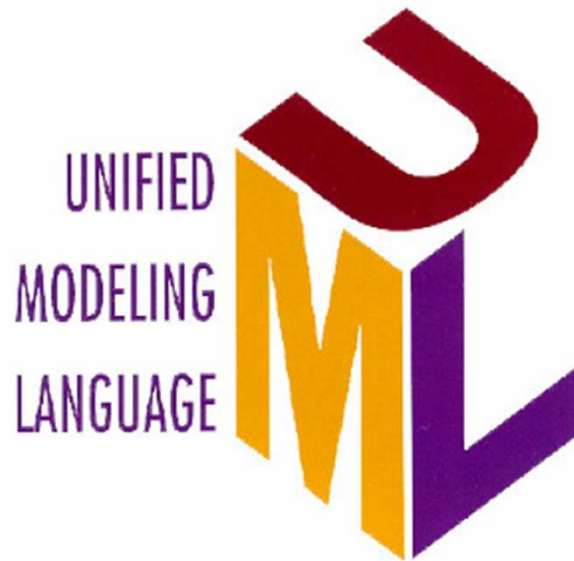


Modelagem e Desenvolvimento de Software

Introdução a UML



Material adaptado do Professor Edson Marchetti

UML - Necessidade de um Padrão

Esforços no sentido de padronização liderado pelos “três amigos”.



James Rumbaugh
General Electric



Ivar Jacobson
Objectory



Grady Booch
Rational Software Corporation

UML - Unified Modeling Language

O que é?

- ◆ Uma linguagem de modelagem unificada que rapidamente se tornou um padrão para a construção de software orientado a objetos.
- ◆ Padronizada pela Object Management Group.
(www.omg.org)



“Curiosamente, muitas empresas de desenvolvimento de software começam querendo construir prédios altos, como se estivessem fazendo uma casinha de cachorro”
(BOOCH, RUMBAUGH, JACOBSON, 2005
pag.5



UML - Unified Modeling Language

Segundo a especificação da OMG:

- ◆ “UML é uma linguagem gráfica para visualizar, especificar, construir e documentar artefatos de sistemas complexos software.
- ◆ UML oferece um modo padrão descrever um sistema, incluindo tanto coisas conceituais, tais como modelos de processo e funções de sistema, como coisas concretas, tais como declarações de linguagens de programação, esquemas de bancos de dados e componentes reusáveis de software”.

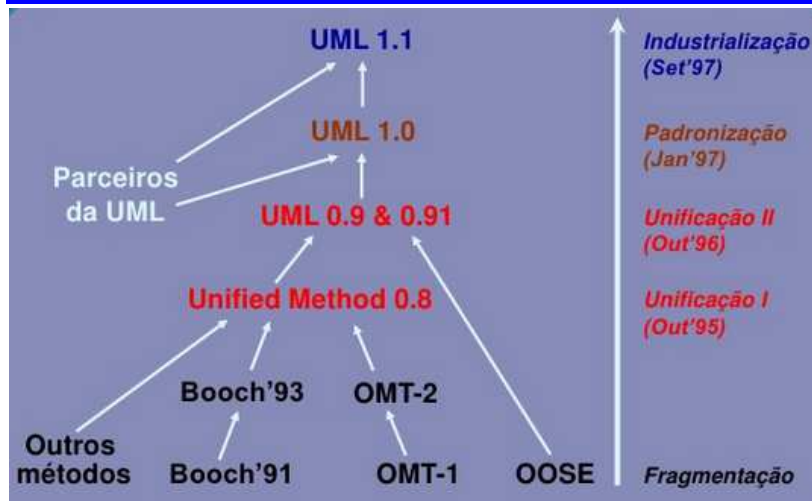
What is UML

"The Unified Modeling Language (UML) is a graphical language for visualizing, specifying, constructing, and documenting the artifacts of a software-intensive system.

The UML offers a standard way to write a system's blueprints¹, including conceptual things such as business processes and system functions as well as concrete things such as programming language statements, database schemas, and reusable software components."

Fonte: <http://www.sparxsystems.com/uml-tutorial.html>

UML – Histórico



1980

- ◆ Grady Booch. Object Oriented Design;

1991

- ◆ Jim Rumbaugh. Criou o modelo de Objetos OMT;

1992

- ◆ Ivar Jacobson. OOSE (Oriented-Object Software Engeneering);

1994

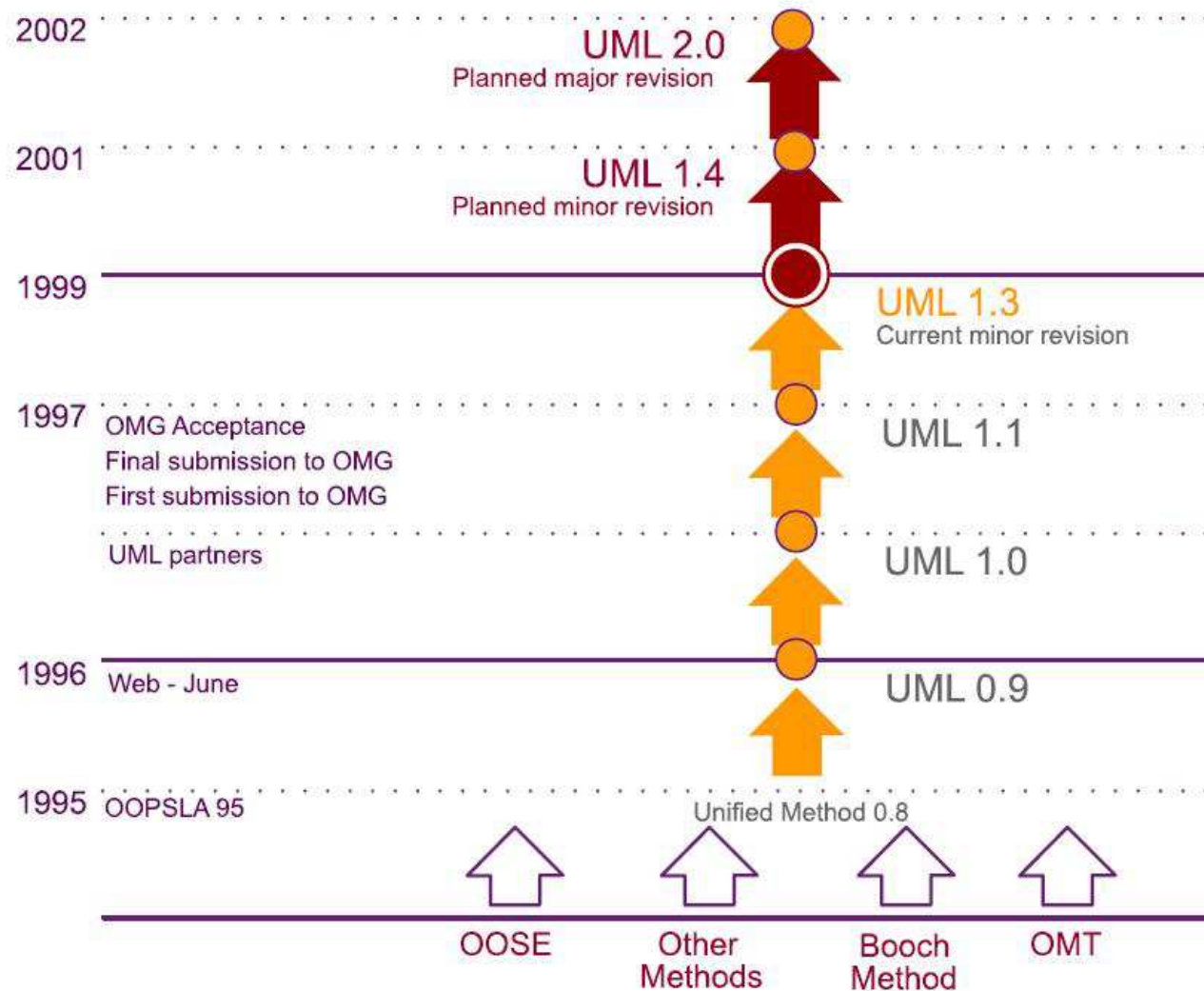
- ◆ 72 métodos disponíveis no mercado!

1999

- ◆ Booch, Rumbaugh e, posteriormente, Jacobson criaram a UML e a submeteram à OMG;

UML – Histórico

2013 - A UML encontra-se atualmente na versão 2.5



UML – Finalidade

- ◆ A UML define um número de diagramas que permitem dirigir o foco para aspectos diferentes do sistema de maneira independente.
- ◆ Facilita a comunicação de todas as pessoas envolvidas no processo de desenvolvimento de um sistema – gerentes, coordenadores, analistas, desenvolvedores – por apresentar um vocabulário de fácil entendimento.

UML Estrutura

- UML 2.0 define 13 diagramas básicos, distribuídos em dois conjuntos gerais:
 - ◆ Diagrama de Modelagem Estrutural:
 - ☞ Definem a arquitetura estática de um modelo;
 - ☞ Descrevem “coisas” que fazem parte de um modelo (classes, objetos, interfaces e componentes físicos);
 - ☞ Modelam a relação e interdependência entre elementos;
 - ◆ Diagramas de Modelagem Comportamental:
 - ☞ Capturam as variedades de interação e estados instantâneos em um modelo, à medida que este “executa” no tempo;

UML - Diagramas

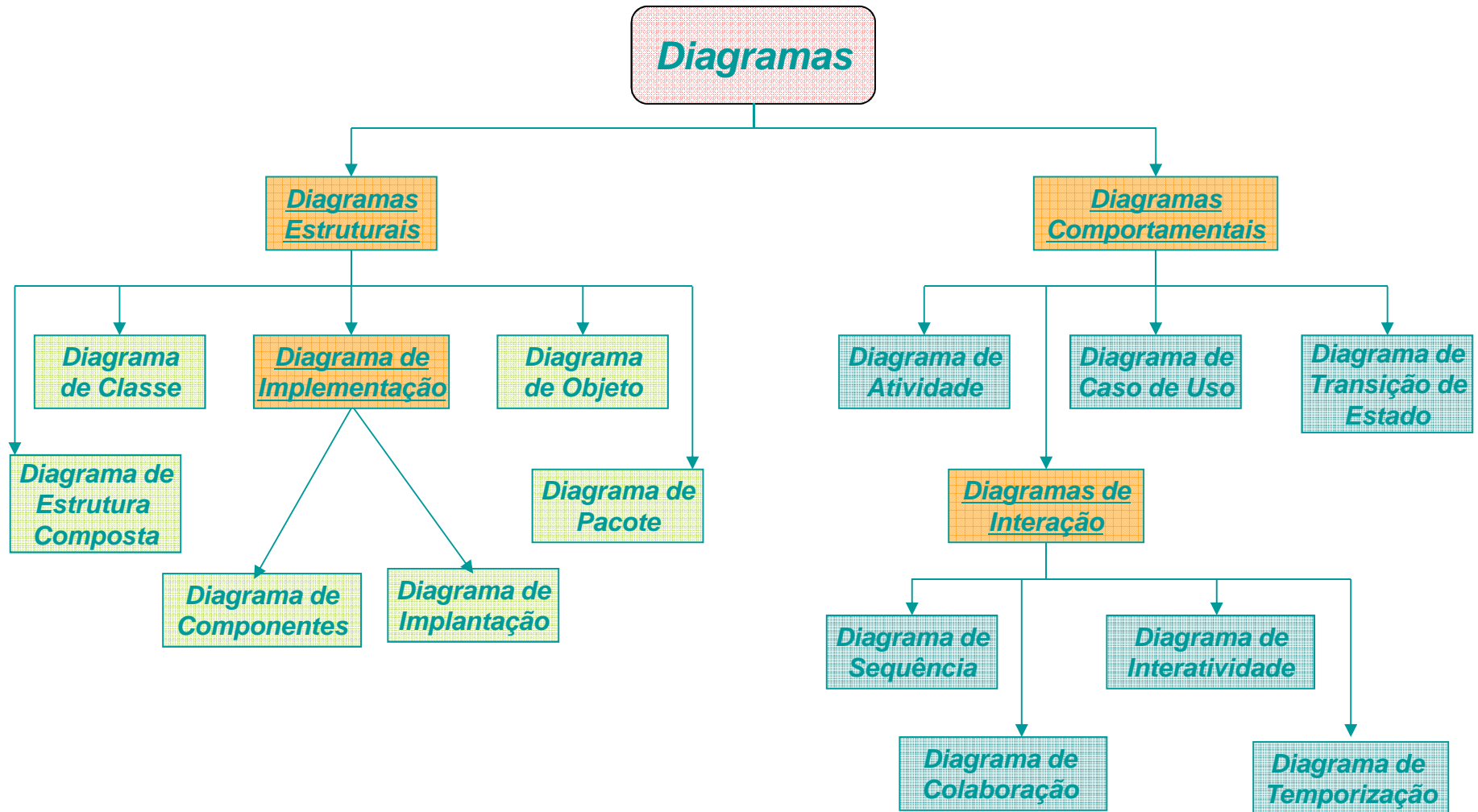


Diagrama *versus* tipo de sistema

Em sistemas

que utilizam muitos dados predominarão os modelos de visão estática.

GUI são importantes as visões estáticas e dinâmicas dos casos de uso.

de execução crítica em tempo real, as visões dinâmicas tendem a ser mais importantes.

distribuídos (web) os modelos de implementação e implantação é que tendem a ser mais importantes.

Por que construir modelos?

Os modelos ajudam a visualizar o sistema como ele é, ou como desejamos que seja.

Os modelos permitem especificar a estrutura ou comportamento de um sistema.

Os modelos proporcionam um guia para a construção do sistema.

Os modelos documentam as decisões tomadas.

Um modelo é uma simplificação da realidade.

Com a ajuda da modelagem, delimitamos o problema que estamos estudando, restringindo o nosso foco a um aspecto por vez. Booch, Rumbaugh, Jacobson, (2005, p.7).

Princípios básicos 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 / abstração.
- ✓ Os melhores modelos estão relacionados à realidade.
- ✓ Nenhum modelo único é suficiente. Qualquer sistema não trivial será melhor investigado por meio de um pequeno conjunto de modelos quase independentes com vários pontos de vista.

UML Linguagem ou Processo?

A UML é uma **linguagem** destinada a:

- Visualizar
- Especificar
- Construir
- Documentar

Artefatos de um sistema complexo de *software*.

Ela fornece vocabulário e regras para combinar as palavras desse vocabulário.

A UML indica como criar e ler modelos bem formulados, mas **não** explica **quando**, **onde** e **qual** diagrama usar. A definição da sequência das atividades faz parte do **processo** de desenvolvimento do software.

UML pode ser utilizada para modelar sistemas

- Informação corporativos
- Bancários e financeiros
- Telecomunicações
- De defesa
- Transporte
- Venda de varejo
- Científicos
- Distribuídos baseados na Web

Mas, não está restrita à modelagem de software. Ela é suficientemente expressiva para modelar:

- Fluxos de trabalho
- Regras de negócio
- Projetos de hardware.

Modelagem orientada a Objetos

O principal bloco de construção do software é o objeto ou a classe.

O **objeto** é alguma coisa geralmente estruturada a partir do vocabulário no contexto do problema ou da solução.

A **classe** é a descrição de um conjunto de objetos comuns.

Um objeto tem:

- **Identidade** – é o que diferencia um objeto de outro;
- **Estado** – associado ao valor de seus atributos num dado instante;
- **Comportamento** – associado ao que o objeto faz, ou o que ele pode fazer em outros objetos.

UML - Tendências

O diagrama UML, a partir da versão 2.0 e das novas ferramentas implementadas, não é mais que apenas uma figura para modelar e documentar sistemas.

Ferramentas no padrão MOF-aware (Meta-Object Framework) podem capturar significado nos elementos gráficos e de seus relacionamentos através das notações UML, de forma a serem entendidos pela máquina.

Isso permite executar validações de consistência nos modelos e possibilita gerar automaticamente partes do código fonte da aplicação.

Conclusão: A criação, armazenamentos e manutenção dos modelos lidos por máquina se transformem na base para produção das aplicações e a base para o MDA (Model Driven Architecture).

UML - Tendências

Uso de ferramentas de BPMN (Business Process Modelling Notation), para a criação de representações gráficas do modelo de negócio das empresas de forma integrada com os diagramas propostos pela UML.

BPMN são facilmente compreendidos pelos todos os envolvidos no negócio, incluindo os analistas de negócios que criam os rascunhos iniciais dos processos, os desenvolvedores técnicos responsáveis pela implementação da tecnologia que irá implementá-los.

BPMN aplica princípios de modelagem para representar os processos de negócios, mesmo aqueles que são completamente realizados pelos seres humanos e não envolvem nenhum software.



Leiam o paper: WATSON, Andrew. Visual Modelling: past, present and future.

Blocos de construção da UML

O vocabulário da UML abrange três tipos de blocos de construção.

1 – Item

- 1.1 – Itens estruturais
- 1.2 – Itens comportamentais
- 1.3 – Itens de agrupamento
- 1.4 – Itens anotacionais

2 – Relacionamentos

- 2.1 – Dependência
- 2.2 – Associação
- 2.3 – Generalização
- 2.4 - Realização

3 – Diagramas

- 3.1 – Classes
- 3.2 – Objetos
- 3.3 – Componentes
- 3.4 – estruturas compostas
- 3.5 – Casos de uso
- 3.6 – sequências
- 3.7 – Comunicações
- 3.8 – Estados
- 3.9 – Atividades
- 3.10 – Implementação
- 3.11 – Pacotes
- 3.12 – Temporização
- 3.13 - Interação

1.1 - Itens estruturais

- São os substantivos utilizados em modelos UML.
- São partes estáticas do modelo que representam elementos conceituais ou físicos.
- Coletivamente são chamados de classificadores.

Janela
Coordenada Tamanho
Abrir () Fechar() Exibir()

Classe – descrições de um conjunto de objetos que compartilham os mesmos atributos, operações, relacionamentos e semântica.

<<Interface>> IJanela
Abrir () Fechar() Exibir()

Interface – coleção de operações que especificam os serviços de uma classe ou componente. Ela descreve todo o comportamento externo visível, ou apenas uma parte desse comportamento. Na interface são apresentados apenas as operações, os atributos não são relevantes.

1.1 - Itens estruturais

Colaborações – definem interações que funcionam em conjunto, para proporcionar um comportamento cooperativo superior à soma de todos os elementos.

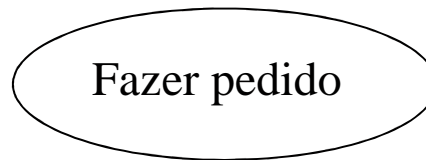
Uma classe pode participar em várias colaborações. Representada graficamente por:



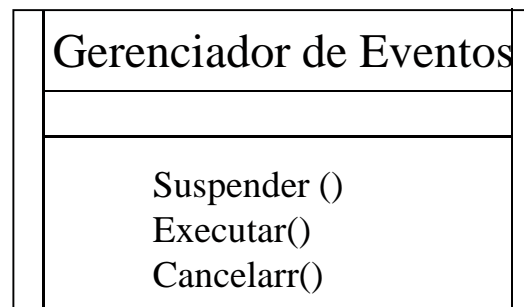
Podem ser usados em diagramas de Caso de Uso, de Classe, de Comunicação.

1.1 - Itens estruturais

Caso de Uso – descrição da sequência de ações realizadas pelo sistema que proporciona resultados observáveis para um determinado ator. Um Caso de Uso é realizado por uma colaboração. Representado graficamente por:

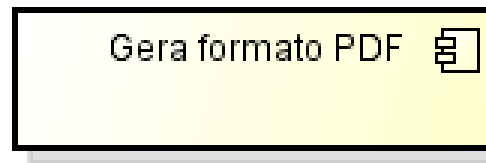


Classes ativas – São classes cujos objetos têm um ou mais processos ou threads, e, portanto, podem iniciar atividade de controle. Representam comportamento concorrente com outros elementos.



1.1 - Itens estruturais

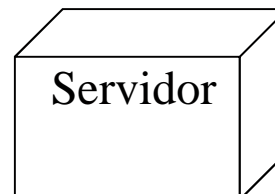
Componentes – são partes modulares de um sistema que ocultam sua representação através de um conjunto de interfaces externas.



Artefato – Representam itens físicos da implantação (programas executáveis, scripts, etc).



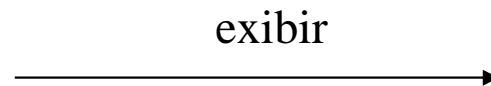
Nó – É um elemento físico existente em tempo de execução que representa um recurso computacional.



1.2 - Itens comportamentais

São as partes dinâmicas dos modelos UML. São os verbos de um modelo, representando comportamento no tempo e no espaço.

Interação – É um comportamento que abrange um conjunto de mensagens trocadas entre objetos em um determinado contexto para realização de propósitos específicos. Representado graficamente por:

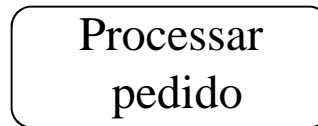


Estado – É um comportamento que especifica as sequências de estados que objetos ou interações passam durante sua existência em resposta a eventos.

Aguardando

1.2 - Itens comportamentais

Atividade – É um comportamento que especifica a sequência de etapas que um processo computacional realiza. Representa uma ação.



Na **interação** o foco está no **conjunto de objetos** que ele interage.

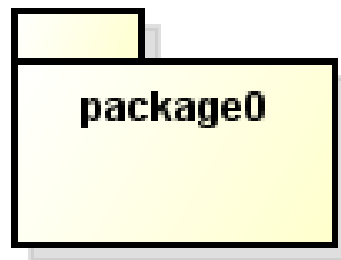
No **estado** o foco está no ciclo de vida de **um objeto por vez**.

Na **atividade** o foco está nos **fluxos entre as etapas**, independente de qual objeto realiza cada etapa.

1.3 - Itens de agrupamento

São as partes organizacionais dos modelos de UML. São blocos em que os modelos podem ser decompostos.

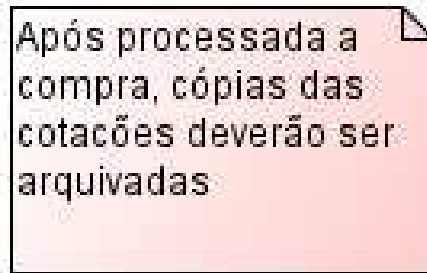
Pacotes é um mecanismo de propósito geral para organizar o projeto. É puramente conceitual e é utilizado para agrupar itens estruturais, comportamentais e até outros itens.



1.4 - Itens anotacionais

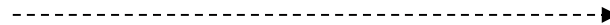
São as partes explicativas dos modelos UML. São comentários, incluídos para descrever, esclarecer e fazer alguma observação sobre qualquer elemento do modelo.

Nota é um símbolo para as restrições e comentários anexados a um elemento ou coleção de elementos.

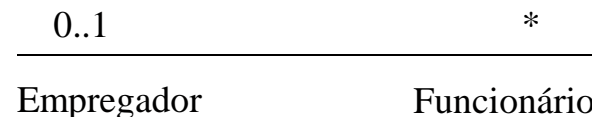


2 – Relacionamento

2.1 – Dependência - É um relacionamento semântico entre dois itens, nos quais a alteração de um (o item independente) pode afetar a semântica do outro (item dependente). Graficamente representado por:



2.2 – Associação - É um relacionamento estrutural entre classes que descreve um conjunto de ligações que representam conexões entre objetos que são instâncias das classes. Graficamente representado por:



A Agregação - É um tipo especial de associação, representando relacionamento estrutural entre o **todo** e as **suas partes**.

2 – Relacionamento

2.3 – Generalização - É um relacionamento de especialização / generalização, no qual os objetos dos elementos especializados (os filhos) são substituíveis por objetos do elemento generalizado (o pai classe base). Graficamente representado por:



2.4 – Realização - É um relacionamento semântico entre classificadores, em que um classificador especifica um contrato em que outro classificador garante executar. Esses relacionamentos são encontrados entre interfaces e classes ou componentes que as realizam.



Classificador – O nome geral para uma classe, interface, componentes, casos de uso e atores.

3 – Diagramas

Não percam nos próximos capítulos...

Algumas das ferramentas de modelagem UML



Astah Community (antigo Jude)

<http://astah.net/download#community>



Enterprise Architect (EA)

<http://www.sparxsystems.com/products/ea/index.html>



Rational Rose

<http://www-01.ibm.com/software/awdtools/developer/rose/>



Microsoft Visio

<http://office.microsoft.com/pt-br/visio/>

Referência

BOOCH, Grady, RUMBAUGH, James, JACOBSON, Ivar. UML Guia do Usuário 2ª ed. Campus, 2005.

BEZERRA, Eduardo. Princípios de análise e projeto de sistemas com UML. 3ª ed. Campus, 2015.

MEDEIROS, Ernani S. Desenvolvendo software com UML 2.0: definitivo. Pearson Makron Books 2004.