

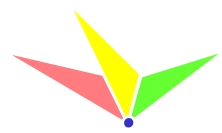
Padrões de Projeto J2EE

1

Introdução

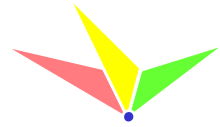
J2EE Blueprints

- *Procedimentos recomendados para desenvolver aplicações J2EE. Divide aplicações em camadas*
 - *Camada cliente: interface do usuário ou de serviços. Tipicamente representa uma aplicação independente ou browser rodando applets ou páginas HTML*
 - *Camada Web: consiste de servlets e páginas JSP com o objetivo de capturar requisições e processar respostas para a camada do cliente*
 - *Camada EJB: contém toda a lógica da aplicação e representa o modelo de negócio implementado em EJBs. Tipicamente é dividido em camadas adicionais*
 - *Camada de dados (EIS): consiste de sistemas de bancos de dados, transações e outros recursos legados*



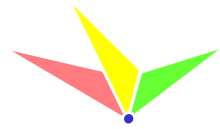
J2EE Patterns

- *Soluções de design baseadas no J2EE Blueprints*
 - *Representam soluções consideradas melhores práticas para implementar vários componentes essenciais em cada uma das camadas identificadas pelo J2EE Blueprints*
 - *Usam e se baseiam em vários padrões GoF*
- *São padrões de design e não de implementação (idioms)*
 - *Podem ser implementados usando tecnologias diferentes (por exemplo, usando RMI ou CORBA)*
- *Objetivos*
 - *Reduzir o tráfego de rede, aumentando a eficiência e facilitando a escalabilidade*
 - *Reduzir o acoplamento entre as camadas e os componentes*



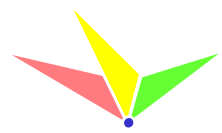
SJC J2EE Patterns

- Não existe um só catálogo de padrões
 - *Este curso se baseia no catálogo do Sun Java Center (SJC), cujos padrões estão documentados no site da Sun e em livros como "Core J2EE Patterns"*
 - *Os SJC J2EE Patterns são divididos em camadas lógicas que refletem a organização dos J2EE Blueprints*
- O catálogo atual (março/2003) define 15 padrões
 - *Camada de apresentação: 6 padrões*
 - *Camada de negócios: 7 padrões*
 - *Camada de integração: 2 padrões*
- Os nomes dos padrões são significativos



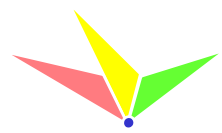
Padrões vs. Estratégias

- *Padrões refletem soluções para problemas genéricos descritos em abstrações de alto nível*
- *Estratégias mostram formas de implementar os padrões usando tecnologias e linguagens de programação*
- *Um padrão geralmente possui diversas diferentes estratégias de implementação*
- *Neste curso serão apresentados os padrões e suas principais estratégias*



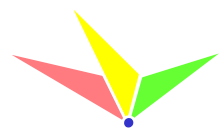
Padrões J2EE: apresentação

- (1) *Front Controller*
 - *Oferece um controlador centralizado para gerenciar o processamento de uma requisição*
- (2) *View Helper*
 - *Encapsula lógica não-relacionada à formatação*
- (3) *Composite View*
 - *Cria uma View composta de componentes menores*
- (4) *Intercepting Filter*
 - *Viabiliza pré- e pós-processamento de requisições*
- (5) *Service To Worker* e (6) *Dispatcher View*
 - *Combinam Front Controller com um Dispatcher e Helpers. O primeiro concentra mais tarefas antes de despachar a requisição. O segundo realiza mais processamento depois.*



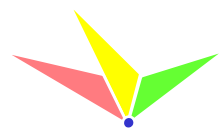
Padrões J2EE: negócios

- (7) *Business Delegate*
 - *Desacopla camadas de apresentação e de serviços*
- (8) *Value Object ou Transfer Object*
 - *Reduz tráfego na rede e facilita transferência de dados entre camadas*
- (9) *Session Façade*
 - *Esconde complexidade dos objetos de negócio e centraliza controle*
- (10) *Composite Entity*
 - *Agrupar composições de entity beans*
- (11) *Value Object Assembler ou Transfer Object Factory*
 - *Constrói um Value Object composto de múltiplas fontes*
- (12) *Value List Handler*
 - *Lida com execução de queries, caching de resultados, etc.*
- (13) *Service Locator*
 - *Encapsula complexidade de lógica de consulta e criação de objetos de serviço*

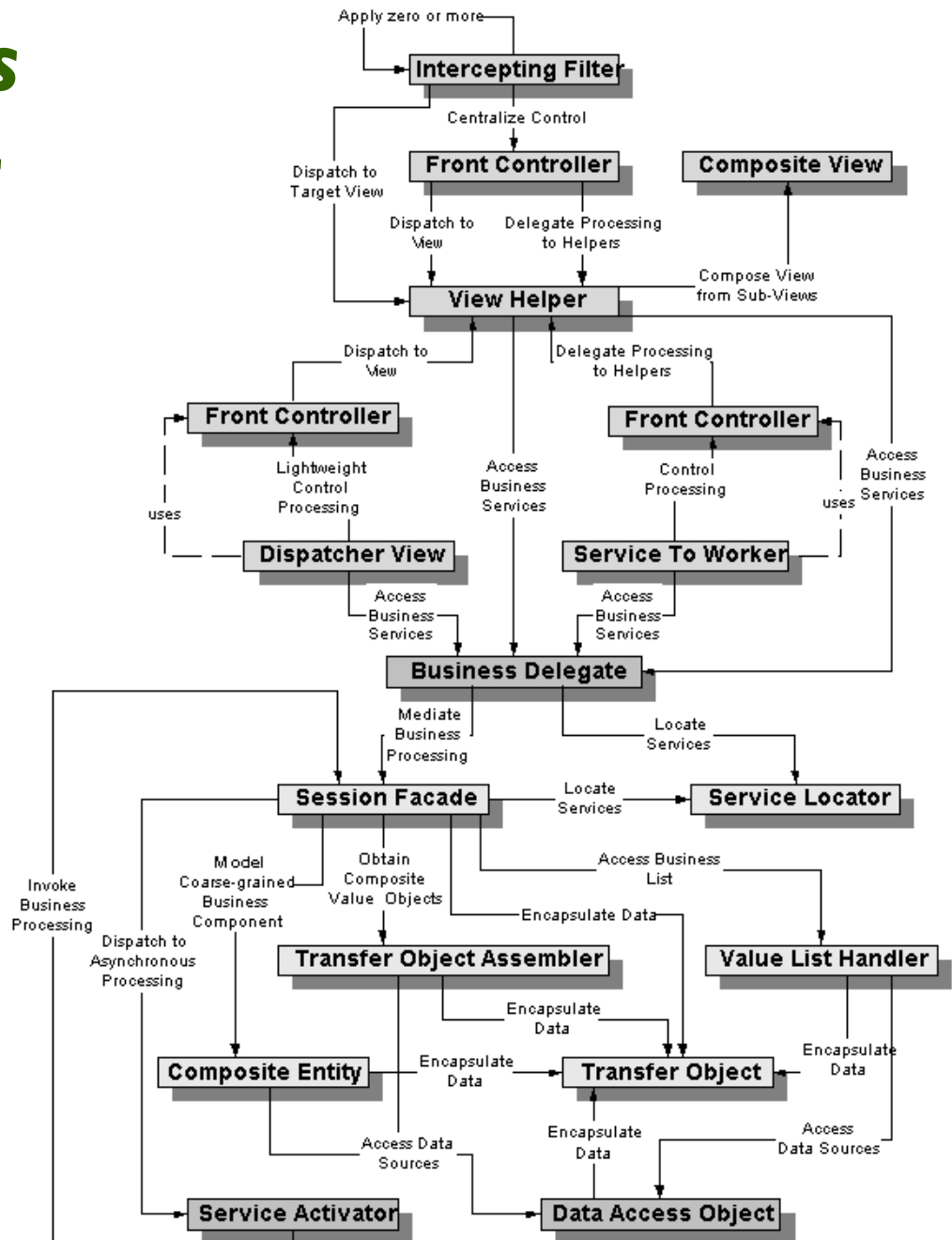


Padrões J2EE: integração

- *(14) Data Access Object*
 - *Abstrai fontes de dados e oferece acesso transparente aos dados*
- *(15) Service Activator*
 - *Facilita o processamento assíncrono para componentes EJB*



Relacionamentos entre os padrões



Fontes

[SJC] *SJC Sun Java Center J2EE Patterns Catalog.*

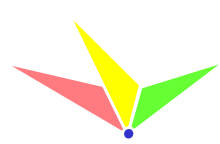
<http://developer.java.sun.com/developer/restricted/patterns/J2EEPatternsAtAGlance.html>.

[Blueprints] *J2EE Blueprints patterns Catalog.*

<http://java.sun.com/blueprints/patterns/catalog.htm>.

[Core] Deepak Alur, John Crupi, Dan Malks. *Core J2EE Patterns: Best Practices and Design Strategies.* Prentice-Hall, 2001.

<http://java.sun.com/blueprints/corej2eepatterns/index.html>.



Curso J93 I: J2EE Design Patterns

Versão 1.0

www.argonavis.com.br

© 2003, *Helder da Rocha*
(*helder@acm.org*)