genéricos são configurados para uma organização.
Aplicações verticais configuráveis	Sistemas genéricos são projetados para poder ser configurados para as necessidades dos clientes de sistemas específicos.
Bibliotecas de programas	Bibliotecas de classe e funções que implementam abstrações comumente usadas são disponibilizadas para reúso.
Engenharia dirigida a modelos	O software é representado como modelos de domínio e modelos de implementação independentes. O código é gerado a partir desses modelos. São descritos no Capítulo 5.
Geradores de programas	Um sistema gerador incorpora o conhecimento de um tipo de aplicação, e é usado para gerar sistemas nesse domínio a partir de um modelo de sistema fornecido pelo usuário.
Desenvolvimento de software orientado a aspectos	Quando o programa é compilado, os componentes compartilhados são integrados em uma aplicação em diferentes locais. São descritos no Capítulo 21.

Abordagens/ Tecnologias para Reúso

Abordagem	Descrição
Padrões de arquitetura	Padrões de arquitetura de software que oferecem suporte a tipos comuns de sistemas de aplicação são usados como base de aplicações. São descritos nos capítulos 6, 13 e 20.
Padrões de projeto	Abstrações genéricas que ocorrem em todas as aplicações são representadas como padrões de projeto, mostrando os objetos abstratos e concretos e as interações. São descritos no Capítulo 7.
Desenvolvimento baseado em componentes	Sistemas são desenvolvidos através da integração de componentes (coleções de objetos) que atendem aos padrões de modelos e componentes. São descritos no Capítulo 17.
Framework de aplicações	Coleções de classes abstratas e concretas são adaptadas e estendidas para criar sistemas de aplicação.
Empacotamento de sistemas legados	Sistemas legados (veja o Capítulo 9) são 'empacotados' pela definição de um conjunto de interfaces e acesso a esses sistemas legados por meio dessas interfaces.
Sistemas orientados a serviços	Sistemas são desenvolvidos pela ligação de serviços compartilhados, que podem ser fornecidos externamente. São descritos no Capítulo 19.
Linhas de produtos de software	Um tipo de aplicação é generalizado em torno de uma arquitetura comum para que esta possa ser adaptada para diferentes clientes.
Reúso de produto COTS	Sistemas são desenvolvidos pela configuração e integração de sistemas de aplicação existentes.

Padrões de Software

- Padrões de Software:
  - Descrevem soluções para problemas que ocorrem com freqüência no desenvolvimento de software *(Gamma 95)*
- Um bom padrão:
  - resolve um problema
  - é um conceito aprovado (não apenas teoria)
  - a solução não é óbvia
  - descreve um relacionamento (estruturas e mecanismos)
  - tem um componente humano significativo

Tipos ou Categorias de Padrões

- Padrões podem ser
  - Organizacionais
  - Arquiteturais
  - De Análise
  - De Projeto
    - Gamma et al
    - Outros : p.ex J2EE
  - De programação (estilos)
  - De usabilidade
  - Educacionais
  - Etc.

Padrões de arquitetura

- Resolvem problemas ao se projetar a arquitetura de um sistema, delineando os elementos da arquitetura do software e seus relacionamentos
- Podem ser usados no início do projeto de alto nível, quando se faz a especificação da estrutura fundamental de uma aplicação
- Livro Sommerville
  - Cap. 6 – alguns padrões arquiteturais
  - Cap. 13 – padrões voltados a segurança
  - Cap. 20 – padrões voltados à sistemas embarcados

Tabela 5.2. Classificação dos padrões arquiteturais (Buschmann et al., 1996).

Categorias de Problemas	Categorias de Padrões		
	Padrões Arquitetônicos	**Padrões de Projeto**	**Idiomas**
Da Lama à Estrutura	Camadas Dutos e Filtros Quadro negro		
Sistemas Distribuídos	Intermediário Dutos e Filtros Microkernel		
Sistemas Interativos	Modelo-Visão-Controlador (MVC) Apresentação-Abstração-Controle		
Sistemas Adaptáveis	Microkernel Reflexão		
Decomposição Estrutural		Todo-parte	
Organização de Trabalho		Mestre-Escravo	
Controle de Acesso		Procurador	
Gerenciamento		Processador de Comandos Tratador de Visões	
Comunicação		Forwarder-Receiver Cliente-Despachante-Servidor Publicação-Assinatura	
Tratamento de Recursos			Ponteiro Contador

Padrões Arquiteturais

Nome do Padrão	Problema
Layers	Como projetar um sistema no qual a característica dominante é uma mistura de questões de alto e de baixo nível, onde operações de alto-nível necessitam de operações de mais baixo-nível?
Pipes and Filters	Como construir um sistema que deve processar ou transformar uma seqüência de dado de entrada?
Blackboard	Como transformar um seqüência de dados “crus” em uma estrutura de alto-nível, tais como tabelas, diagramas ou frases escritas?
Broker	Como construir sistemas de software complexos como um conjunto de componentes separados e trabalhando juntos, ao contrário de uma aplicação monolítica?
Model-View-Controller	Como construir interfaces do usuário fáceis de serem modificadas?
Presentation-Abstraction-Control	Como construir sistemas interativos decompostos em componentes que cooperam entre si para realizar uma determinada funcionalidade?
Microkernel	Como criar sistemas de software no qual o ciclo de vida é bastante longo, muitas vezes, de dez anos ou mais, e que necessitarão se adequar a novas tecnologias de software e hardware?
Reflection	Como projetar sistemas de larga-escala que mais tarde serão mudados e evoluídos?

Abordagens/ Tecnologias para Reúso

Abordagem	Descrição
Padrões de arquitetura	Padrões de arquitetura de software que oferecem suporte a tipos comuns de sistemas de aplicação são usados como base de aplicações. São descritos nos capítulos 6, 13 e 20.
Padrões de projeto	Abstrações genéricas que ocorrem em todas as aplicações são representadas como padrões de projeto, mostrando os objetos abstratos e concretos e as interações. São descritos no Capítulo 7.
Desenvolvimento baseado em componentes	Sistemas são desenvolvidos através da integração de componentes (coleções de objetos) que atendem aos padrões de modelos e componentes. São descritos no Capítulo 17.
Framework de aplicações	Coleções de classes abstratas e concretas são adaptadas e estendidas para criar sistemas de aplicação.
Empacotamento de sistemas legados	Sistemas legados (veja o Capítulo 9) são 'empacotados' pela definição de um conjunto de interfaces e acesso a esses sistemas legados por meio dessas interfaces.
Sistemas orientados a serviços	Sistemas são desenvolvidos pela ligação de serviços compartilhados, que podem ser fornecidos externamente. São descritos no Capítulo 19.
Linhas de produtos de software	Um tipo de aplicação é generalizado em torno de uma arquitetura comum para que esta possa ser adaptada para diferentes clientes.
Reúso de produto COTS	Sistemas são desenvolvidos pela configuração e integração de sistemas de aplicação existentes.

Padrões de projeto

- Resolvem problemas ao se projetar software, em especial os padrões GoF (Gang of Four – Gamma, 1994), software orientado a objetos
- São úteis durante o projeto, por envolver experiência de muitos anos dos autores no desenvolvimento de grandes projetos OO.

Exemplo: Catálogo de Padrões de Projeto [GOF 94]

Escopo		Propósito		
		De Criação	Estrutural	Comportamental
	Classe	Factory Method	Adapter	Interpreter Template Method
	Objeto	Abstract Factory Builder Prototype Singleton	Adapter Bridge Composite Decorator Facade Flyweygth Proxy	Chain of Responsability Command Iterator Mediator Memento Observer State Strategy Visitor

Padrão Observador

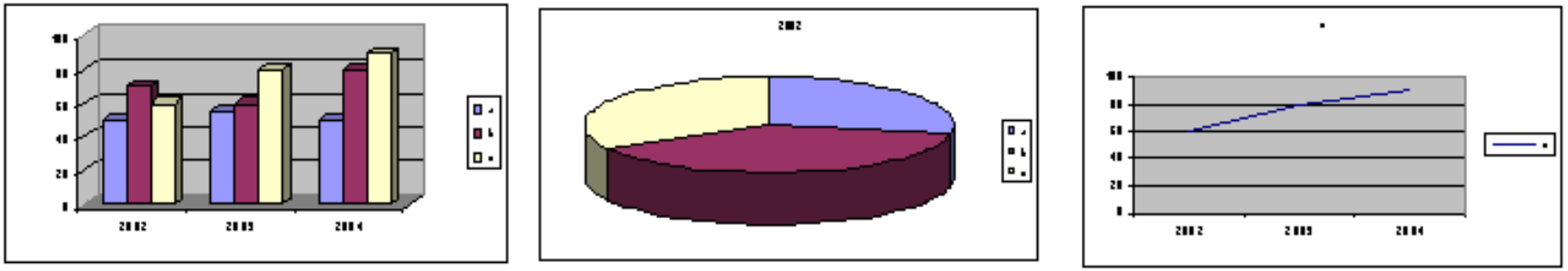
- Intenção: Definir uma dependência de um-para-muitos entre objetos, de forma que quando um objeto muda de estado, todos os seus dependentes são notificados e atualizados automaticamente.

Padrão Observer

- Aplicabilidade: Use o padrão Observer em quaisquer das seguintes situações:
  - Quando uma abstração tem dois aspectos, um dependente do outro. Encapsular esses aspectos em objetos separados permite variar e reutiliza-los independentemente.
  - Quando uma mudança em um objeto requer mudar outros, e não se sabe quantos objetos devem ser mudados.
  - Quando um objeto deve ser capaz de notificar outros objetos sem assumir quem são esses objetos. Em outras palavras, não é desejável que esses objetos estejam fortemente acoplados.

Padrão Observer

observadores



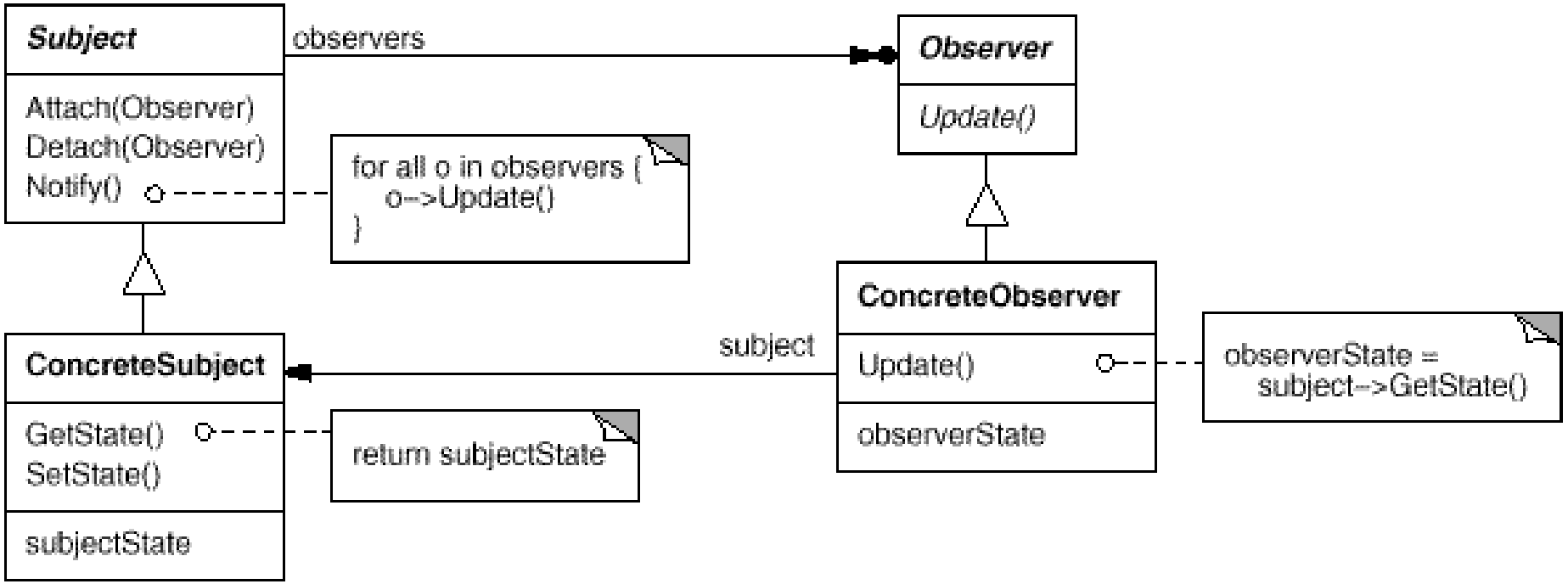
	2002	2003	2004
a	50	55	50
b	70	60	80
c	60	80	90

sujeito

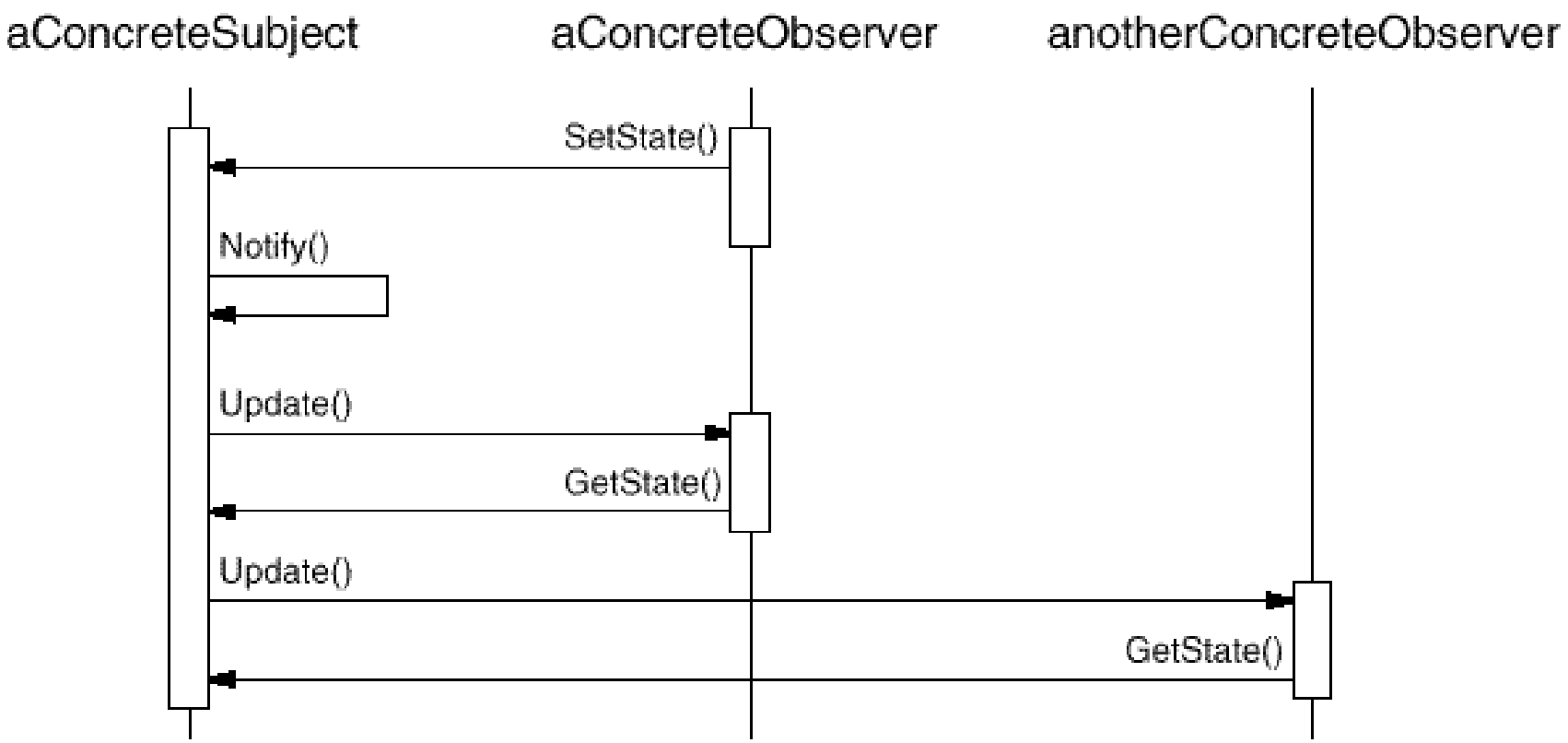
alterações, requisições

Notificação

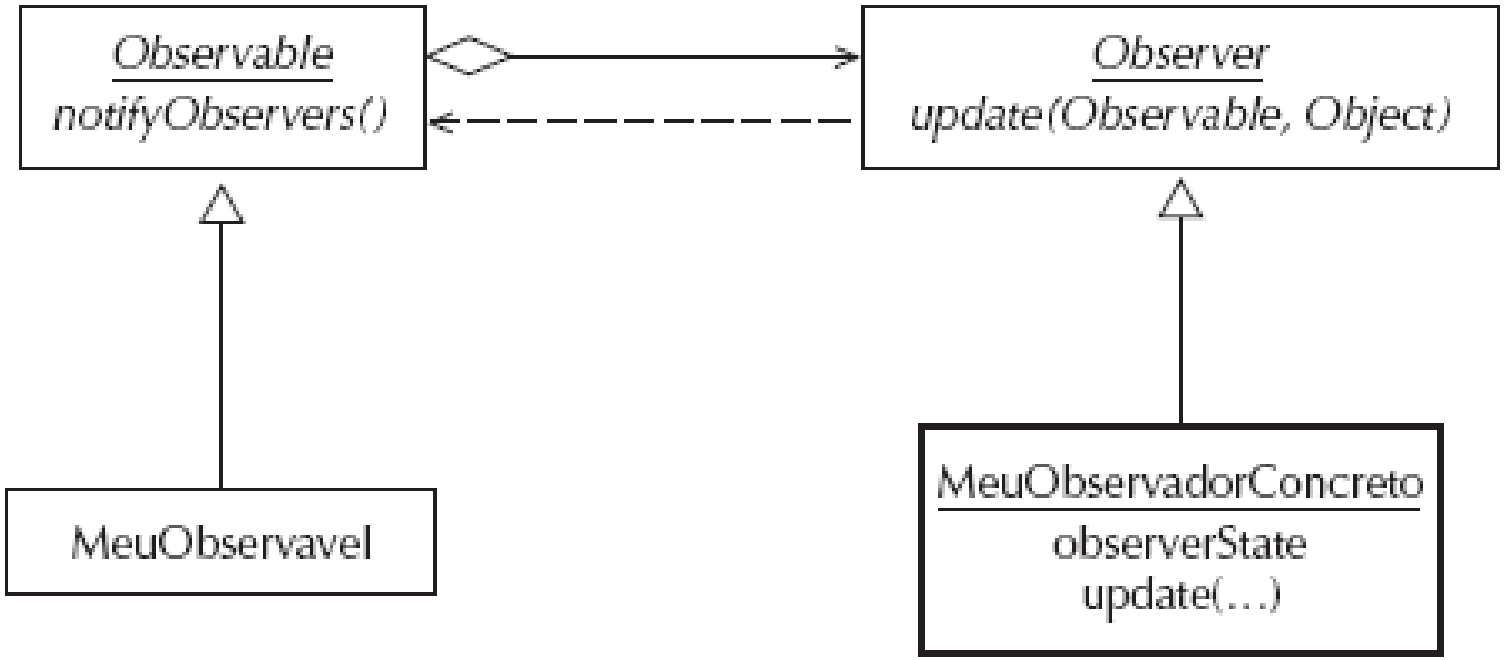
Padrão Observer



Padrão Observer



Observer na API Java



Legenda: Classe API Java | Classe do Desenvolvedor

Observer na API Java

- A API Java utiliza praticamente os mesmos termos de Gamma et al. [Ga].
- Observe que update(..) é um método retroativo (callback), porque fornece aos objetos Observer uma referência a sua origem, desta forma permitindo que eles comparem seus dados etc. com o objeto Observable na execução de update().
- Como a atualização (update) é implementada de modo retroativo, não há necessidade das classes concretas Observer manterem referências ao objeto Observable.

Abordagens/ Tecnologias para Reúso

Abordagem	Descrição
Padrões de arquitetura	Padrões de arquitetura de software que oferecem suporte a tipos comuns de sistemas de aplicação são usados como base de aplicações. São descritos nos capítulos 6, 13 e 20.
Padrões de projeto	Abstrações genéricas que ocorrem em todas as aplicações são representadas como padrões de projeto, mostrando os objetos abstratos e concretos e as interações. São descritos no Capítulo 7.
Desenvolvimento baseado em componentes	Sistemas são desenvolvidos através da integração de componentes (coleções de objetos) que atendem aos padrões de modelos e componentes. São descritos no Capítulo 17.
Framework de aplicações	Coleções de classes abstratas e concretas são adaptadas e estendidas para criar sistemas de aplicação.
Empacotamento de sistemas legados	Sistemas legados (veja o Capítulo 9) são 'empacotados' pela definição de um conjunto de interfaces e acesso a esses sistemas legados por meio dessas interfaces.
Sistemas orientados a serviços	Sistemas são desenvolvidos pela ligação de serviços compartilhados, que podem ser fornecidos externamente. São descritos no Capítulo 19.
Linhas de produtos de software	Um tipo de aplicação é generalizado em torno de uma arquitetura comum para que esta possa ser adaptada para diferentes clientes.
Reúso de produto COTS	Sistemas são desenvolvidos pela configuração e integração de sistemas de aplicação existentes.

COTS – Commercial Off-the-shelf

- Componentes comerciais prontos para uso podem estar disponíveis → reúso imediato
- Muito comuns nos anos 80 e 90, perderam a força por serem pouco flexíveis a novos requisitos do cliente

Abordagens/ Tecnologias para Reúso

Sistemas de ERP	Sistemas de grande porte que sintetizam a funcionalidade e as regras de negócios genéricos são configurados para uma organização.
Aplicações verticais configuráveis	Sistemas genéricos são projetados para poder ser configurados para as necessidades dos clientes de sistemas específicos.
Bibliotecas de programas	Bibliotecas de classe e funções que implementam abstrações comumente usadas são disponibilizadas para reúso.
Engenharia dirigida a modelos	O software é representado como modelos de domínio e modelos de implementação independentes. O código é gerado a partir desses modelos. São descritos no Capítulo 5.
Geradores de programas	Um sistema gerador incorpora o conhecimento de um tipo de aplicação, e é usado para gerar sistemas nesse domínio a partir de um modelo de sistema fornecido pelo usuário.
Desenvolvimento de software orientado a aspectos	Quando o programa é compilado, os componentes compartilhados são integrados em uma aplicação em diferentes locais. São descritos no Capítulo 21.

ERP (*Enterprise Resource Planning*)

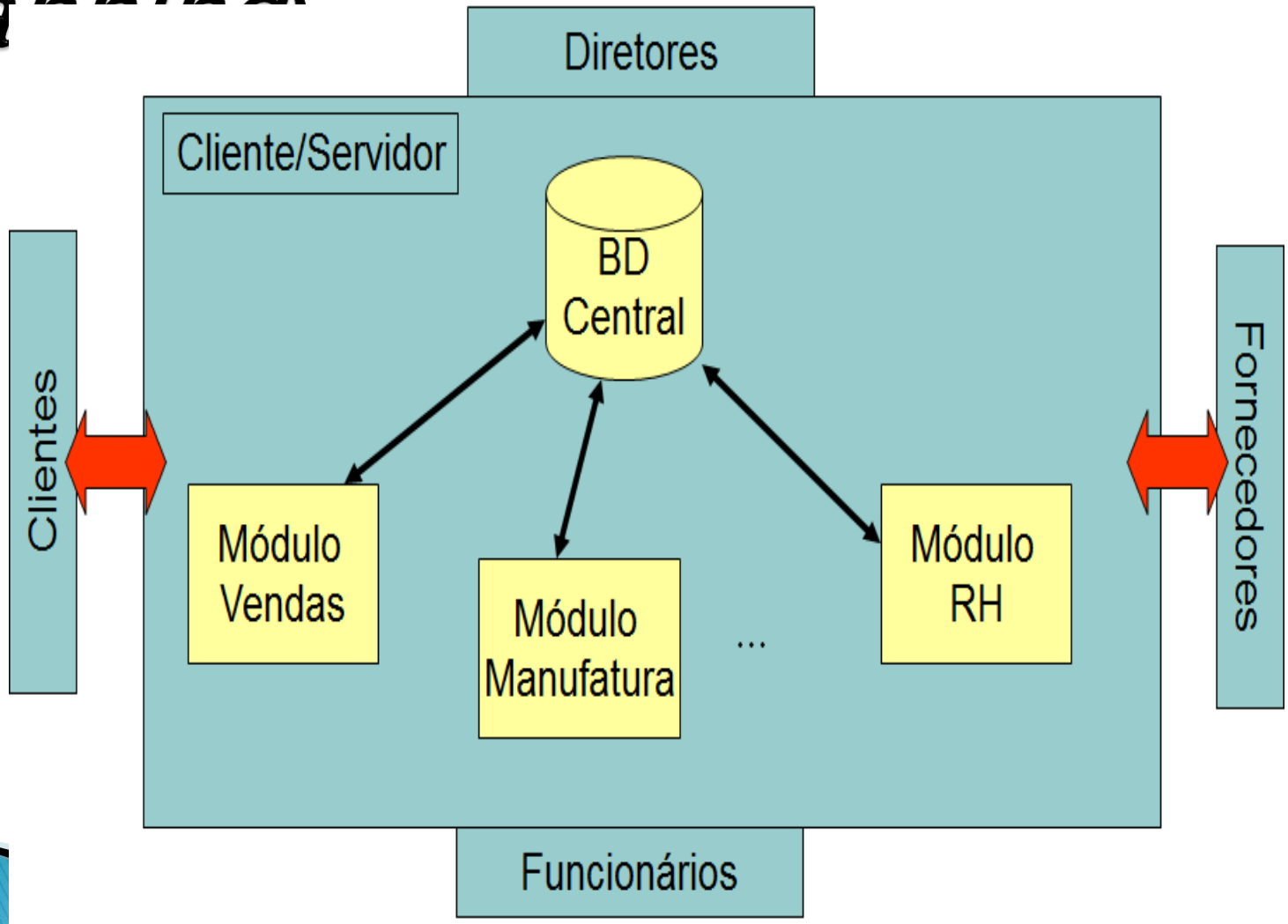
Sistemas de gestão de informação (**SIG**) que utilizam uma base de dados **única**. Possui diversos **módulos** que conversam entre si e trocam informações. Cada módulo é responsável por uma função específica do sistema, possibilitando à empresa acesso às informações de forma **integrada**, em uma única ferramenta e com um mesmo padrão de apresentação das informações.

ERP

Desenvolvidos de forma que uma **solução genérica** seja **customizada**.

Possui **arquitetura aberta**, possibilitando utilizar diferentes sistemas operacionais, bancos de dados, etc.

ERP (*Enterprise Resource Planning*)



ERP – Principais fornecedores:

Abordagens/ Tecnologias para Reúso

Sistemas de ERP	Sistemas de grande porte que sintetizam a funcionalidade e as regras de negócios genéricos são configurados para uma organização.
Aplicações verticais configuráveis	Sistemas genéricos são projetados para poder ser configurados para as necessidades dos clientes de sistemas específicos.
Bibliotecas de programas	Bibliotecas de classe e funções que implementam abstrações comumente usadas são disponibilizadas para reúso.
Engenharia dirigida a modelos	O software é representado como modelos de domínio e modelos de implementação independentes. O código é gerado a partir desses modelos. São descritos no Capítulo 5.
Geradores de programas	Um sistema gerador incorpora o conhecimento de um tipo de aplicação, e é usado para gerar sistemas nesse domínio a partir de um modelo de sistema fornecido pelo usuário.
Desenvolvimento de software orientado a aspectos	Quando o programa é compilado, os componentes compartilhados são integrados em uma aplicação em diferentes locais. São descritos no Capítulo 21.

Abordagens/ Tecnologias para Reúso

Sistemas de ERP	Sistemas de grande porte que sintetizam a funcionalidade e as regras de negócios genéricos são configurados para uma organização.
Aplicações verticais configuráveis	Sistemas genéricos são projetados para poder ser configurados para as necessidades dos clientes de sistemas específicos.
Bibliotecas de programas	Bibliotecas de classe e funções que implementam abstrações comumente usadas são disponibilizadas para reúso.
Engenharia dirigida a modelos	O software é representado como modelos de domínio e modelos de implementação independentes. O código é gerado a partir desses modelos. São descritos no Capítulo 5.
Geradores de programas	Um sistema gerador incorpora o conhecimento de um tipo de aplicação, e é usado para gerar sistemas nesse domínio a partir de um modelo de sistema fornecido pelo usuário.
Desenvolvimento de software orientado a aspectos	Quando o programa é compilado, os componentes compartilhados são integrados em uma aplicação em diferentes locais. São descritos no Capítulo 21.