



Fabrício Cabral &lt;fabriciofx@gmail.com&gt;

## Dúvida no livro sobre DDD do Evans - Posicionamento da classe FundsTransferService

7 messages

**Fabrício Cabral** <fabriciofx@gmail.com>  
To: dotnetarchitects@googlegroups.com

Mon, May 7, 2012 at 6:11 PM

Olá todos,

olha no livro de DDD do Evans, no exemplo "Partitioning Online Banking Functionality into Layers", na figura 4.1, vemos, claramente, que ele posiciona a classe FundsTransferService na camada de aplicação.

No entanto, na parte sobre Serviços, na seção "SERVICES and Isolated Domain Layer", podemos ver:

"On the other hand, a feature that can transfer funds from one account to another is a domain SERVICE because it embeds significant business rules (crediting and debiting the appropriate accounts, for example) and because a "funds transfer" is a meaningful banking term."

Pergunto: e agora? O Evans trocou as mãos pelas pernas? Na minha opinião, a classe FundsTransferService está mais para a camada de domínio do que para a camada de aplicação. O que vocês acham?

[]'s

--

-fx

**Rodrigo Silva de Andrade** <dangling@gmail.com>  
Reply-To: dotnetarchitects@googlegroups.com  
To: dotnetarchitects@googlegroups.com

Mon, May 7, 2012 at 7:19 PM

Depende, eu imagino que a classe possa existir tanto na domain layer quanto na application layer. a questão é que uma lida com a interação entre o domínio e a outra com a infraestrutura. sua classe mexe com qual?

2012/5/7 Fabrício Cabral <fabriciofx@gmail.com>

[Quoted text hidden]

--

Você recebeu esta mensagem porque faz parte do grupo .Net Architects hospedado no Google Groups.  
Para postar envie uma mensagem para [dotnetarchitects@googlegroups.com](mailto:dotnetarchitects@googlegroups.com)  
Para sair do grupo envie uma mensagem para [dotnetarchitects+unsubscribe@googlegroups.com](mailto:dotnetarchitects+unsubscribe@googlegroups.com)  
Para mais opções visite o grupo em <http://groups.google.com/group/dotnetarchitects?hl=pt-br>

--

Você recebeu esta mensagem porque faz parte do grupo .Net Architects hospedado no Google Groups.  
Para postar envie uma mensagem para [dotnetarchitects@googlegroups.com](mailto:dotnetarchitects@googlegroups.com)  
Para sair do grupo envie uma mensagem para [dotnetarchitects+unsubscribe@googlegroups.com](mailto:dotnetarchitects+unsubscribe@googlegroups.com)  
Para mais opções visite o grupo em <http://groups.google.com/group/dotnetarchitects?hl=pt-br>

**Fabrício Cabral** <fabriciofx@gmail.com>  
To: dotnetarchitects@googlegroups.com

Mon, May 7, 2012 at 7:53 PM

Olá Rodrigo,

me corrija se eu estiver enganado, mas no livro do Evans, pelo que eu pude entender,

a classe FundsTransferService lida com o domínio E com a infraestrutura. Explicando melhor: esta classe lida com os objetos Account para fazer a transferência e também com a classe UnitOfWork Manager para assegurar a transação de transferência.

Ao meu ver, ele deveria:

1) Ter duas classes, sendo uma na camada de domínio, que faria a transferência entre os objetos Account e outra, na camada de aplicação, que utilizaria o UnitOfWork Manager para assegurar esta operação da camada de domínio;

2) Ter duas classes, sendo uma na camada de domínio, que faria a transferência entre os objetos Account E utilizaria o UnitOfWork Manager para assegurar esta operação e uma outra classe, na camada de aplicação, que só faria chamar este procedimento.

Então, qual a melhor opção? 1, 2 ou eu estou completamente errado? :) Opiniões são bem-vindas! :)

[]'s

2012/5/7 Rodrigo Silva de Andrade <[dangling@gmail.com](mailto:dangling@gmail.com)>

[Quoted text hidden]

--  
-fx

---

**Fernando Mondo** <[fernando.mondo@gmail.com](mailto:fernando.mondo@gmail.com)>

Tue, May 8, 2012 at 12:02 AM

Reply-To: [dotnetarchitects@googlegroups.com](mailto:dotnetarchitects@googlegroups.com)

To: [dotnetarchitects@googlegroups.com](mailto:dotnetarchitects@googlegroups.com)

Este exemplo do evans, foi bem simples. Ele usa uma classe conta para fazer a transferência. Neste contexto o serviço é de aplicação. Mas no segundo exemplo eu acho que ele esta falando sobre ter uma classe para realizar a transferência, tirando esta responsabilidade da classe conta, oque faz bastante sentido, e neste caso esta classe é um serviço de domínio.

[Quoted text hidden]

---

**Fabício Cabral** <[fabriofx@gmail.com](mailto:fabriofx@gmail.com)>

Tue, May 8, 2012 at 12:16 AM

To: [dotnetarchitects@googlegroups.com](mailto:dotnetarchitects@googlegroups.com)

@Fernando,

na verdade, no exemplo do primeiro contexto, ele usa uma classe da camada de aplicação (FundsTransferService) para fazer um crédito em um objeto Account (Conta) e um débito em outro objeto Account.

Eu entendi os contextos, mas IMHO, acho que no mínimo, ele dificultou um pouco a didática, haja vista que ele cita um exemplo e depois, mais para frente, muda aquele exemplo...

[]'s

2012/5/8 Fernando Mondo <[fernando.mondo@gmail.com](mailto:fernando.mondo@gmail.com)>

Este exemplo do evans, foi bem simples. Ele usa uma classe conta para fazer a transferência. Neste contexto o serviço é de aplicação. Mas no segundo exemplo eu acho que ele esta falando sobre ter uma classe para realizar a transferência, tirando esta responsabilidade da classe conta, oque faz bastante sentido, e neste caso esta classe é um serviço de domínio.

--  
-fx

**Fernando Mondo** <fernando.mondo@gmail.com>

Tue, May 8, 2012 at 9:46 AM

Reply-To: dotnetarchitects@googlegroups.com

To: dotnetarchitects@googlegroups.com

Na verdade, o exemplo é mais ou menos assim:

//o controller cria ou busca as entidades do dominio de um repositório e entrega para um serviço da aplicação

```
class transfererController{  
  
void method(){  
    FundsTransferService.Transferer(a123,a456, $100);  
}  
  
}
```

//a camada de aplicação não tem regra de negócio então ela só sabe iniciar o unitOfWork, chamar os metodos das classes de dominio e dar o commit no final

```
class FundsTransferService{  
  
void Transferer(Account a132, Account a234, Money value){  
    UnitOfWorkMenager.BeginTransaction();  
  
    confirm = a123.TransferTo(a234, value);  
  
    UnitOfWorkMenager.Commit();  
}  
  
}
```

//e quem realmente faz a transferencia é a própria conta, o que leva a uma discussão sobre se é responsabilidade de uma Conta transferir dinheiro dela para outra.

```
class Account{  
  
Confirm TransferTo(a234, Money value){  
  
    confirm = a234.Credit(value);  
    if(confirm)  
        this.debit(value);  
    UnitOfWorkManager.addToUnitOfWork(this);  
    return true;  
}  
  
void credit(Money value){  
    UnitOfWorkManager.addToUnitOfWork(this);  
}  
  
}
```

Ah alguns pontos que eu gostaria de observar.

\* Ele deixou bem claro no livro que era um exemplo o mais simples possível, então em uma parte mais avançada ele cria um serviço de domino para realizar a transferencia de uma conta para outra isto esta perfeitamente correto.

\* A própria classe de dominio diz para o unitOfWork que foi modificada, isto eu não concordo e prefiro deixar na camada de aplicação.

\* A maior confusão se dá com o uso excessivo da palavra Service, pois existem Service em todas as camadas do DDD.

[Quoted text hidden]

[Quoted text hidden]

--

Você recebeu esta mensagem porque faz parte do grupo .Net Architects hospedado no Google Groups.

Para postar envie uma mensagem para [dotnetarchitects@googlegroups.com](mailto:dotnetarchitects@googlegroups.com)

Para sair do grupo envie uma mensagem para [dotnetarchitects+unsubscribe@googlegroups.com](mailto:dotnetarchitects+unsubscribe@googlegroups.com)

Para mais opções visite o grupo em <http://groups.google.com/group/dotnetarchitects?hl=pt-br>

[Quoted text hidden]

---

**Igor** <[oroshy@gmail.com](mailto:oroshy@gmail.com)>

Tue, May 8, 2012 at 1:08 PM

Reply-To: [dotnetarchitects@googlegroups.com](mailto:dotnetarchitects@googlegroups.com)

To: [dotnetarchitects@googlegroups.com](mailto:dotnetarchitects@googlegroups.com)

Existe serviços de Dominio, serviços de Aplicação e serviços da camada de Serviços.

Posso ter serviços nessas 3 camadas.

[Quoted text hidden]