

Introdução a UML

Cleidson de Souza
(Rodrigo Reis)

Agenda

- Definição
- Histórico
- Contribuições
- Diagramas
- Observações

Definição

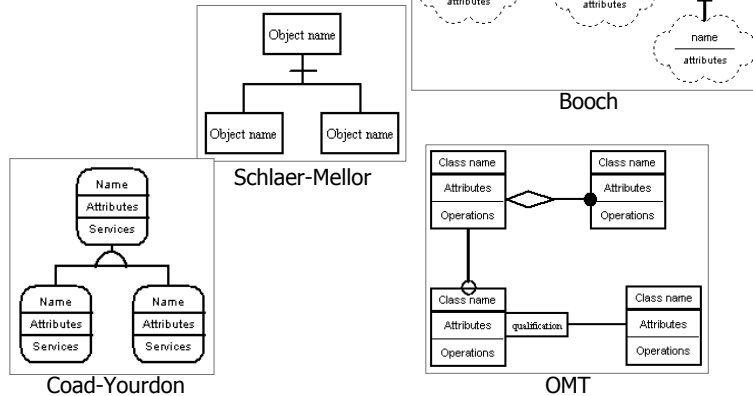
- A UML (*Unified Modeling Language*) é uma notação para descrição de sistemas orientados:
 - “The Unified Modeling Language for Object-Oriented Development” de Grady Booch, James Rumbaugh e Ivar Jacobson.
- Baseia-se na experiência dos principais autores dos 3 principais métodos OO.
- Esta notação foi padronizada pela OMG (*Object Management Group*) em 1997.

Histórico

- Metodologias para análise e projeto orientado a objetos (OOAD)
 - Evolução a partir de linguagens de programação
 - Anos 80-90: diversidade de autores
 - Anos 98-2000: unificação em torno de UML
- Até 1994, existiam pelo menos 50 métodos diferentes para desenvolvimento OO.

Histórico

- Exemplos
 - Classes



Histórico

- Os principais métodos utilizados eram os de Booch, Rumbaugh(*OMT*) e Jacobson(*Objectory*).
- Outros métodos importantes são o de Coad-Yourdon, Shlaer-Mellor e o método Fusão.

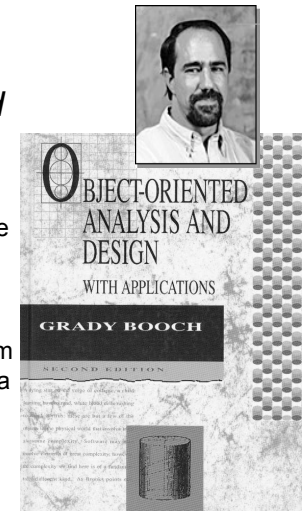
Histórico

- Grady Booch
 - Um dos pioneiros da OO
 - 1980: ênfase em técnicas de projeto para Ada
 - 1992-1994: livros
 - Object-Oriented Design with Applications*
 - projeto de programas em C++ e Ada



Histórico

- 1994: *Object-Oriented Analysis and Design with Applications*
 - texto sobre conceitos de OO e modelagem de objetos
 - projeto de várias aplicações-exemplo com diferentes linguagens da época
- 1998: Fundação da Rational



Histórico

- Ivar Jacobson
 - Modelagem OO baseado em Casos de Uso
 - *Objectory*



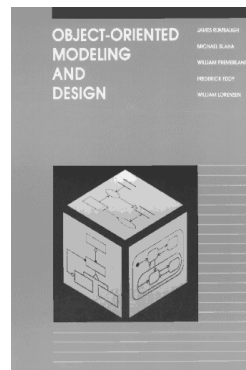
Histórico

- James Rumbaugh
 - Object Modeling Technique (OMT)
 - Desenvolvida na GE
 - Metodologia baseada em notações pré-existentes (ER, DTE, DFD)
 - Clara distinção entre as três visões do problema



Histórico

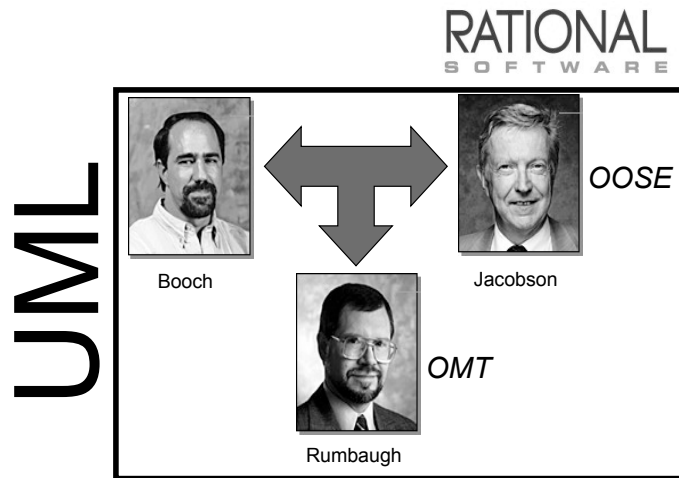
- James Rumbaugh (cont.)



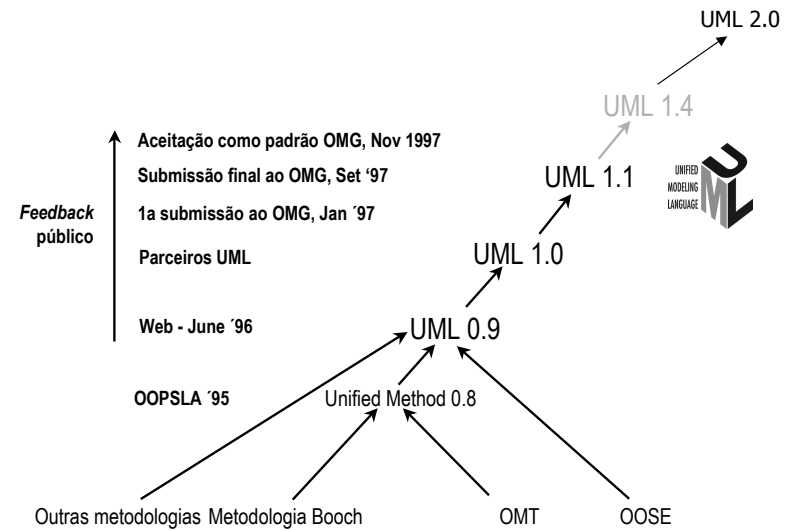
Histórico

- Em 1994, Rumbaugh e Booch decidiram terminar a “guerra” de métodos e se uniram visando criar um único método de desenvolvimento de software;
- A idéia era criar um “Método Unificado” que incorporasse as melhores características dos métodos existentes e resolvesse os problemas de cada um dos métodos;
- Em 1996, Jacobson se uniu e decidiu-se criar uma **linguagem de modelagem** unificada.

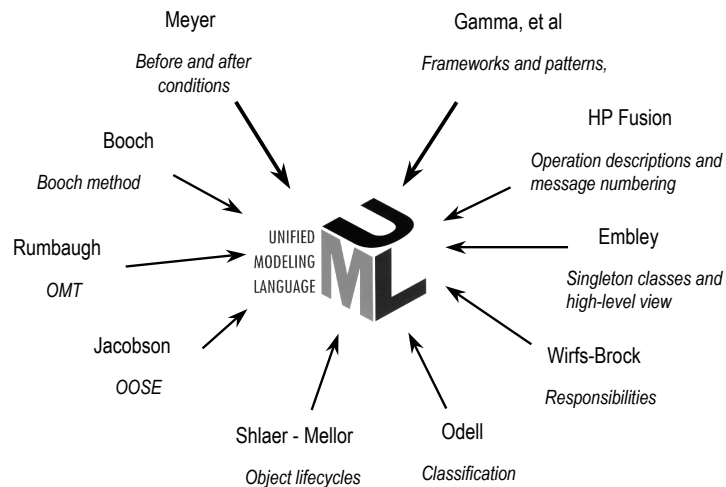
Histórico



Histórico



Contribuições



Contribuições



UML

- O que é UML
 - Linguagem visual para especificação (modelagem) de sistemas orientados a objetos
 - Fornece representação gráfica para os elementos essenciais do paradigma de objetos
 - Classes, atributos, objetos, troca de mensagens, ...

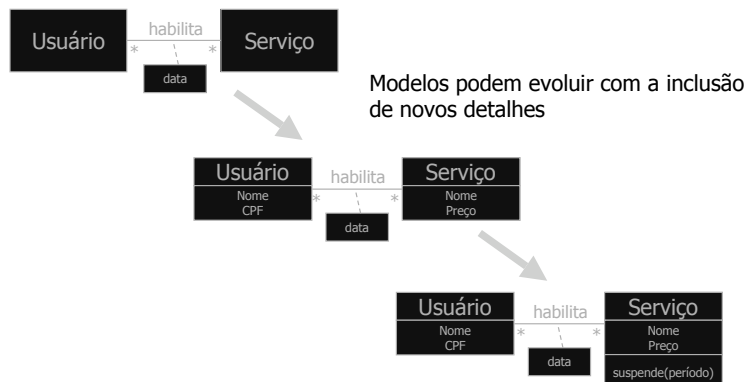


UML

- O que é UML
 - De propósito geral
 - Não está presa a uma etapa do desenvolvimento de software
 - Análise
 - Projeto
 - Implementação
 - Testes
 - Não está presa a um processo
 - Ciclo de vida em cascata
 - Incremental
 - Processo Unificado
 - ...
 - Não está presa a uma linguagem de programação

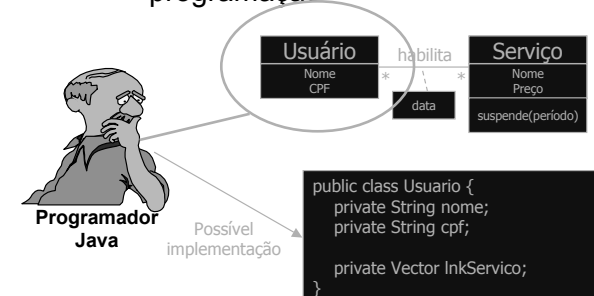
UML

- UML apóia o desenvolvimento incremental



UML

- O que é UML
 - De propósito geral
 - Não está presa a uma linguagem de programação



UML

- O que é UML
 - Padrão OMG
 - Em <http://www.omg.org> estão disponíveis documentos eletrônicos que contém
 - Sumário da UML
 - Semântica
 - Guia da Notação
 - Extensões da Linguagem

UML

- O que é UML
 - Privilegia a descrição de um sistema segundo três perspectivas:
 - **Dados (estrutural)**
 - Diagrama de Classes
 - **Operações (funcional)**
 - Diagrama de Caso de Uso
 - **Eventos (temporal)**
 - Diagramas de Seqüência, Atividades, de Transição de Estados



Ferramentas CASE para UML

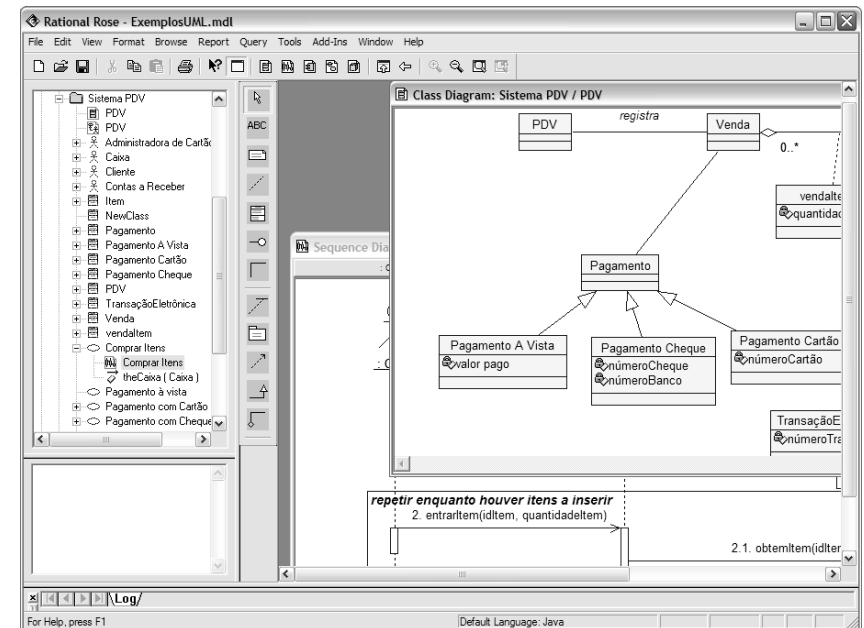
- Grande variedade de ferramentas com diferentes recursos
 - Mais conhecidas
 - Rational Rose
 - <http://www.rational.com>
 - Omondo
 - <http://www.omondo.com>
 - Plug-in para o Eclipse
 - TogetherSoft ControlCenter (Java)
 - <http://www.togethersoft.com>

Ferramentas CASE para UML

- Grande variedade de ferramentas com diferentes recursos (continuação)
 - *Open Source*
 - Argo UML (Java)
 - <http://argouml.tigris.org>
 - Fujaba (Java)
 - <http://www.fujaba.de>
 - *Gratuito (com alguma limitação)*
 - Poseidon (Java)
 - <http://www.gentleware.com>
 - Visual Paradigm
 - <http://www.visual-paradigm.com/>
 - *Delphi*
 - <http://www.sparxsystems.com.au/>

Ferramentas CASE para UML

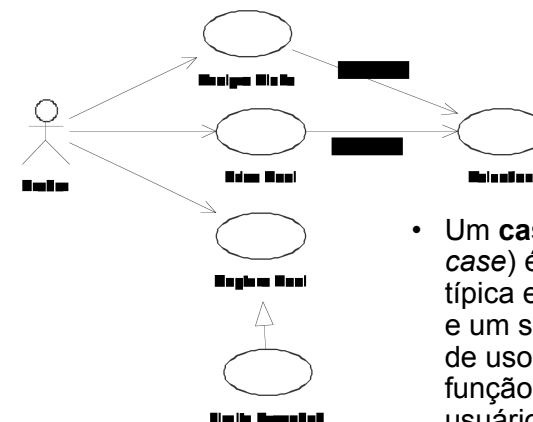
- Observações:
 - Algumas ferramentas assumem “liberalidades artísticas” para representar os diagramas de maneira diferente do previsto na linguagem
 - XML é o padrão cada vez mais adotado usado para intercâmbio de modelos feitos por diferentes ferramentas



Diagramas da UML

- Casos de Uso (OOSE)
- Diagramas de Classe (OMT, Booch e OOSE)
- *Deployment* Diagramas (Booch)
- Diagramas de Interação
 - Diagramas de Sequência (OMT e Booch)
 - Diagramas de Colaboração (Booch)
- Diagramas de Pacotes (Booch)
- Diagramas de Transição de Estados (OMT e Booch)
- Diagramas de Atividade

Diagrama de Casos de Uso



- Um **caso de uso** (*use case*) é uma interação típica entre um usuário e um sistema. Um caso de uso captura alguma função visível ao usuário e, em especial, busca atingir uma meta do usuário.

Diagramas de Classes

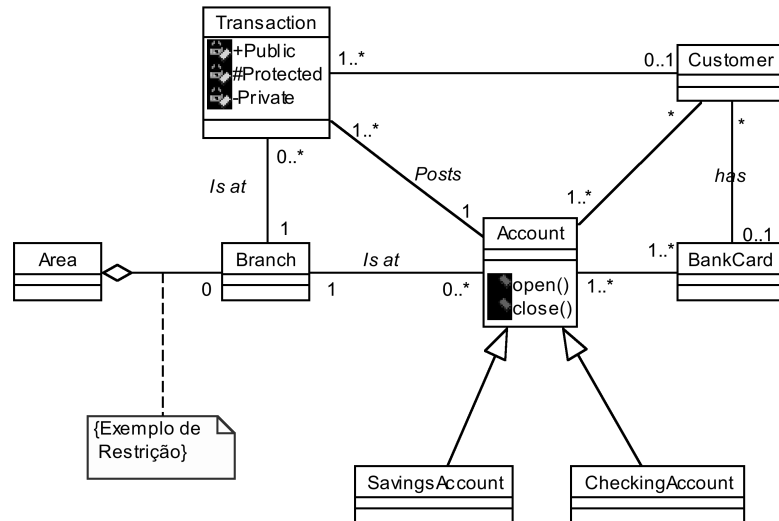


Diagrama de Transição de Estados

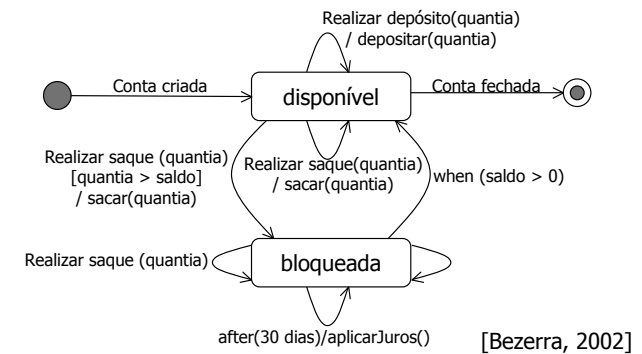


Diagrama de Atividades

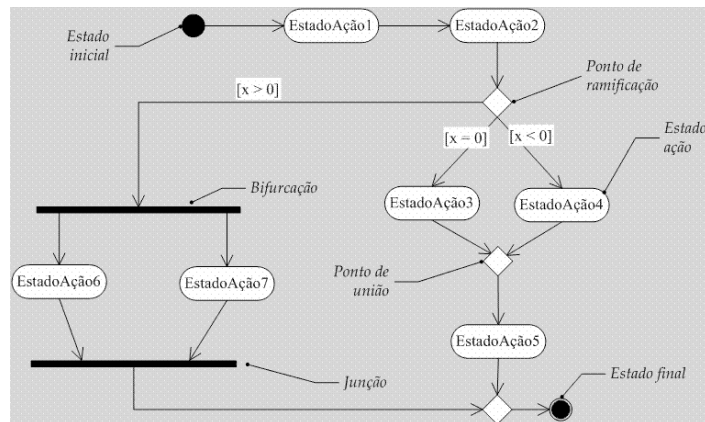


Diagrama de Seqüência

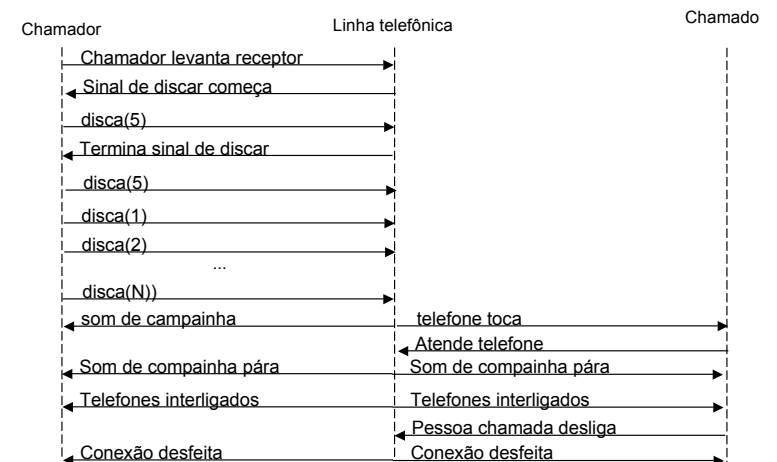
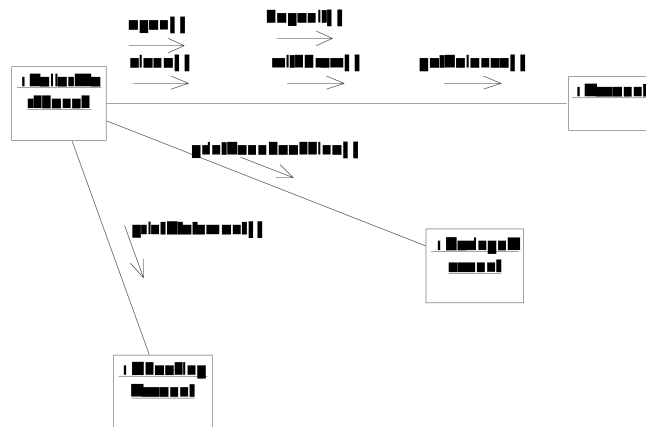
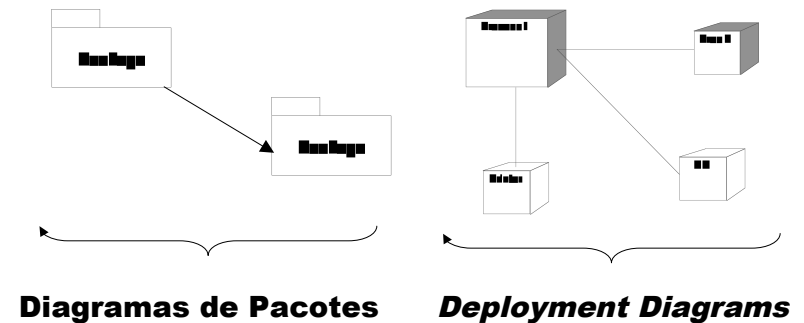


Diagrama de Colaboração



Outros Diagramas



Observações

- Ela é uma linguagem de modelagem porque não apresenta um processo para o desenvolvimento de software;
- Como a UML é apenas uma notação, ela pode ser usada com outras metodologias:
 - Processo da OMT e notação UML.
- A UML possui diversos mecanismos de extensão que permitem que ela possa ser utilizada em diversos domínios diferentes.

Observações

- Em 1999, os mesmos autores da UML lançaram a proposta de um novo método de desenvolvimento de SW chamado Processo Unificado que tenta integrar as 3 metodologias, assim como outras metodologias de OOAD;
- Uma abordagem similar foi proposta pelo método *Fusion* (Fusão).

Vantagens

- Define um mapeamento da análise ao projeto e à implementação
- Define uma notação expressiva e consistente
 - Facilita a comunicação entre as pessoas;
 - Ajuda a apontar inconsistências e omissões;
 - Suporta a análise e projeto de sistemas de pequeno e grande porte.

Leituras

- Booch, Grady. "UML in Action", Communications of the ACM, vol 42, n 10, Oct 1999.
- Kobryn, Cris. "UML 2001: A Standardization Odyssey", Communications of the ACM, vol 42, n 10, Oct 1999.