

Introdução à UML e Diagrama de Casos de Uso

Eduardo Figueiredo

<http://www.dcc.ufmg.br/~figueiredo>
dcc603@gmail.com

28 Março 2012

Tópicos da Aula

- Introdução a UML
- Visão geral de alguns diagramas
- Diagrama de Casos de Uso

A Linguagem UML

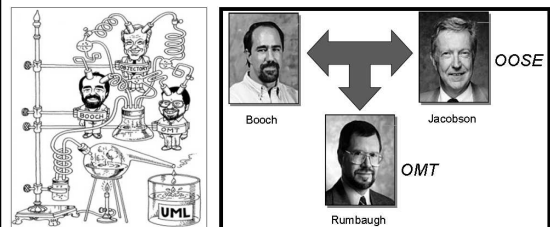
A Linguagem UML

- UML (Linguagem de Modelagem Unificada)
- É uma notação gráfica (visual) para projetar sistemas OO
 - Não é uma linguagem de programação
- Define diagramas padronizados
- É extensível
- É complexa (muitos diagramas)
 - Usaremos apenas um sub-conjunto da UML

De onde surgiu?

- Da união de três metodologias de modelagem
 - Método de Booch - Grady Booch
 - Método OMT - Ivar Jacobson
 - Método OOSE - James Rumbaugh
- Os três amigos

Fundadores da UML



História da UML

- 1994: Booch, Jacobson e Rumbaugh começaram a unificar suas notações
- 1996: Primeira versão (beta) da UML foi liberada
- 1996/97: Grandes empresas formaram a "UML Partners"
 - HP, IBM, Microsoft, Oracle, etc.
- 1997: UML foi adotada pela a OMG (*Object Management Group*) como linguagem padrão de modelagem

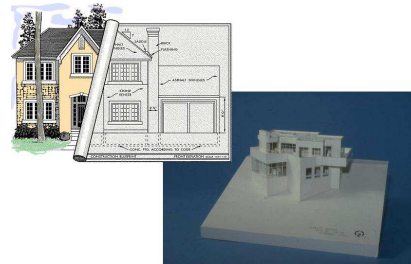
O que é modelagem?

- Atividade de construir modelos que expliquem as características ou comportamentos de um sistema
- A UML pode ser usada com todos os processos de software
 - Modelo Cascata, Iterativo, Espiral ...
- Utilizado principalmente em duas atividades dos processos
 - Especificação de requisitos
 - Projeto

Por que modelar?

- Tão essencial quanto ter uma planta antes da construção de uma casa
 - Melhora a comunicação entre os membros da equipe e o cliente
 - A equipe entende melhor o sistema
 - Permite analisar o sistema sobre vários aspectos
 - Facilita a programação e a manutenção
 - Diminui a possibilidade de erros

Projetar é Fundamental



Por que usar UML?

- Bons modelos são essenciais para a comunicação entre os *stakeholders*
- Padronização
 - Todo o time entende a modelagem, facilitando a manutenção
- Facilita a programação
 - Ferramentas para modelagem e geração de código

Modelagem Orientada a Objetos

- Maneira natural de visualizar o software
- Modela o software semelhante ao mundo real - usando objetos
 - Pessoas, animais, plantas, carros, etc.
- Humanos pensam em termos de objetos
 - Mais alto nível

UML Define Diagramas

- Tipos Principais de Diagramas
 - Estrutural
 - Comportamental
- Objetivos
 - Visualizar o sistema
 - Especificar estrutura e/ou comportamento
 - Guiar e documentar as decisões

Alguns Diagramas UML

- Diagramas Estruturais (Estáticos)
 - Diagrama de Casos de Uso
 - Diagrama de Classes
 - Diagramas de Objetos
 - Diagrama de Componentes, etc.
- Diagramas Dinâmicos
 - Diagrama de Sequência
 - Diagrama de Estados
 - Diagrama de Atividades
 - Diagrama de Colaboração, etc.

Resumo dos Diagramas UML

Diagrama de Caso de Uso

- Diagrama mais geral da UML
- Usado geralmente na fase de Especificação de Requisitos
- Mostra
 - Quais usuários realizam que funcionalidades do sistema
 - Alguns relacionamentos entre estas funcionalidades

Diagrama de Caso de Uso

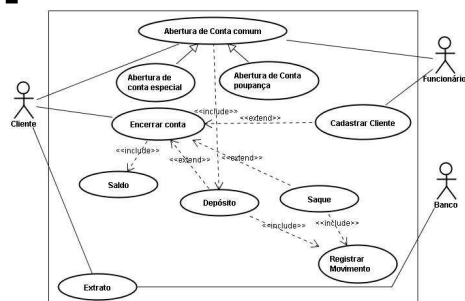


Diagrama de Sequência

- Preocupa-se com a ordem temporal em que as mensagens são trocadas
- Pode se basear em um Caso de Uso
- Identifica
 - Os eventos associados a funcionalidade modelada
 - O ator responsável por este evento

Diagrama de Sequência

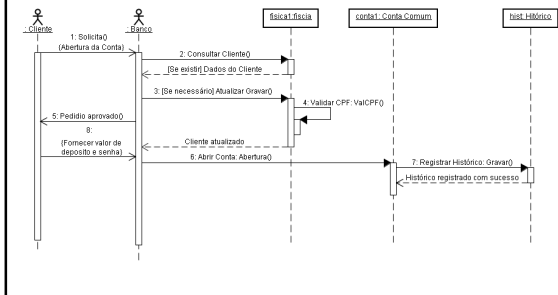


Diagrama de Classes

- Diagrama mais utilizado da UML
- Serve de apoio para a maioria dos outros diagramas
- Define a estrutura das classes do sistema
- Estabelece como as classes se relacionam

Diagrama de Classes

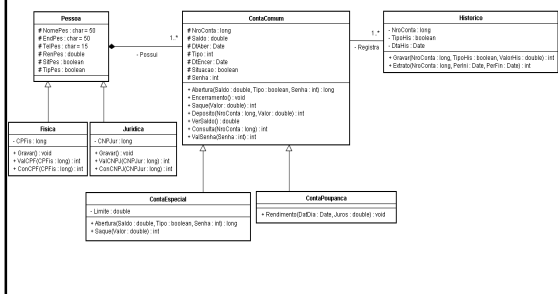


Diagrama de Objetos

- Complemento do Diagrama de Classes
- Exibe os valores armazenados pelos objetos de um Diagrama de Classes

Diagrama de Objetos

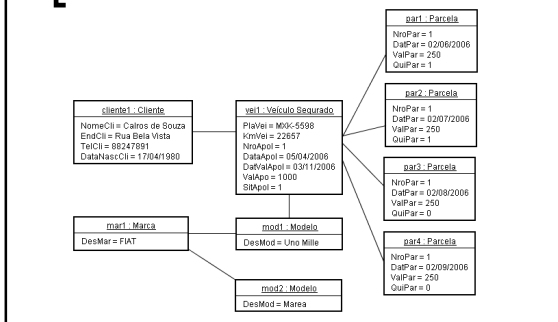


Diagrama de Colaboração

- Amplamente associado ao Diagrama de Sequência
- São complementares
- Não se preocupa com a temporalidade
- Define
 - Como os objetos estão vinculados
 - Quais mensagens são trocadas entre objetos

Diagrama de Colaboração

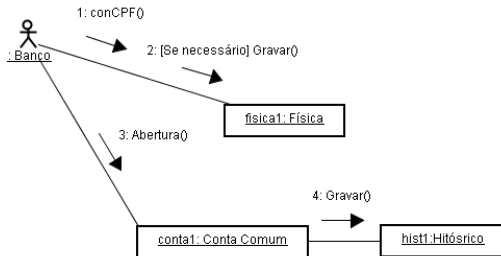


Diagrama de Estados

- Modela as mudanças sofridas por um objeto dentro de um determinado processo
- É utilizado para acompanhar os estados por que passa uma instância de uma classe

Diagrama de Estados

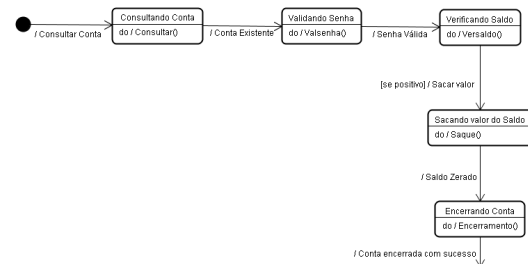


Diagrama de Atividades

- Descreve as atividades a serem executadas para a conclusão de um processo
- Concentra-se na representação do fluxo de controle de um processo

Diagrama de Atividades

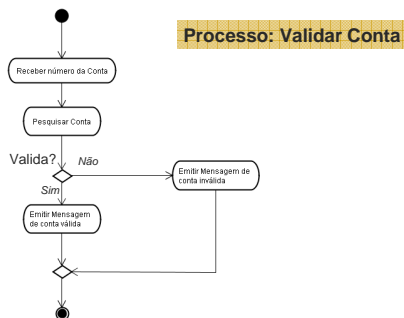


Diagrama de Componentes

- Associado a linguagem de programação a ser utilizada
- Este diagrama representa os componentes do sistema
- Os componentes serão implementados como
 - Classes de código-fonte
 - Bibliotecas
 - Arquivos de ajuda, etc.

Diagrama de Componentes

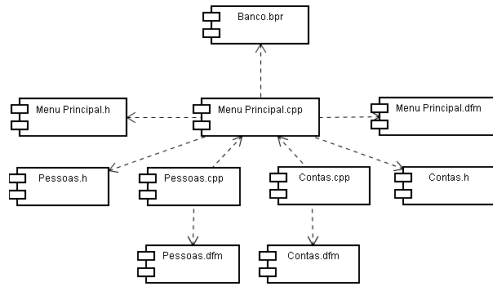
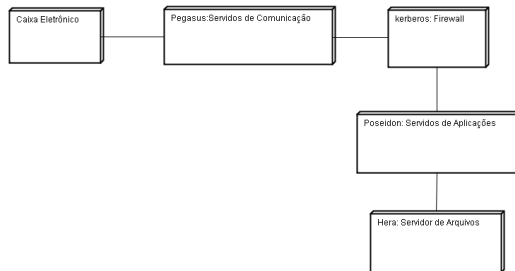


Diagrama de Implantação

- Determina as necessidades de hardware
- Características físicas do sistema
 - Servidores
 - Estações
 - Topologias de rede
 - Protocolos de comunicação, etc.

Diagrama de Implantação



Outros diagramas

- Diagrama de Pacotes
 - Representar os sub-sistemas englobados por um sistema
- Diagrama de Interação Geral
 - Fornece uma visão geral dentro de um sistema ou processo de negócios
- Diagrama de Temporização
 - Descreve a mudança no estado ou na condição de uma instância de uma classe ou seu papel durante um período de tempo

Diagrama de Casos de Uso

Interagindo com o Usuário

Diagrama de Casos de Uso

- Adota uma linguagem simples
 - Acessível ao cliente
- Objetivo é a compreensão do comportamento externo do sistema por qualquer *stakeholder*
- Apresenta o sistema através de perspectivas dos usuários

Diagrama de Casos de Uso

- Diagrama mais abstrato da UML
 - Portanto, o mais flexível e informal
- Geralmente é usado no início da modelagem do sistema
 - Especificação de requisitos
- Uma técnica de elicitação de requisitos

Diagrama de Casos de Uso

- Apresenta uma visão externa geral das funções e serviços do sistema
 - Define **o que** o sistema faz
 - Não se preocupa em **como** o sistema faz
- Um caso de uso indica uma funcionalidade que o sistema deve oferecer
 - Abrir conta, Sacar, Verificar Saldo, etc.

Componentes do Diagrama

- Atores
 - Quem executa a funcionalidade
- Casos de Uso
 - Qual é a funcionalidade
- Relacionamentos
 - Como atores e casos de uso se relacionam

Componentes do Diagrama

- **Atores**
 - **Quem executa a funcionalidade**
- Casos de Uso
 - Qual é a funcionalidade
- Relacionamentos
 - Como atores e casos de uso se relacionam

Atores

- Os atores representam os papéis desempenhados pelos diversos usuários
 - Cliente, Caixa do Banco, Gerente, etc.
- Casos de Uso descrevem interações entre o sistema e estes atores
- Atores podem ser
 - Pessoas que interagem com o sistema
 - Um hardware especial que dispara uma interação
 - Outro software que comunica com o sistema

Definição de Ator

- O ator é algo (usuário, software ou hardware) que não faz parte do sistema mas que interage com ele em algum momento
- Representação
 - Homem Palito + Papel Desempenhado

Exemplos de Atores



Componentes do Diagrama

- Atores
 - Quem executa a funcionalidade
- Casos de Uso
 - Qual é a funcionalidade
- Relacionamentos
 - Como atores e casos de uso se relacionam

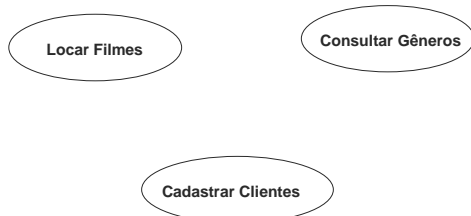
Casos de Uso

- Definem os serviços, tarefas ou funções do sistema
- Os nomes indicam ação (verbos)
 - Cadastrar venda : loja
 - Sacar : banco
 - Consultar um filme : locadora

Representação de Casos de Uso

- Representados por elipses
 - Um texto dentro descreve a funcionalidade do caso de uso
- Geralmente a descrição dentro da elipse é curta e direta
 - Verbo [+ Objeto]

Exemplos de Casos de Uso



Documentação de Casos de Uso

- Os diagramas devem ser explicados por meio de uma linguagem bastante simples
 - Linguagem Natural ou Estruturada
- Elementos que podem ser documentados
 - Quais atores interagem com o caso de uso
 - Quais etapas devem ser executadas
 - Quais parâmetros devem ser fornecidos
 - Quais são restrições e validações, etc.

Documentação de Casos de Uso

- Não existe um formato específico
- Várias formas são usadas, como:
 - Descrição passo a passo
 - Tabelas
 - Pseudo-código
 - Fluxograma
 - Cenários de Caso de Uso

Documentação de Casos de Uso

- Não existe um formato específico
- Várias formas são usadas, como:
 - Descrição passo a passo
 - Tabelas
 - Pseudo-código
 - Fluxograma
 - **Cenários de Caso de Uso**

Cenários

- Descreve uma situação de uso do sistema
- Inclui informações como
 - Nome do Cenário
 - Ator
 - Pré-condição
 - Fluxo normal
 - Fluxos alternativos
 - Pós-condição

Exemplo de Cenário (ATM)

- Nome do Cenário: Sacar Dinheiro
- Ator: Correntista
- Pré-condição: Conta e senha validada
- Fluxo normal
 1. Entrar com valor do saque
 2. Confirmar dados e operação
 3. Debitar valor da conta do cliente
- Fluxos alternativo: Saldo insuficiente
 - 3.1 Apresentar aviso ao cliente
- Pós-condição:
 - Valor sacado é debitado do saldo do cliente

Componentes do Diagrama

- Atores
 - Quem executa a funcionalidade
- Casos de Uso
 - Qual é a funcionalidade
- **Relacionamentos**
 - **Como atores e casos de uso se relacionam**

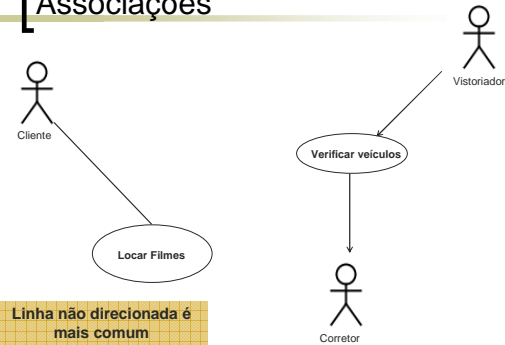
Relacionamentos

- Representam as interações ou associações entre
 - Atores e Casos de Uso
 - Dois ou mais Casos de Uso
 - Dois ou mais Atores
- Principais tipos de relacionamentos
 - Inclusão
 - Extensão
 - Generalização

Ator e Caso de Uso

- Demonstra que o Ator utiliza a função do sistema representada pelo Caso de Uso
 - Requistando a execução da função
 - Recebendo o resultado produzido pela função
- Representada por uma reta ligando o Ator ao Caso de Uso
 - Direcionada ou não

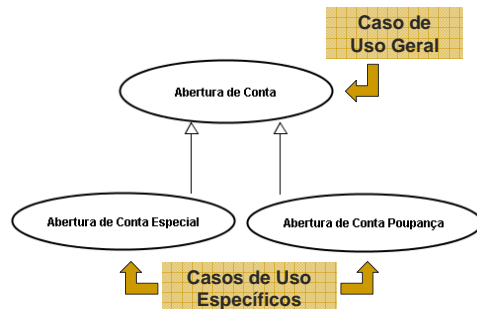
Associações



Especialização / Generalização

- Acontece quando dois ou mais casos de uso possuem características semelhantes
 - Reuso
- O Caso de Uso geral que descreve as características compartilhadas
- As especializações definem características específicas

Especialização / Generalização



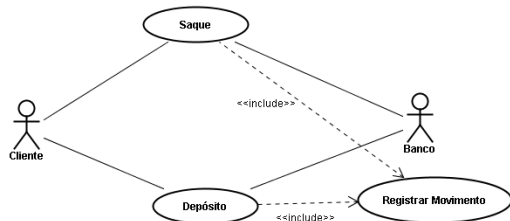
Inclusão (Include)

- Utilizado quando um caso de uso é usado dentro de outro caso de uso
- Os relacionamentos de inclusão indicam obrigatoriedade
 - A execução do primeiro obriga a execução do segundo

Inclusão

- Representada por uma seta tracejada
 - A seta aponta para o Caso de Uso incluído
- Possui a palavra "include" entre dois sinais de menor (<<) e dois sinais de maior (>>)

Inclusão



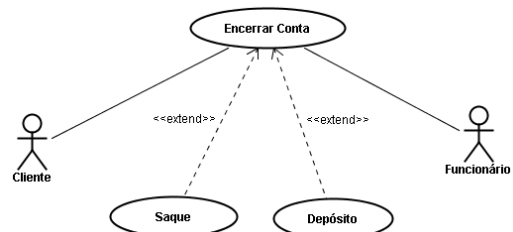
Extensão de Caso de Uso

- Geralmente usado em funcionalidades opcionais de um caso de uso
- Exemplo: cenários que somente acontecerão em uma situação específica
 - Se uma determinada situação for satisfeita
- Extensão pode necessitar um teste para determinar se o caso de uso será estendido

Representação de Extensão

- Semelhante à Inclusão
 - Possuir a palavra “*extend*” entre dois sinais de menor (<<) e dois sinais de maior (>>)

Extensão



Resumo da Aula

- UML
 - Linguagem de modelagem
 - É um padrão de fato
 - Define inúmeros diagramas
- Diagrama de Casos de Uso
 - Utilizado na Especificação de Requisitos
 - Usado em conjunto com a técnica de cenários para documentação

Referências da Aula

- BOOCH, G., RUMBAUGH, J., JACOBSON, I. **UML, Guia do Usuário**. Rio de Janeiro: Campus, 2000.
 - Capítulos 16 e 17
- M. Fowler. **UML Essencial**, 3a Edição. Bookmann, 2004.
 - Capítulo 9