

Discussão de Problema – Modelagem Funcional com Contratos

INFORMAÇÕES DOCENTE						
CURSO:	DISCIPLINA:	TURNO	MANHÃ	TARDE	NOITE	PERÍODO/SALA:
ENGENHARIA DE SOFTWARE	PROJETO DE SOFTWARE		x		x	4º
PROFESSOR (A): João Paulo Carneiro Aramuni						

Modelagem Funcional com Contratos

A modelagem funcional por contratos é uma abordagem usada na engenharia de software para especificar formalmente o comportamento de funções, métodos ou operações de um sistema, utilizando contratos que definem:

- I) Pré-condições: condições que devem ser verdadeiras antes da execução da função (ou operação).
- II) Pós-condições: condições que devem ser verdadeiras após a execução da função, se as pré-condições forem atendidas.
- III)(Opcionalmente) Invariantes: condições que sempre devem se manter verdadeiras, independentemente do estado do sistema.

Essa técnica é parte do paradigma conhecido como programação por contrato (Design by Contract), introduzido por Bertrand Meyer na linguagem Eiffel, mas que também pode ser usada na modelagem de requisitos e análise de sistemas.

Exemplo simples de contrato

Para um método sacar(valor) de um objeto ContaBancária, podemos ter:

- Pré-condição: valor > 0 e saldo >= valor
- Pós-condição: saldo == saldo_inicial - valor

Relação com os Diagramas de Sequência (UML)

Os diagramas de sequência mostram como os objetos interagem entre si ao longo do tempo, representando:

- A ordem das mensagens (métodos chamados),
- Os participantes (objetos ou atores),
- As ativações de métodos,
- E os possíveis retornos ou exceções.



PUC Minas

Como os contratos se relacionam:

I) Complementaridade:

- Enquanto o diagrama de sequência mostra o fluxo de mensagens, a modelagem por contratos especifica as condições para que essas mensagens sejam válidas e o que elas garantem.
- Ou seja, os contratos adicionam semântica e regras de negócio ao que os diagramas de sequência mostram visualmente.

II) Validação de comportamento:

- Você pode usar os contratos para validar se os cenários do diagrama de sequência estão corretamente especificados (ex: a mensagem sacar(200) só deve ocorrer se a pré-condição saldo \geq 200 for verdadeira).

III) Suporte à verificação formal e testes:

- Os contratos ajudam a gerar casos de teste, definir regras de verificação automática e melhorar a documentação do sistema.
- Já os diagramas de sequência ajudam na comunicação com stakeholders e no entendimento do fluxo de interações.

Integração prática no processo de modelagem

- I) Durante a análise ou design, você pode:
 - Criar um contrato para cada operação de um caso de uso.
 - Depois, modelar diagramas de sequência que representem cenários concretos onde essas operações são usadas.
- II) Na engenharia de requisitos, contratos ajudam a:
 - Formalizar regras do negócio.
 - Evitar ambiguidade nos requisitos funcionais.

Linguagem OCL - (*Object Constraint Language*)

A linguagem OCL (*Object Constraint Language*) é usada para expressar de forma formal e precisa os contratos em modelos UML, ou seja, as pré-condições, pós-condições e invariantes que definem o comportamento esperado de operações, atributos e classes.

Ao utilizar OCL, é possível descrever regras de negócio e restrições do sistema de maneira não ambígua, permitindo validar se os modelos estão corretos e completos. Dessa forma, OCL é a ferramenta ideal para aplicar a modelagem funcional por contratos dentro do contexto da UML. A linguagem foi originalmente criada pela IBM como parte de seus esforços para formalizar modelos orientados a objetos e, posteriormente, foi adotada pela OMG (*Object Management Group*) como padrão oficial da UML.

Exercício - Modelagem Funcional com Contratos (Especificando operações com OCL)

Você está modelando o sistema de uma biblioteca universitária, que precisa controlar o empréstimo de livros para seus usuários. Cada usuário pode ter no máximo um empréstimo ativo por vez, e cada empréstimo pode conter vários itens, sendo cada item um livro emprestado.

A biblioteca deseja implementar a operação `realizarEmprestimo(codigoLivro: String)`, que será chamada quando o usuário atual selecionar um livro disponível para emprestar.

Sua tarefa: Modele a operação `realizarEmprestimo(codigoLivro: String)` na classe Biblioteca, utilizando a técnica de modelagem funcional por contratos com OCL, incluindo:

- Alias para facilitar a leitura,
- Pré-condições que garantam que o empréstimo só será feito se for válido,
- Pós-condições que especifiquem a criação do empréstimo e do item associado, a atualização do estado do livro e os dados do item.

Você deve considerar os seguintes requisitos:

- O sistema já tem uma referência ao `usuarioAtual`.
- O livro só pode ser emprestado se estiver disponível.
- Se o usuário ainda não tem um empréstimo ativo, um novo objeto `Emprestimo` deve ser criado.
- Um novo `ItemEmprestimo` deve ser criado e adicionado ao `emprestimoAtual` do usuário.
- O livro emprestado deve passar a não estar mais disponível (`disponivel = false`).
- A categoria do livro define o prazo padrão e o valor diário do empréstimo.

Após modelar a operação usando OCL, crie o **diagrama de sequência** correspondente utilizando a notação PlantUML. Certifique-se de utilizar os comandos *activate* e *deactivate* para representar corretamente os períodos de ativação dos objetos participantes na sequência da operação.
