Visão Geral da UML

SSC 121 - Engenharia de Software I

Profa. Dra. Elisa Yumi Nakagawa

2º semestre de 2012

Conteúdo

- Introdução
- Ferramentas de Apoio
- Diagramas da UML
- Elementos Genéricos
- Material sobre UML

Introdução: Modelo de Software

- Engenheiros de software projetam modelos do software a ser construído.
- Um modelo de software:
  - representa, simplificadamente, o que se pretende construir.
  - permite controlar a qualidade do sistema a ser desenvolvido.
  - permite que o projetista, o construtor e o cliente tenham a mesma visão do problema.
  - pode ser testado e simulado.

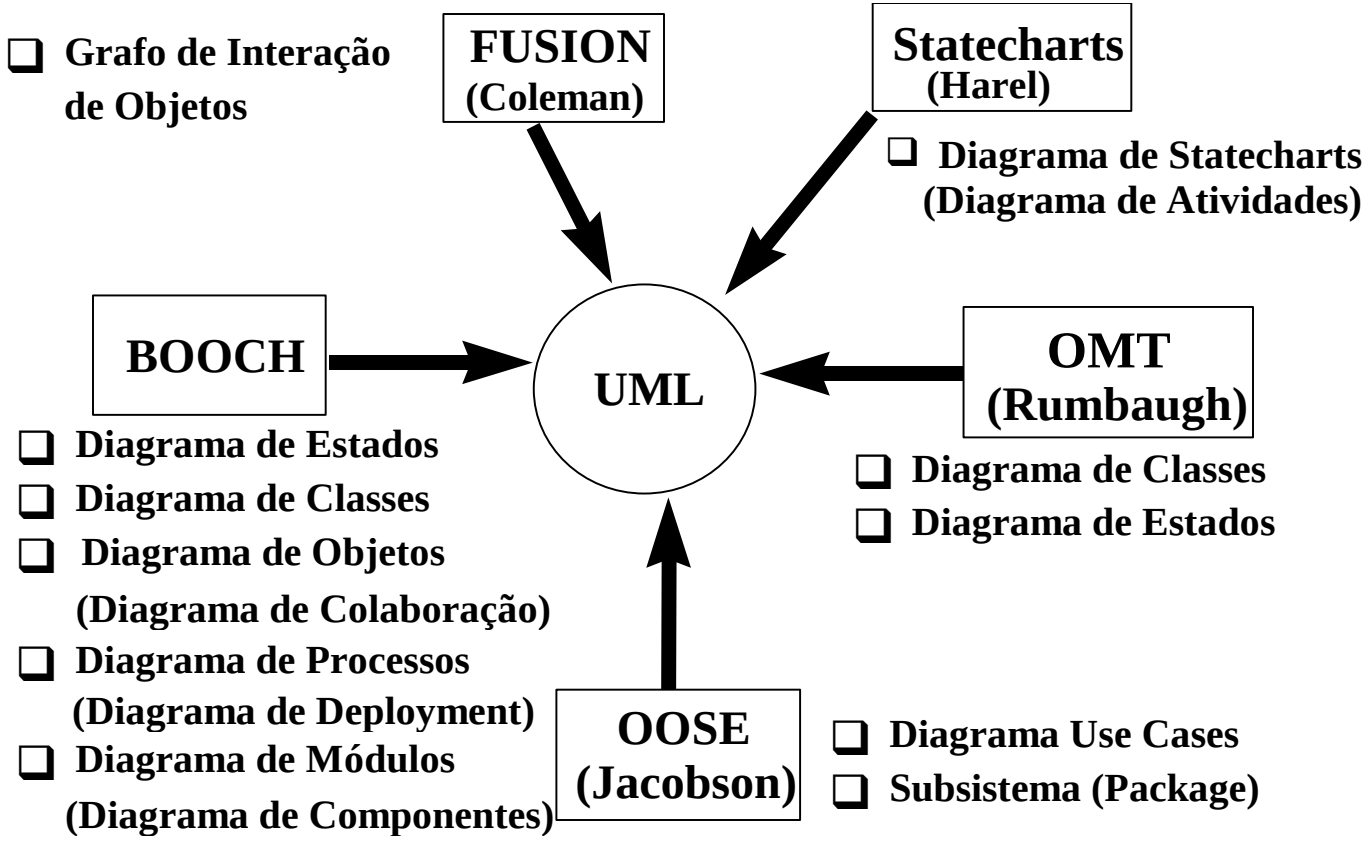
Introdução: Histórico

- Início em Outubro de 1994, Booch e Jim Rumbaugh começaram um esforço para unificar o método de Booch e OMT (*Object Modeling Language*).
- Uma primeira versão, chamada *Unified Method*, foi divulgada em outubro de 1995.
- Jacobson juntou-se ao grupo, agregando o método OOSE (*Object-Oriented Software Engineering*) .
- O esforço dos três resultou na liberação da UML versão 0.9 e 0.91 em junho e outubro de 1996.
- Em janeiro de 1997, foi liberada a versão 1.0 da UML.
- Adotada como padrão segundo a OMG (*Object Management Group,* http://www.omg.org/) em Novembro de 1997
- Versão atual 2.0

Introdução: UML

- UML *(Unified Modelling Language)*
- A UML é uma linguagem para especificação, construção, visualização e documentação de sistemas. A fusão agregou os pontos fortes de cada uma destas metodologias.
- A UML é uma evolução das linguagens para especificação dos conceitos de *Booch*, OMT e OOSE e também de outros métodos de especificação de requisitos de software orientados a objetos ou não.

Introdução: UML



Fonte: Material didático do Prof. Antonio Francisco do Prado (UFSCar)

Ferramentas de Apoio

- Diversas empresas lançaram ferramentas para auxiliar a modelagem e projeto de sistemas utilizando UML, gerar código a partir da modelagem e projeto e realizar engenharia reversa, ou seja, obter o modelo em UML a partir do código.
- Exemplos:
  - A família Rational Rose Interprise (da *Rational Software Corporation* *http://www-306.ibm.com/software/rational/*) que gera código para SMALLTALK, PowerBuilder, C++, J++, Visual Basic.
  - *http://www.omg.org/technology/uml/index.htm#Links-Tools* (lista de ferramentas que envolvem a UML)

Diagramas da UML

- Diagramas de Casos de Uso
- Diagramas de Classe
- Diagramas de Comportamento
  - Diagrama de Estado
  - Diagrama de Atividade
  - Diagrama de Seqüência
  - Diagrama de Colaboração
- Diagramas de Implementação
  - Diagrama de Componente
  - Diagrama de Disponibilidade *(Deployment)*

História da Semântica e da Notação

- Diagrama de caso de uso - similar aos do método OOSE
- Diagrama de classes - fusão do OMT, Booch e maioria dos métodos OO, com novos elementos
- Diagrama de estado - statechart de Harel
- Diagrama de atividade - possui a mesma semântica do diagrama de estado e é parecido com diagramas de *workflow*
- Diagrama de seqüência - são encontrados em diversos métodos sob muitos nomes (interação, eventos, mensagem)
- Diagrama de colaboração - foi adaptado do Booch e Fusion
- Diagramas de implementação (diagrama de componente e diagrama de disponibilidade) - são derivados dos diagramas de módulo e processo do método Booch

A UML aborda:

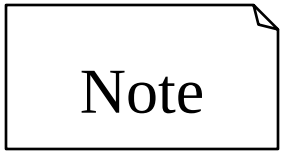
- Comportamento interno do software: Modela a resposta aos estímulos externos e a comunicação entre as partes internas:
  - Diagramas de Estados
  - Diagramas de Interação
- Comportamento externo do software: Descrição de cenários de interação entre elementos externos e o sistema:
  - Diagramas de Casos de Usos

A UML aborda:

- Arquitetura de implementação: Descrição dos componentes de software que formam o sistema e a arquitetura de hardware:
  - Diagramas de Componentes
  - Diagramas de Distribuição
- Estrutura de suporte: Estrutura das partes que formam o sistema e suas relações internas:
  - Diagramas de Classes
  - Diagramas de Pacotes

Elementos Genéricos: Note e Constraints

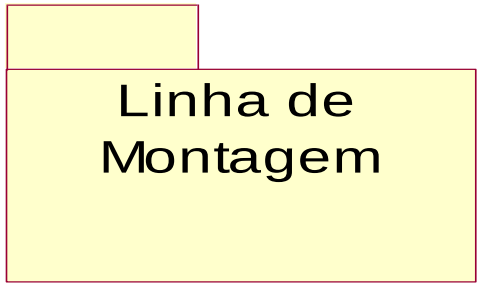
- *Note*: é um comentário inserido no diagrama

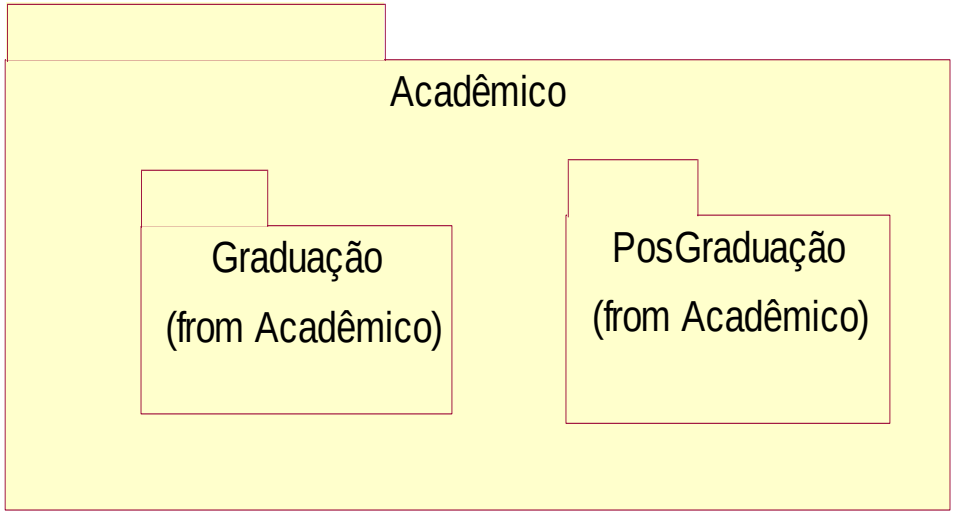


- *Constraint (restrições)*: é uma relação semântica entre elementos do modelo. Especifica condições ou proposições que devem ser mantidas verdadeiras. Uma restrição é mostrada como uma cadeia entre chaves {}. (OCL – Object Constraint Language)

Elementos Genéricos: Packages

- Packages agrupam elementos de modelagem.
- Packages podem conter classes, relacionamentos, classes abstratas, Packages, tipos, etc...





Elementos Genéricos: Estereótipos (“Stereotypes”)

- Mecanismo de extensão introduzido pela UML, que permite estender o meta-modelo para suprir necessidades que não encontram-se entre os meta-elementos disponíveis, desde que a definição semântica da própria linguagem não seja ferida.
- Deve ser baseado em certas classes existentes no meta-modelo e deve estender estas classes apenas em certas formas pré-definidas, porém esses limites não são claramente especificados.
- Um estereótipo pode ser aplicado a classes, relacionamentos de dependência, atributos, operações, etc.
  - Ex: <<abstract>>, <<metaclass>>

Material sobre UML

- http://www-01.ibm.com/software/rational/ (IBM)
- http://www.omg.org (*Object Management Group*)