

Introdução a UML



O que é a UML?

- Linguagem Gráfica de Modelagem para:
 - Visualizar
 - Especificar
 - Construir
 - Documentar
 - ComunicarArtefatos de sistemas complexos
- Linguagem: vocabulário + regras de combinação

Modelos

- O que é um modelo?
 - Um modelo é uma simplificação (representação) da realidade
- O que modelamos?
 - Dimensões: dados, função, comportamento

Objetivos da Modelagem

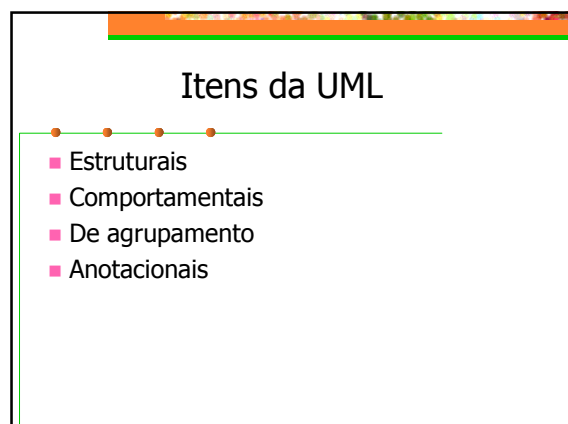
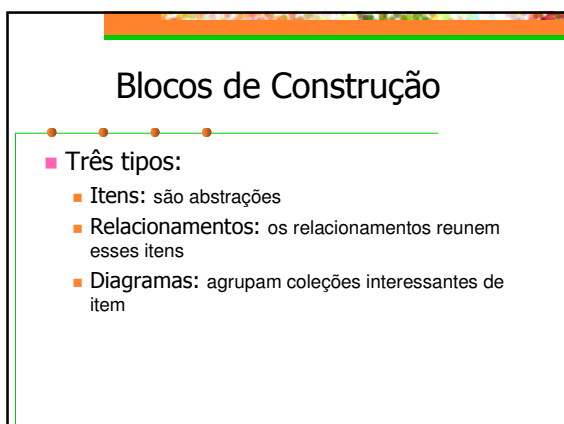
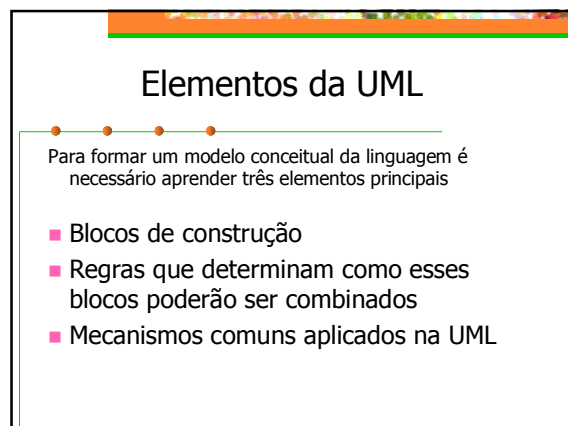
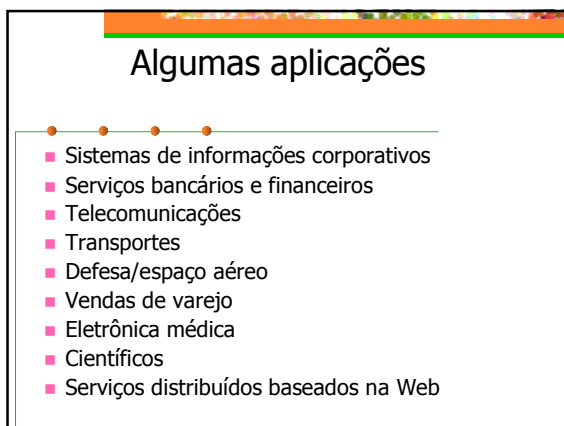
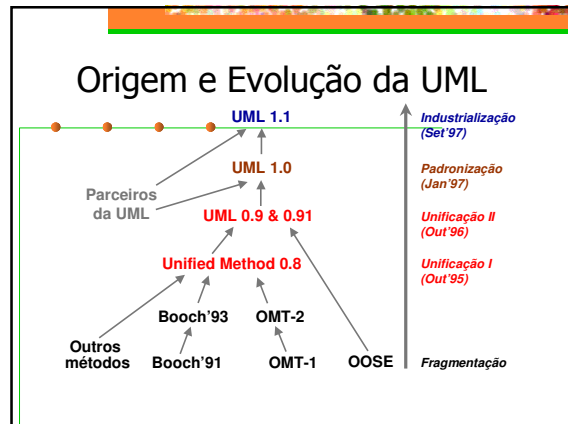
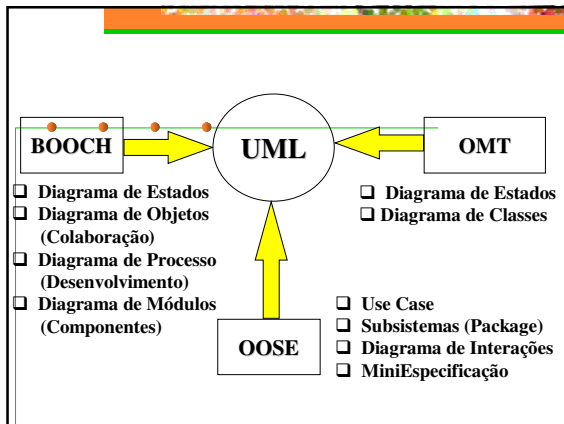
- Compreender melhor o sistema que estamos desenvolvendo
- Visualizar o sistema
- Documentar decisões tomadas
- Especificar comportamento ou a estrutura de um sistema

Princípios da Modelagem

- A escolha dos modelos a serem criados tem profunda influência sobre a maneira como um determinado problema é atacado e como uma solução é definida
- Cada modelo poderá ser expresso em diferentes níveis de precisão
- Os melhores modelos estão relacionados à realidade
- Nenhum modelo único é suficiente. Qualquer modelo não-trivial será melhor investigado por meio de um pequeno conjunto de modelos quase independentes

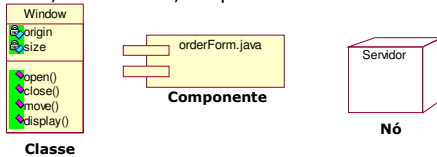
A UML não é

- um processo
- uma metodologia
- Análise e Projeto OO
- regras de projeto



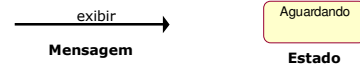
Itens estruturais

- São os substantivos dos modelos. São a parte estática, representando elementos conceituais ou físicos
- Sete tipos: classes, interfaces, colaborações, casos de uso, classes ativas, componentes e nós



Itens comportamentais

- Representam as partes dinâmicas dos modelos. São os verbos, representando comportamentos no tempo e no espaço
- Dois tipos: interação e máquina de estado



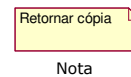
Itens de agrupamento

- São as partes organizacionais dos modelos de UML. São os blocos em que os modelos podem ser decompostos – pacotes
- Um pacote é um mecanismo de propósito geral para a organização de elementos em grupos



Itens anotacionais

- Partes explicativas dos modelos UML. São comentários, incluídos para descrever, esclarecer e fazer alguma observação importante sobre qualquer elemento do modelo - notas

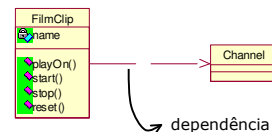


Relacionamentos

- Dependência
- Associação
- Generalização
- Realização

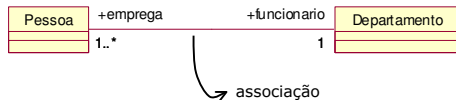
Dependência

- Relacionamento semântico entre dois itens, nos quais a alteração de um (o item independente) pode afetar a semântica do outro (o item dependente)



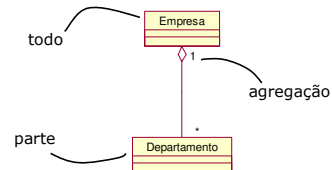
Associação

- É um relacionamento estrutural que descreve um conjunto de ligações, em que as ligações são conexões entre objetos



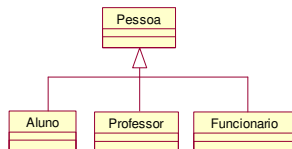
(Agregação)

- A agregação é um tipo especial de associação representando um relacionamento estrutural entre o todo e sua parte



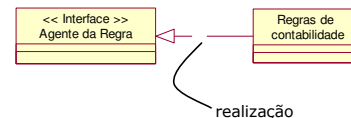
Generalização

- É um relacionamento de especialização/generalização, nos quais os objetos dos elementos especializados (os filhos) são substituíveis por objetos do elemento generalizado (os pais)



Realização

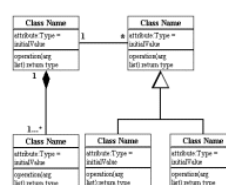
- É um relacionamento semântico entre classificadores, em que um classificador especifica um contrato que outro classificador garante executar



Diagramas

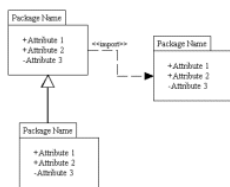
- Apresentações gráficas de um conjunto de elementos, geralmente representadas como gráficos de vértices (itens) e arcs (relacionamentos)
- Nove tipos: classes, objetos, pacotes, casos de uso, seqüências, colaborações, estados, atividades, componentes e implantação

Diagramas de classes



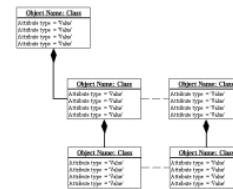
- Diagramas de classe são a espinha dorsal da maioria dos métodos orientados a objeto, inclusive UML
- Descrevem a estrutura estática do sistema (entidades e relacionamentos)

Diagramas de pacotes



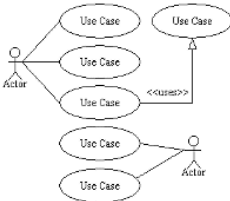
- Organizam elementos do sistema em grupos relacionados a fim de minimizar a dependência entre eles

Diagramas de objetos



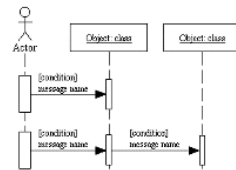
- Descrevem a estrutura estática de um sistema em um determinado momento
- Podem ser usados para testar a precisão dos diagramas de classe

Diagramas de casos de uso



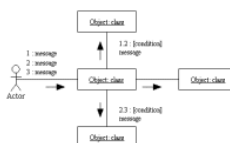
- Modelam a funcionalidade do sistema através de atores e casos de uso
- Casos de uso são serviços ou funções fornecidas pelo sistema aos seus usuários

Diagramas de seqüências



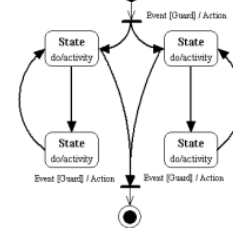
- Descreve as interações entre as classes através das trocas de mensagens ao longo do tempo

Diagramas de colaborações



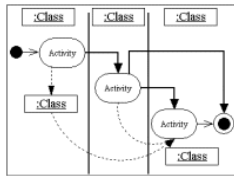
- Representam as interações entre objetos em termos de mensagens em seqüência
- Descrevem tanto a estrutura estática como o comportamento dinâmico do sistema

Diagramas de estados



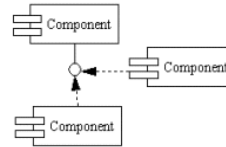
- Descrevem o comportamento dinâmico do sistema em resposta a estímulos externos
- São especialmente úteis para modelar objetos reativos cujos estados são disparados por eventos específicos

Diagramas de atividades



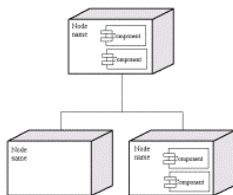
- Ilustram a natureza dinâmica de um sistema modelando o fluxo de controle de uma atividade para outra
- Uma atividade representa uma operação em uma classe do sistema que resulta na mudança do estado do sistema
- Tipicamente, são usados para modelar fluxo de trabalho ou processos de negócio e funcionamento interno

Diagramas de componente



- Descreve a organização dos componentes físicos de software
- Ex.: código-fonte, código em tempo de execução (binário) e executáveis

Diagramas de implantação



- Descrevem os recursos físicos em um sistema, incluindo nós, componentes e conexões

Regras da UML

- Especificam o que deverá ser um *modelo bem-formado*
- *Modelos bem-formados* são aqueles autoconsistentes semanticamente e em harmonia com todos os modelos a ele relacionados
- Regras para: nome, escopo, visibilidade, integridade e execução
- Mais modelos: parciais, incompletos e inconsistentes

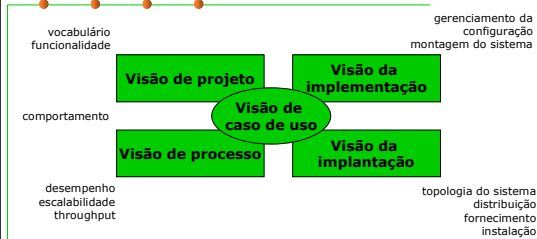
Mecanismos básicos da UML

- Especificações
- Adornos
- Divisões comuns
- Mecanismos de extensão

Arquitetura

- Decisões significativas acerca de:
 - A organização do sistema de software
 - A seleção dos elementos estruturais e suas interfaces
 - Seu comportamento, conforme especificado nas colaborações entre esses elementos
 - A composição desses elementos estruturais e comportamentais em subsistemas cada vez maiores
 - O estilo de arquitetura que orienta a organização: os elementos estáticos e dinâmicos e suas respectivas interfaces, colaborações e composição

Modelagem da arquitetura



Referências

- Boock, G. and Rumbaugh, J. *The Unified Modeling Language User Guide*. Addison-Wesley, 1999
- Arlow, J. and Neustadt, I. *UML 2 and the Unified Process: Practical Object-Oriented Analysis and Design*, 2nd Edition, The Addison-Wesley Object Technology Series, 2005.
- Rumbaugh, J.; Jacobson, I. and Boock, G. *The Unified Modeling Language Reference Manual*, 2nd Edition, The Addison-Wesley Object Technology Series, 2004.
- Boock, G.; Rumbaugh, J. and Jacobson, I. *Unified Modeling Language User Guide*, 2nd Edition, The Addison-Wesley Object Technology Series, 2005.
- Jacobson, I.; Boock, G. and Rumbaugh, J., *Unified Software Development Process*, Addison-Wesley, Janeiro 1999.
- Larman, C. *Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design* Prentice-Hall, New Jersey - USA, 1997