

Introdução à UML

Prof. Fernando Maia da Mota

Slides gentilmente cedidos por Profa. Dra. Maria Istela Cagnin Machado
UFMS/FACOM

Introdução

- Necessidade de abordagens para desenvolver software de maneira organizada e estruturada
 - Desenvolvimento Estruturado
 - □ Desenvolvimento OO
 - Desenvolvimento baseado em Componentes
 - ...

Introdução

- Análise: descobrir o enunciado do problema (faça a coisa certa)
- Projeto: resolver o problema (faça certo a coisa)
- Análise OO: ênfase em encontrar e descrever os objetos – ou conceitos – do domínio do problema. Ex: domínio: sistema de informação de vôo - avião, vôo e piloto (conceitos)
- Projeto OO: ênfase na definição dos objetos de software e como eles colaboram para a satisfação dos requisitos. Ex: objeto de software avião possui o atributo `numDaCauda` e o método `obterHistoricoVoo()`

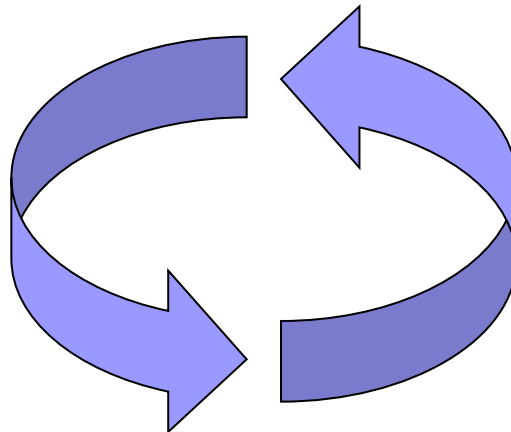


Introdução

Investigar o problema não significa ficar meses fazendo diagramas. O método de análise e projeto que vamos ver na disciplina é **cíclico**

Introdução

- Quando se trabalha com ciclos iterativos, a análise, o projeto e a implementação são realizados sequencialmente em ciclos curtos, produzindo partes do sistema implementadas em sua forma final desde muito cedo





Introdução

- A utilização de técnicas sistemáticas nas fases de análise e projeto faz com que o processo de geração de código seja quase automático
- Cabe ao programador dominar as características específicas das linguagens, ferramentas e frameworks para adaptar o código gerado aos requisitos, quando necessário

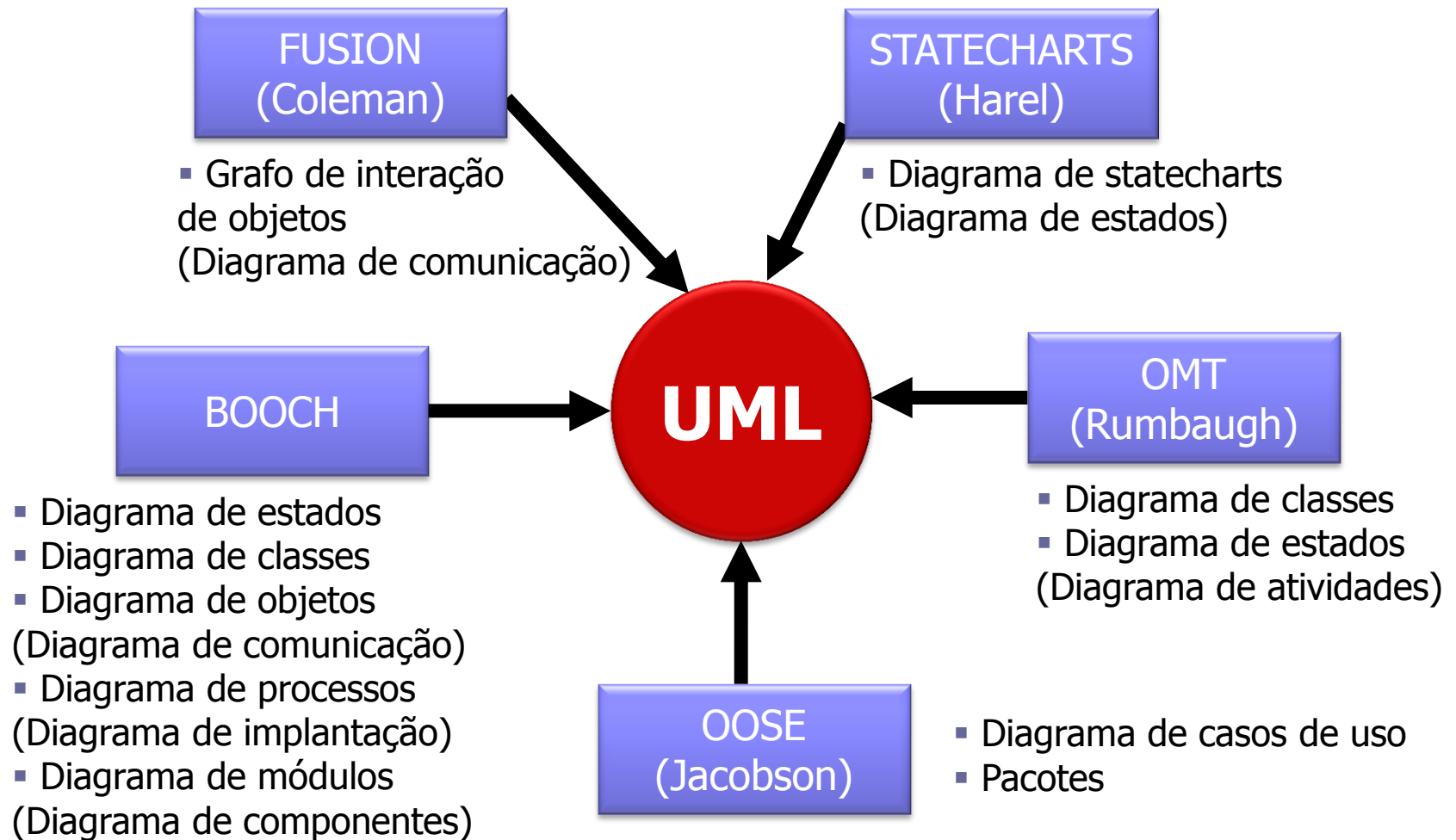
UML: Pontos chave

- UML (*Unified Modelling Language*)
- É uma linguagem para especificação, construção, visualização e documentação de sistemas
- Não é uma metodologia
- É uma evolução de outras linguagens para especificação de requisitos de software orientados a objetos ou não

UML: Pontos chave

- Conhecer uma linguagem de especificação não implica habilidade de saber usá-la para produzir artefatos úteis
- Existe algo por trás da linguagem denominado **método** ou **processo**, que auxilia os desenvolvedores a colocar os elementos da linguagem na ordem e na estrutura adequadas

Histórico da UML





Ferramentas de Apoio

- CASE significa “Computer-Aided Software Engineering”
 - “Engenharia de Software Auxiliada por Computador”
- É um aplicativo que auxilia os profissionais envolvidos na tarefa de produzir sistemas
- O tipo de “ajuda” que a ferramenta fornece, depende exclusivamente da proposta do fabricante
 - ferramentas de análise, projeto e implementação



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
 - realizar engenharia reversa, ou seja, obter o modelo em UML a partir do código

Ferramentas de Apoio

■ Exemplos:

- A família Rational Rose Interprise (www.rational.com) que gera código em Smalltalk, PowerBuilder, C++, J++ e VB
- ArgoUML- free <http://argouml.tigris.org/>
- www.objectsbydesign.com/tools/umltools_byCompany.html (lista de ferramentas que envolvem a UML)
- MVCase: Desenvolvida por pesquisadores da UFSCAR. Disponível em <https://mvcase.dev.java.net/>
- Visual Paradigm of UML - <http://www.visual-paradigm.com/>
- Astah: <http://astah.change-vision.com/en/index.html>
- StarUML: <http://staruml.sourceforge.net/en/>



Famílias de Diagramas UML

- Diagramas estruturais
 - Diagramas de pacotes, classes, objetos, estrutura composta, componentes e distribuição.
- Diagramas comportamentais
 - Diagramas de casos de uso, atividades e máquina de estados.
- Diagramas de interação
 - Diagramas de comunicação, sequência, tempo e visão geral de integração.




Preciso utilizar todos os diagramas da UML?

Família de Diagramas da UML

- Diagramas estruturais
 - Diagramas de pacotes, **de classes**, de objetos, de estrutura composta, de componentes e de distribuição
- Diagramas comportamentais
 - **Diagramas de casos de uso**, **de atividades** e **de estados**
- Diagramas de interação
 - Diagramas de comunicação, **sequência**, tempo e visão geral de integração

**Três a cinco diagramas podem representar
a essência de um sistema**

- 
- Nem todos os diagramas precisam ser usados durante o desenvolvimento de um sistema
 - Usam-se apenas aqueles que possam apresentar alguma informação útil para o processo

Material sobre UML

- <http://www.rational.com> (Rational)
- <http://www.omg.org> (*Object Management Group*)
- Page-Jones, M.; Fundamentos do desenho orientado a objeto com UML, Makron Books, 2001.
- Furlan, J. D.; Modelagem de Objetos Através da UML, Makron Books, 1998.
- Rumbaugh, J., Jacobson, I., Booch, G.; The Unified Modeling Language Reference Manual, Addison-Wesley, c1999.
- Conallen, J.; Building Web Applications with UML, Addison-Wesley, 1999.
- Fowler, M.; Scott, K.; UML Essencial, Bookman, 2000.