

Transparent Façade

Anti-Padrão de Session Bean

Engenharia de Software com Ênfase em Padrões de Software
(UECE – Universidade Estadual do Ceará)

Padrões e Anti-Padrões J2EE

David de Almeida Ferreira
<davidferreira.fz@gmail.com>

Karine Roberta Vieira
<karinerobertavieira@gmail.com>

Transparent Façade

- Conhecido como
 - Direct Mapping
- Mais encontrado
 - Aplicação, Arquitetura
- Refatoramento
 - Session Façade
- Tipo de Solução
 - Software

Transparent Façade

- Causa Raiz
 - Falta de Conhecimento
- Forças Desbalanceadas
 - Complexidade e Recursos
- Evidencia
 - Eu li em algum lugar que deveríamos usar Session Façades

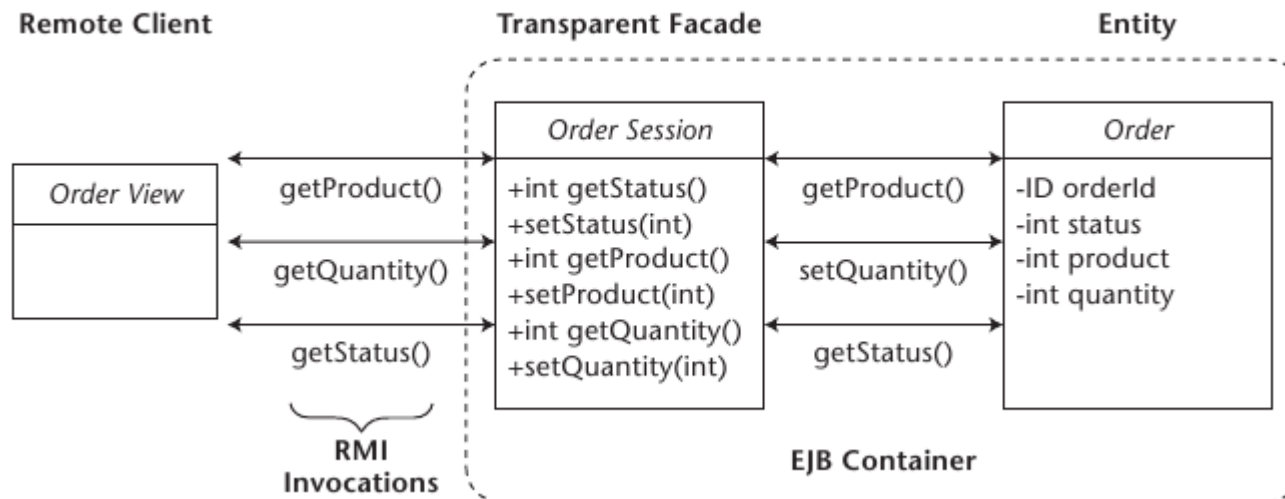
Transparent Façade

- Forma Geral

- É composto por um Session Bean e um Entity Bean, onde Session Bean contem diversos métodos base, possuindo uma alta granularidade.
- É chamado de Transparent Façade, devido a incorreta implementação do padrão Façade, pois neste caso a fachada não simplifica e nem esconde nada, ficando de fato transparente.

Transparent Façade

- Exemplo (Problema)



Transparent Façade

- Sintomas e Conseqüências
 - Baixa Performance
 - Não otimiza as chamadas remotas, já que externaliza a interface base da entidade;
 - Consumo de recursos devido a camada EJB adicional;

Transparent Façade

- Sintomas e Conseqüências
 - Alto acoplamento entre as interfaces de entidade, de sessão e do cliente
 - Por emular diretamente a interface da entidade, qualquer mudança na entidade provoca mudança na sua camada de sessão, o que leva a mudanças no cliente;
 - Devido essa características mudanças no lado dos Entity Beans acarretam grandes impactos nas aplicações;

Transparent Façade

- Causas Típicas
 - Falta de compreensão e/ou entendimento do padrão Session Façade
 - Inexperiência e falta de entendimento das intenções por trás do padrão Session Façade são as causas mais comuns deste anti-padrão.

Transparent Façade

- Causas Típicas

- Não entendimento do conceito de projeto modular e subsistema de camadas

- A motivação fundamental para o padrão Session Façade é a implementação de subsistemas em camadas, onde processos de granularidade baixa são compostos de camadas mais baixas de processos de granularidade alta. Subsistemas em camadas permitem processos de diferentes granularidades para serem usadas em diferentes contextos, e isto é o aspecto chave do projeto modular e flexível.

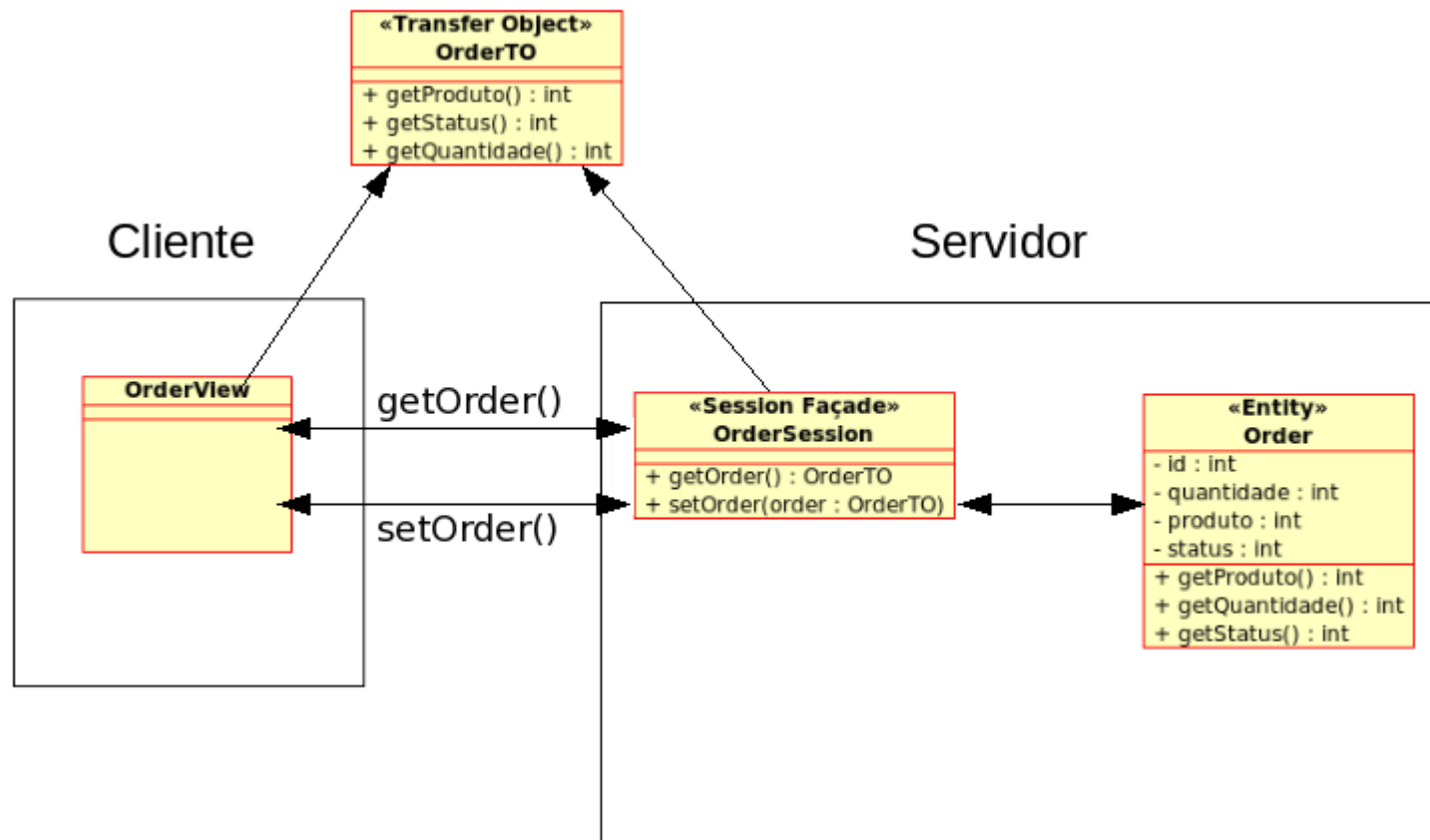
Transparent Façade

- Refatoração

- A refatoração de uma Transparent Façade consiste em aplicar o padrão Session Façade da forma correta;
- O resultado deve ser um Session Bean com poucos métodos, fazendo uso do padrão TO para reduzir as interações na rede;
- Os clientes que estavam usando a Transparent Façade, são então refatorados para interagir com o TO;

Transparent Façade

- Exemplo (Solução)



Você pode:



Copiar, distribuir, exibir e executar a obra



Criar obras derivadas



Sob as seguintes condições:



Atribuição. Você deve dar crédito aos autores originais [David de Almeida Ferreira](#) e [Karine Roberta Vieira](#) (incluindo o link : <http://davidferreirafz.wordpress.com>)



Compartilhamento pela mesma Licença. Se você alterar, transformar, ou criar outra obra com base nesta, você somente poderá distribuir a obra resultante sob uma licença idêntica a esta.

- Para cada novo uso ou distribuição, você deve deixar claro para outros os termos da licença desta obra.
- Qualquer uma destas condições podem ser renunciadas, desde que Você obtenha permissão do autor.
- Nothing in this license impairs or restricts the author's moral rights.

