



Fabrício Cabral <fabriciofx@gmail.com>

Modelagem de papéis utilizando classes de associação

4 messages

Fabrício Cabral <fabriciofx@gmail.com>
To: UML-BR@yahoogrupos.com.br

Tue, Mar 12, 2013 at 4:17 PM

Olá todos!

Olha eu aqui de novo com aquelas minhas dúvidas "bobinhas" mas que fazem a gente pensar um pouquinho! :)

Primeiramente peço desculpas pelo e-mail que é longo, mas preferi assim para dar a maior quantidade de detalhes possível.

Estava lendo um livro aqui (calma MT! Eu sei que livro nacional é complicado! Mas pelo menos faz a gente pensar!) e, numa seção, o mesmo manda:

- Classes de Associação

Uma questão frequentemente mal modelada diz respeito a conceitos que no senso comum poderiam ser considerados subtipos, mas que na verdade são papéis. Por exemplo, em uma livraria, poderiam ser identificados conceitos como Comprador e Funcionário. Ao descobrir que ambos têm atributos em comum, como nome, endereço, etc., um analista menos avisado poderia criar uma superclasse Pessoa para generalizar esses dois conceitos, o que resultaria em um problema muito sério de modelagem, pois não trata de tipos diferentes de pessoas, mas de papéis que pessoas têm em relação a uma empresa.

...

A solução, nesse caso, é considerar que existe uma Pessoa que pode se relacionar com uma Empresa de pelo menos duas formas: como comprador ou como funcionário. As propriedades específicas de comprador (cartões de crédito, por exemplo) e de funcionário (salário, por exemplo), seriam propriedades da associação e não da pessoa. Para representar essas propriedades, define-se uma classe de associação para cada associação específica, como na figura.

<http://www.flickr.com/photos/62145274@N08/8552798042/in/photostream>

...

Para diferenciar a situação na qual se usa herança e a situação na qual se usa classe de associação, deve-se verificar se os subtipos do conceito considerado existem em função de um terceiro tipo ou não. Caso só exista de forma relativa, então se deve usar classe de associação. Por exemplo, ninguém pode ser *aluno* simplesmente. Para alguém ser aluno, deve haver uma instituição de ensino ou um professor. A pessoa tem de ser aluno *de* alguém ou alguma instituição.

Pode-se verificar, então, que é inadequado criar classes para representar tipos de pessoas que na verdade não são subclasses, mas papéis. Funcionários, professores, alunos, diretores, compradores, etc. nunca poderiam ser subclasses de Pessoa. Esses conceitos só fazem sentido quando relacionados a outro conceito, como empresa, escola, departamento etc. Deve-se ser professor *de* alguém, comprador *de* uma empresa,

diretor *de* um departamento e assim por diante.

Bom, depois desse longo texto, a pergunta que eu faço é: faz sentido esta modelagem de papéis como classes de associação? Já vi em outras modelagens, seguirem outras abordagens. Como vocês modelam papéis nos seus sistemas?

[]'s

--

--fx

Marcelo Elias Del Valle <mvallebr@gmail.com>

Wed, Mar 20, 2013 at 7:18 PM

Reply-To: UML-BR@yahoogrupos.com.br

To: UML-BR@yahoogrupos.com.br

Fabício,

Você sabe como eu penso a respeito disso, né?

Eu acho que só faria sentido modelar algo assim para uso com MDA. Aí eu te pergunto: você, que manja mais de MDA que eu, enxerga um modelo assim como útil quando modelando em MDA?

Eu não me enxergo, numa empresa, entregando esse modelo para um desenvolvedor e ele entendendo o que tem que ser feito.

Abraços,
Marcelo.

Em 12 de março de 2013 16:17, Fabrício Cabral <fabriciofx@gmail.com> escreveu:

> **

[Quoted text hidden]

> ...

>

> Para diferenciar a situação na qual se usa herança e a situação na qual
> se usa classe de associação, deve-se verificar se os subtipos do conceito
> considerado existem em função de um terceiro tipo ou não. Caso só
> exista de forma relativa, então se deve usar classe de associação. Por
> exemplo, ninguém pode ser *aluno* simplesmente. Para alguém ser aluno,
> deve haver uma instituição de ensino ou um professor. A pessoa tem de
> ser aluno *de* alguém ou alguma instituição.

>

> Pode-se verificar, então, que é inadequado criar classes para representar
> tipos de pessoas que na verdade não são subclasses, mas papéis.
> Funcionários, professores, alunos, diretores, compradores, etc. nunca
> poderiam ser subclasses de Pessoa. Esses conceitos só fazem sentido
> quando relacionados a outro conceito, como empresa, escola, departamento
> etc. Deve-se ser professor *de* alguém, comprador *de* uma empresa,
> diretor *de* um departamento e assim por diante.

>

> Bom, depois desse longo texto, a pergunta que eu faço é: faz sentido
> esta modelagem de papéis como classes de associação? Já vi em
> outras modelagens, seguirem outras abordagens. Como vocês modelam
> papéis nos seus sistemas?

>

> []'s

>

> --

> --fx

>

> [As partes desta mensagem que não continham texto foram removidas]

>

>
>

—

Marcelo Elias Del Valle
<http://mvalle.com> - @mvallebr

[As partes desta mensagem que não continham texto foram removidas]

Links do Yahoo! Grupos

<*> Para visitar o site do seu grupo na web, acesse:
<http://br.groups.yahoo.com/group/UML-BR/>

<*> Para sair deste grupo, envie um e-mail para:
UML-BR-unsubscribe@yahoogrupos.com.br

<*> O uso que você faz do Yahoo! Grupos está sujeito aos:
<http://br.yahoo.com/info/utos.html>

Fabício Cabral <fabriciofx@gmail.com>
To: UML-BR@yahoogrupos.com.br

Mon, Mar 25, 2013 at 9:57 AM

Grande Marcelo! Tudo bem? :)

2013/3/20 Marcelo Elias Del Valle <mvallebr@gmail.com>
Fabício,

Você sabe como eu penso a respeito disso, né?
Eu acho que só faria sentido modelar algo assim para uso com MDA. Aí
eu te pergunto: você, que manja mais de MDA que eu, enxerga um modelo assim
como útil quando modelando em MDA?

Marcelo, com toda a sinceridade, eu tenho certeza **absoluta** que você sabe
mais MDA do que eu! :)

Agora, se este modelo é mais útil modelando em MDA, acho que vai depender
muito da ferramenta que vai pegar este UML e transformar em código.

Eu não me enxergo, numa empresa, entregando esse modelo para um
desenvolvedor e ele entendendo o que tem que ser feito.

Marcelo, só por curiosidade, como você modelaria então esta questão de
papéis em um sistema?

[]'s

—

—fx

Marcelo Elias Del Valle <mvallebr@gmail.com>
Reply-To: UML-BR@yahoogrupos.com.br

Sun, Apr 7, 2013 at 10:17 PM

To: UML-BR@yahoo grupos.com.br

Fala Fabrício,

Peço desculpas, fiquei com acesso restrito ao meu email por um tempo, só hoje vi a mensagem. Respondendo a sua pergunta de como eu modelaria, eu digo que é possível criar um modelo com o exato mesmo efeito do que aquele com classe de associação, sem utilizar a mesma. Então, para elucidar, considere o seguinte código JAVA, para iterar em um List<String>:

```
List<String> minhaLista = /*... algo aqui ...*/ ;  
for (int i=0; i<minhaLista.size(); i++)  
    System.out.println(minhaLista.get(i));
```

Esse código imprime no console os elementos da lista. Contudo, a partir do Java 5, foi criado um atalho para isso:

```
List<String> minhaLista = /*... algo aqui ...*/ ;  
for (String elemento: minhaLista)  
    System.out.println(elemento);
```

Nesse caso, a linguagem java permite 2 notações diferentes para realizar a mesma coisa. Qual a vantagem de utilizar a segunda? Clareza. A vantagem de utilizar a primeira? Maior flexibilidade.

Assim é para o caso da UML. Não utilizando a classe de associação, você pode modelar algo realmente fora do usual, mas as chances de errar no modelo e criar algo incoerente são muito maiores. O modelo equivalente sem utilizar a classe de associação teria uma associação direta entre as classes MAIS uma terceira classe que se relaciona com as duas. Você poderia utilizar uma cardinalidade diferente para a associação direta e para a classe sendo associada, por exemplo, o que seria incoerente na grande maioria dos casos.

Por isso, da mesma forma que no Java, o UML provê um "atalho" que te permite criar um modelo mais "lean" e ao mesmo tempo diminui suas chances de erro. Veja o link:

<http://etutorials.org/Programming/UML/Chapter+6.+Class+Diagrams+Advanced+Concepts/Association+Class/>

Nesse link, ele exemplifica isso em detalhes, além de explicar a diferença em detalhes também:

What benefit do you gain with the association class to offset the extra notation you have to remember? The association class adds an extra constraint, in that there can be only one instance of the association class between any two participating objects. I feel the need for an example.

Espero que ele mate completamente toda e qualquer dúvida que você tenha. Agora, como eu tinha dito antes, você acha que se você der um diagrama assim para um desenvolvedor, isso melhoraria a comunicação? Particularmente, duvido.

Abraços,
Marcelo.

Em 25 de março de 2013 09:57, Fabrício Cabral <fabriciofx@gmail.com> escreveu:

```
> **  
>  
>  
> Grande Marcelo! Tudo bem? :)  
>  
>  
> 2013/3/20 Marcelo Elias Del Valle <mvallebr@gmail.com>  
>  
> > Fabrício,  
> >
```

> > Você sabe como eu penso a respeito disso, né?
> > Eu acho que só faria sentido modelar algo assim para uso com MDA. Aí
> > eu te pergunto: você, que manja mais de MDA que eu, enxerga um modelo
> assim
> > como útil quando modelando em MDA?
> >
>
> Marcelo, com toda a sinceridade, eu tenho certeza **absoluta** que você sabe
> mais MDA do que eu! :)
>
> Agora, se este modelo é mais útil modelando em MDA, acho que vai depender
> muito da ferramenta que vai pegar este UML e transformar em código.
>
>
> Eu não me enxergo, numa empresa, entregando esse modelo para um
> > desenvolvedor e ele entendendo o que tem que ser feito.
> >
>
> Marcelo, só por curiosidade, como você modelaria então esta questão de
> papéis em um sistema?
>
>
> []'s
>
> --
> --fx
>

[Quoted text hidden]