
Engenharia de Software

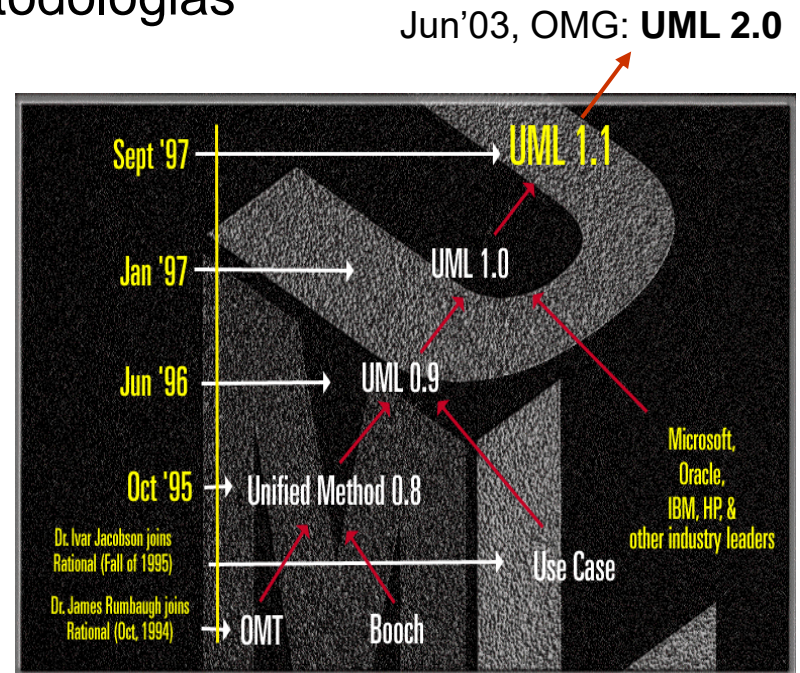
Linguagem UML

Luís Morgado

Instituto Superior de Engenharia de Lisboa
Departamento de Engenharia de Electrónica e Telecomunicações e de Computadores

Linguagem UML

- **UML** - *Unified Modeling Language*
- Resulta da combinação de várias metodologias de modelação orientadas a objectos
 - OMT (*Jim Rumbaugh*)
 - Booch Method (*Grady Booch*)
 - OOSE (*Ivar Jacobson*)
- Linguagem para descrever conhecimento
 - Acerca do domínio do problema
 - Acerca do domínio da solução
 - Linguagem gráfica
 - Elementos gráficos e textuais
- Normalizada no âmbito do OMG (*Object Management Group*)
 - Especificações formais
 - Adequada a diferentes âmbitos de modelação de sistemas



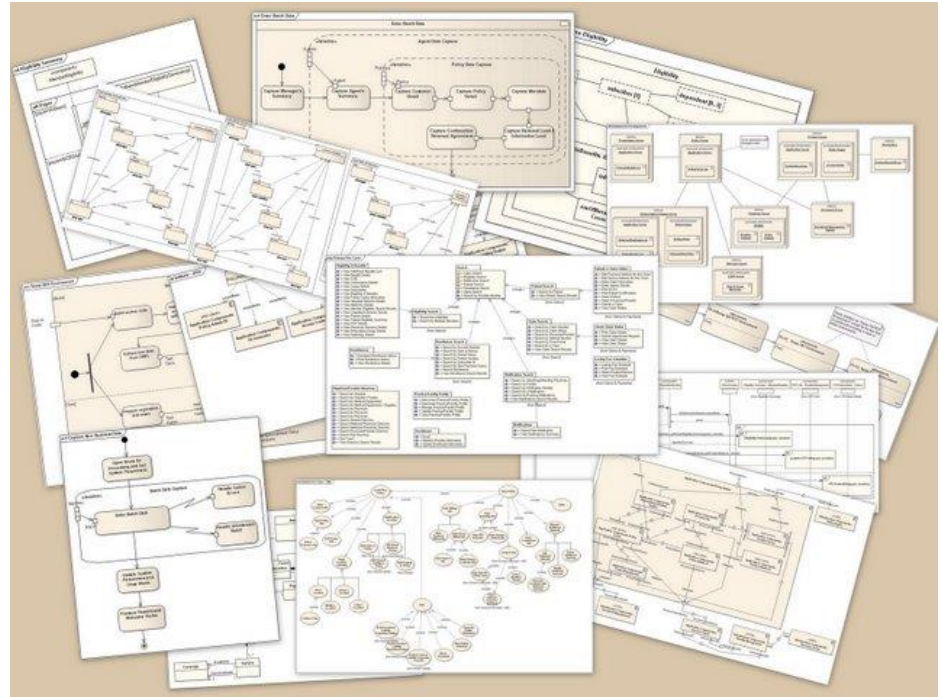
[Rational]

Linguagem UML

- Áreas de aplicação
 - Modelação de arquitectura lógica e física
 - Estrutura
 - Dinâmica
 - Funcionalidade
 - Modelação de processos de negócio
 - Modelação de dados
 - Modelação geral de sistemas
- Modos de utilização
 - Esboços
 - Especificações a diferentes nível de detalhe
 - Geração de código (*forward / reverse engineering*)

Linguagem UML

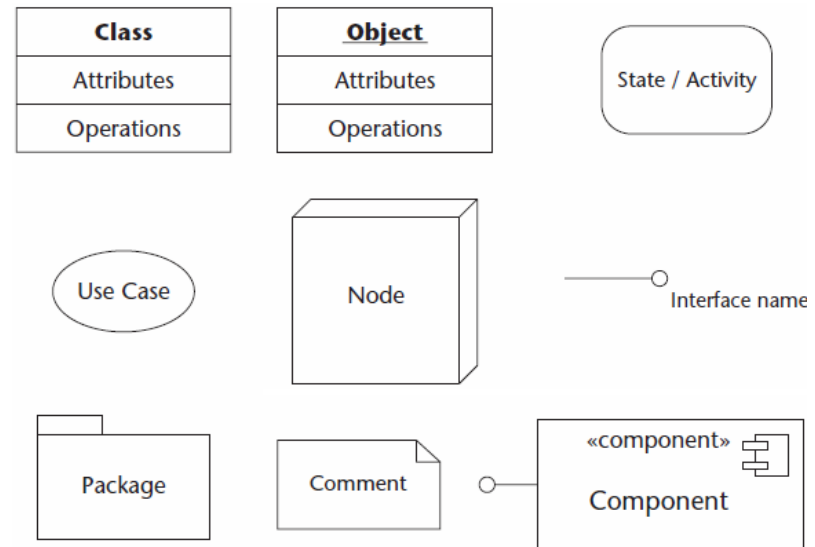
- Linguagem gráfica (“visual”)
 - Conceitos
 - Representados por símbolos gráficos
 - Organizados em diagramas
 - Eficácia
 - Descrição
 - Comunicação
 - Compreensão
- Elementos base
 - Classificadores
 - Mecanismos de extensão
 - Diagramas



Linguagem UML

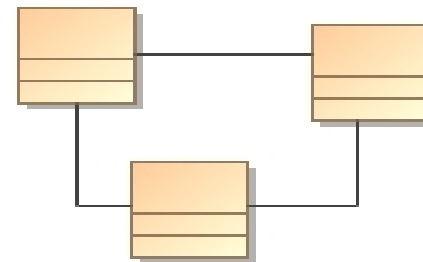
- Classificadores (tipos de elementos)

- Actor
- Association
- Class
- Component
- Datatype
- Interface
- Node
- Signal
- Subsystem
- Use Case



- Diagramas

- Representação 2D facilita a percepção das relações entre elementos



Linguagem UML

- Mecanismos de Extensão
(*Extensibility Mechanisms*)

- Estereótipos (*Stereotypes*)

- Permitem estender o vocabulário UML de modo a criar novos elementos de modelação, com características específicas, a partir dos já existentes
 - São utilizados para criar novos elementos de modelação vocacionados para um domínio de representação específico

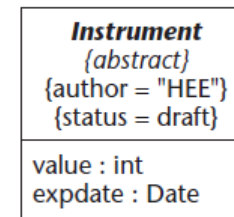
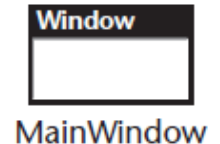
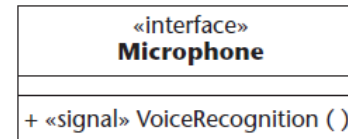
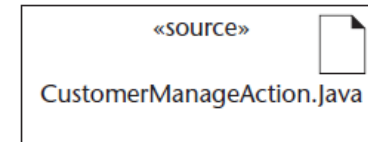
- Anotações (*Tagged values*)

- Propriedades associadas a elementos representadas por pares *palavra_chave* – *valor*
 - São utilizadas para estender as propriedades dos elementos da linguagem de modo a ser possível a associação de informação específica

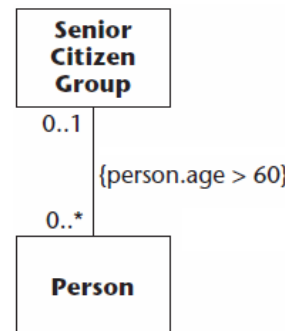
- Restrições (*Constraints*)

- Propriedades que especificam a semântica e/ou condições que se devem verificar em relação a elementos de um modelo
 - São utilizadas para estender a semântica dos elementos da linguagem no sentido da sua modificação ou extensão

Notação: «estereótipo»



Notação:
{propriedade}



Notação:
{restrição}

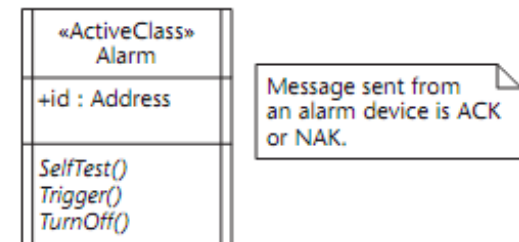
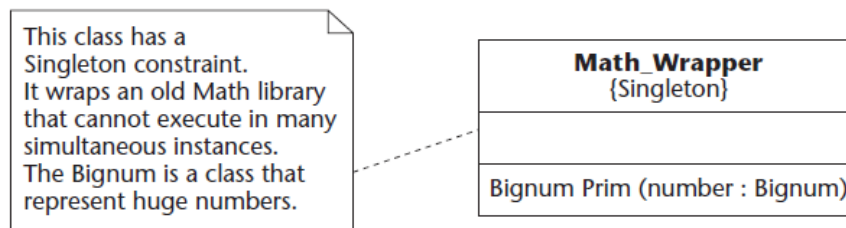
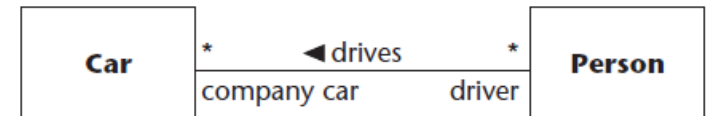
[Eriksson et al., 2004]

Linguagem UML

- Complementos de Informação
 - Adornos
 - Informação textual ou gráfica indicativa de aspectos específicos de um elemento
 - Comentários
 - Elemento gráfico contendo informação textual



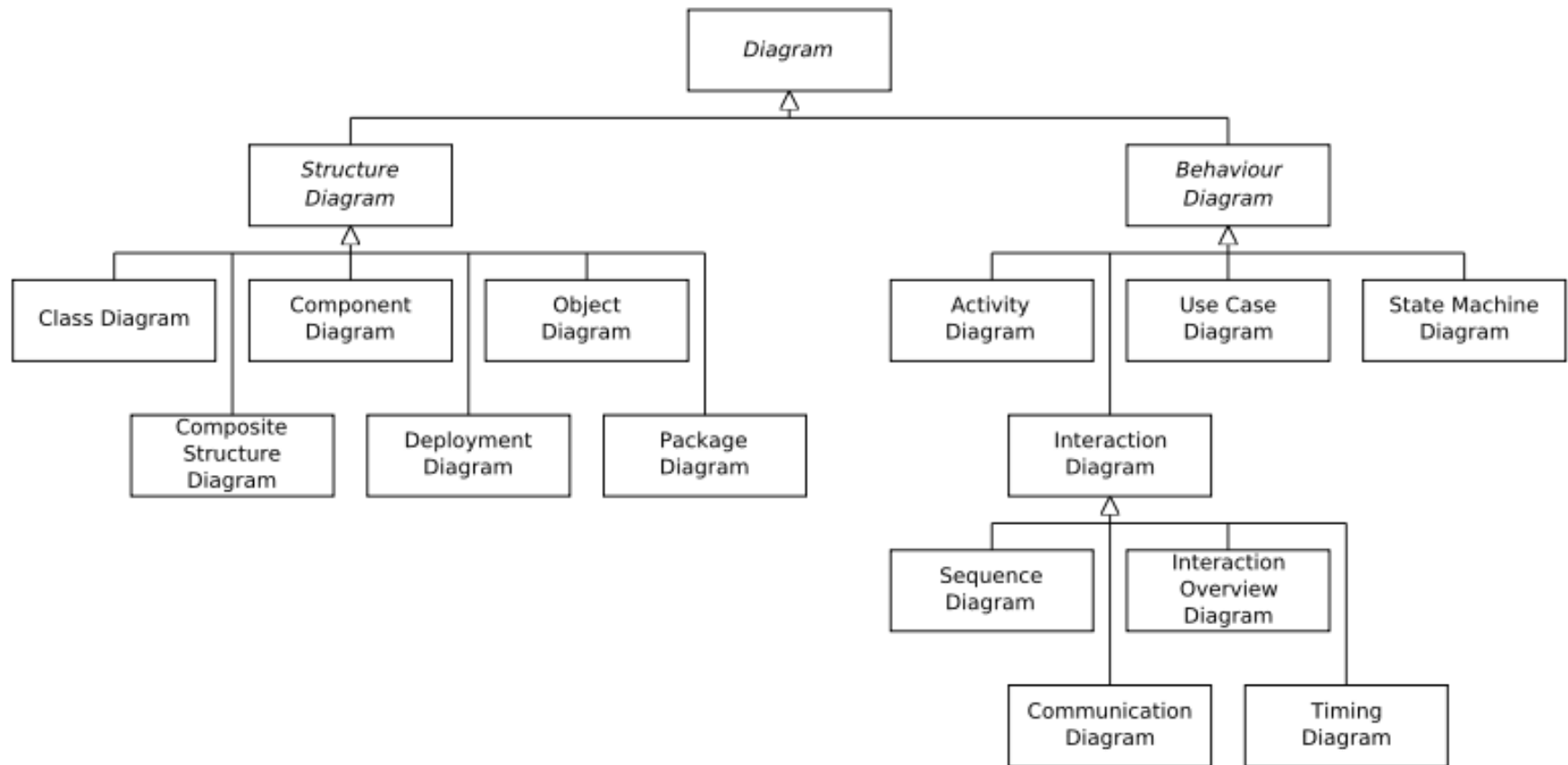
Classe: **Negrito**
Objecto: Sublinhado



[Eriksson et al., 2004]

Linguagem UML

PERSPECTIVAS DE MODELAÇÃO



Tipos de Diagramas UML

- **Funcionalidade**

- Diagramas de Casos de Utilização

- **Interacção**

- Diagramas de Sequência
- Diagramas de Comunicação
- Diagramas Temporais (*Timing Diagrams*)
- Diagramas de Enquadramento de Interacção (*Interaction Overview Diagrams*)

- **Estrutura**

- Diagramas de Classes
- Diagramas de Objectos
- Diagramas de Estrutura Composta

- **Dinâmica**

- Diagramas de Transição de Estado
- Diagramas de Actividade

- **Implementação**

- Diagramas de Componentes

- **Ambiente**

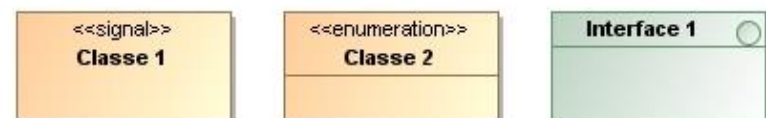
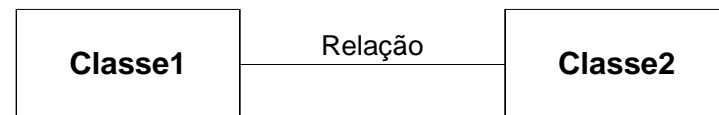
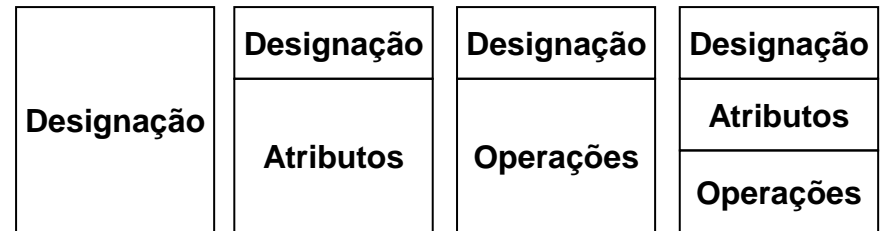
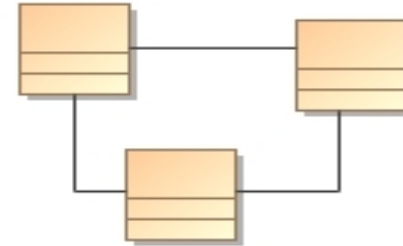
- Diagramas de Implantação

- **Organização**

- Diagramas de Pacotes (*Package Diagrams*)

Diagramas de Classes

- Descrevem uma abstracção das partes e das relações entre partes de um sistema
 - Organização estática do sistema
 - Foco na estrutura
- Classes
 - Atributos
 - Definição de estrutura
 - Operações
 - Encapsulamento de comportamento
- Relações
 - Representação de interdependências
- Estereótipos de classes



Diagramas de Classes

- Atributos

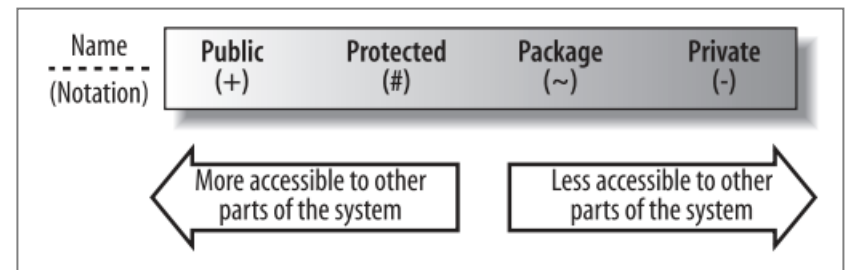
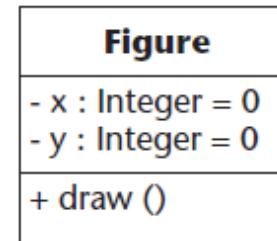
- Representação de estrutura
- Caracterizados por:
 - Designação
 - Tipo
 - Visibilidade
 - Público (+)
 - Privado (-)
 - Protegido (#)
 - Pacote (~)

- Sintaxe:

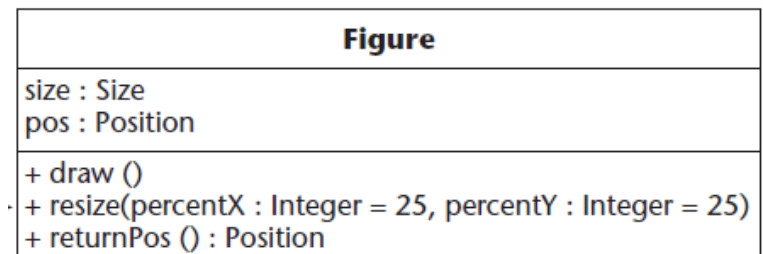
- visibility name:type = init_value
{property_string}

- Operações

- Representação de comportamento

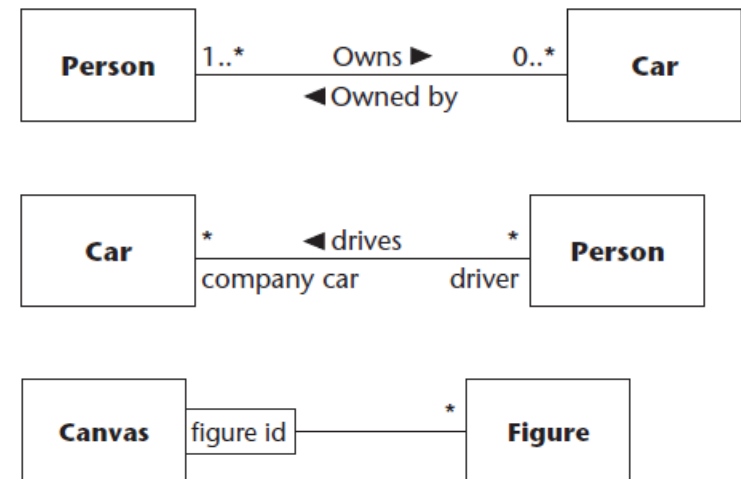
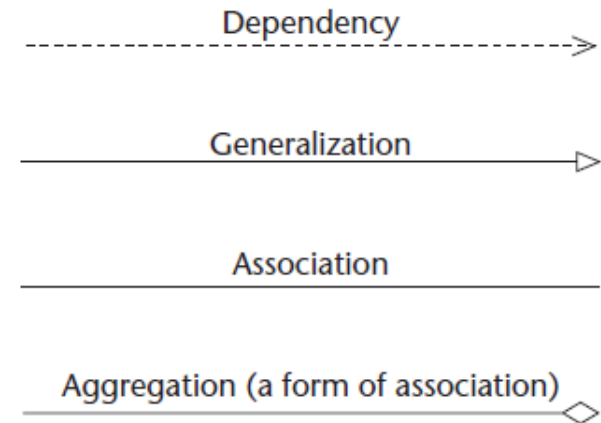


[Miles & Hamilton, 2006]



Diagramas de Classes

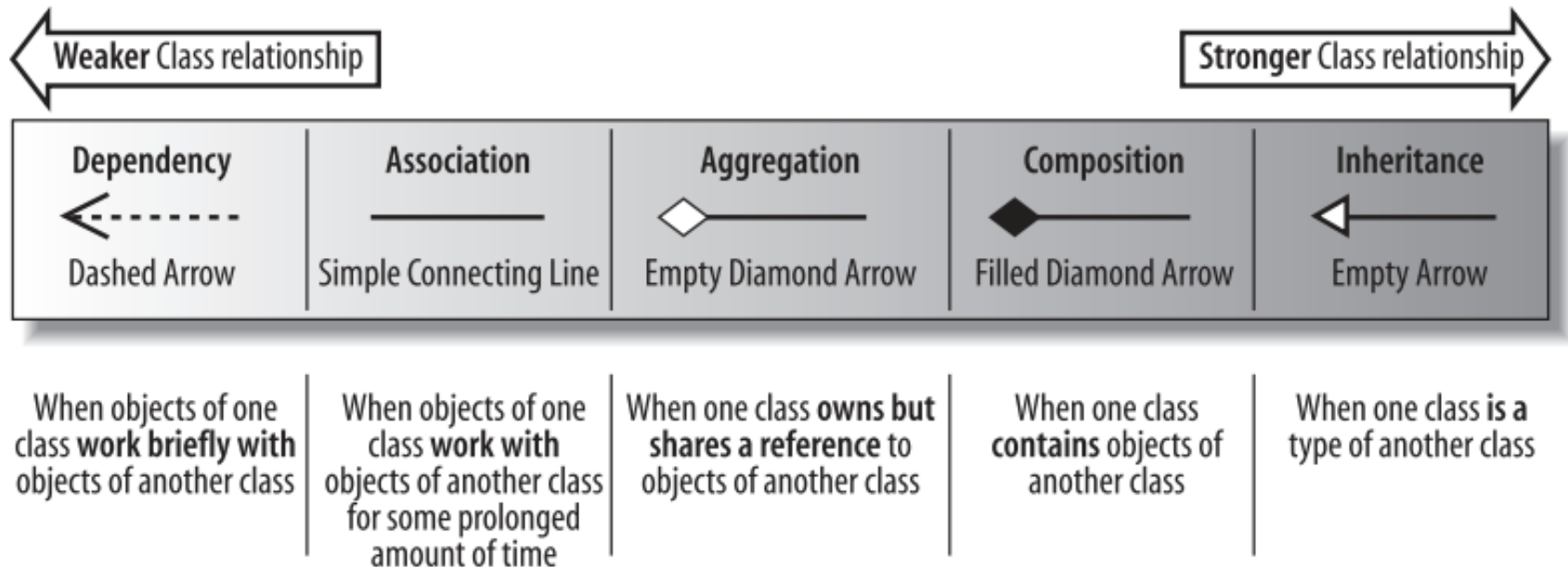
- Relações entre classes
 - Dependência
 - Associação
 - Agregação
 - Composição
 - Generalização
- Propriedades das relações
 - Direcção
 - Multiplicidade
 - Papéis
 - Qualificação
 - O qualificador de uma associação designa uma chave utilizada para obter um item da colecção respectiva



[Eriksson et al., 2004]

Diagramas de Classes

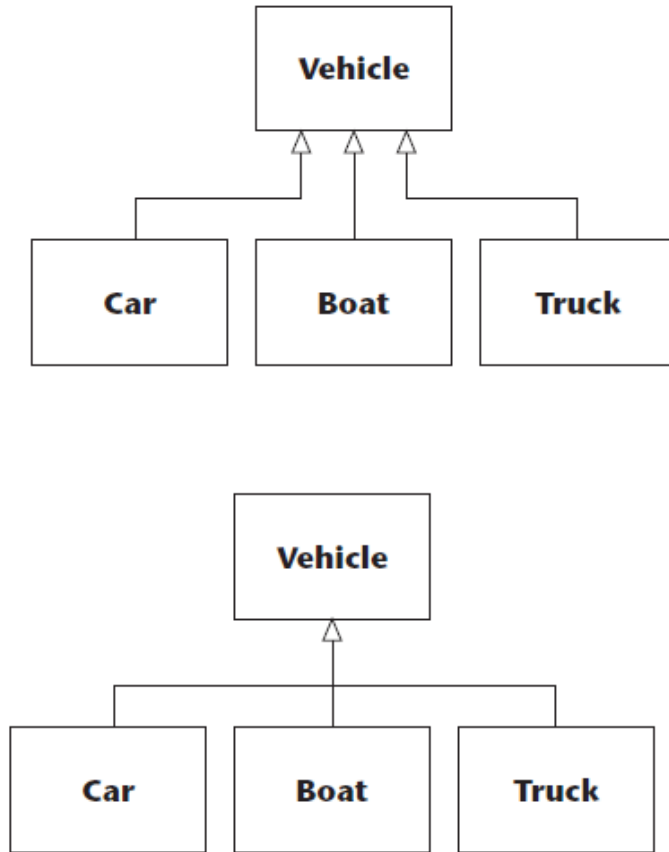
Relações entre classes e nível de acoplamento



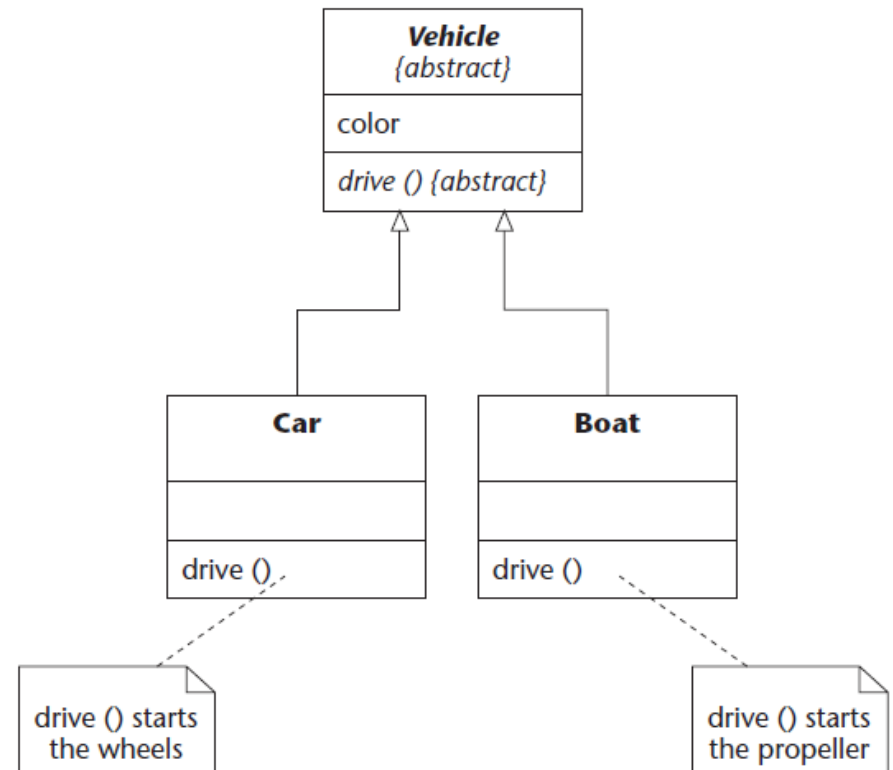
[Miles & Hamilton, 2006]

Diagramas de Classes

- Herança



- Polimorfismo

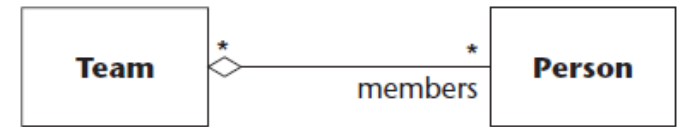


[Eriksson et al., 2004]

Diagramas de Classes

- **Agregação**

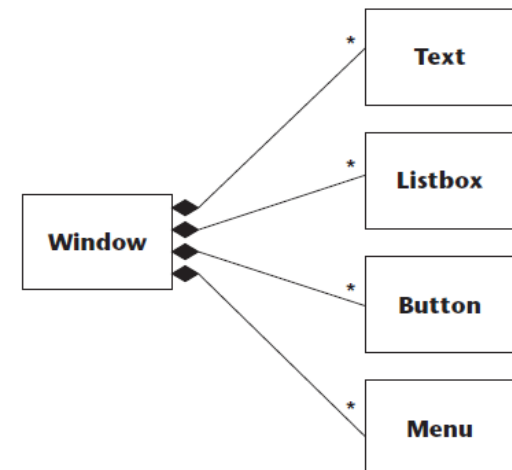
- Relação Parte - Todo
- Denota que uma parte contém outra em termos lógicos ou físicos
- Forma fraca de composição de partes
 - Não define restrições semânticas acerca de:
 - Pertença das partes
 - Criação e destruição das partes
 - Períodos de existência das partes



Relação *parte-todo*

- **Composição**

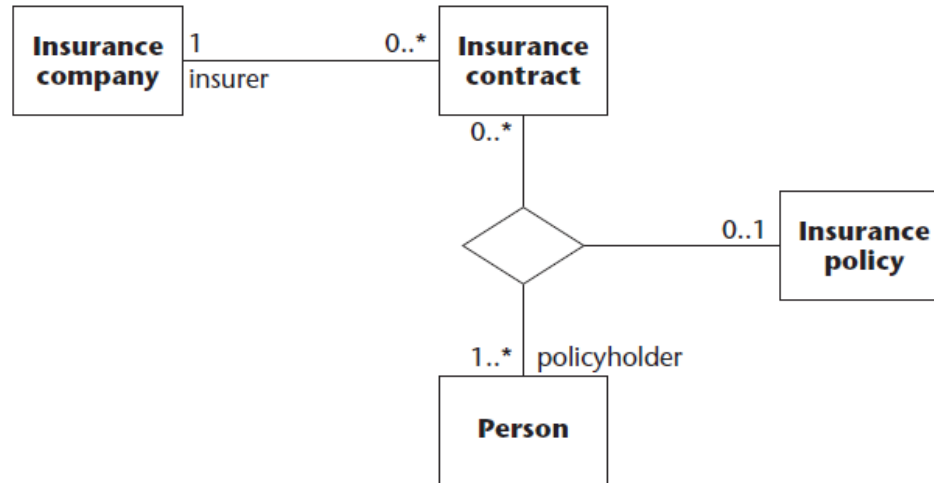
- Forma forte de composição de partes
 - Define restrições semânticas específicas
 - O todo cria e destrói as partes
 - Sobreposição de períodos de existência das partes
 - Implica exclusividade na composição das partes
 - Organização hierárquica em árvore
 - Níveis de abstracção



Regra da *não-partilha*

Diagramas de Classes

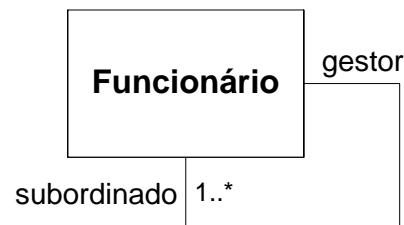
- Associações ternárias (n -árias)



[Eriksson et al., 2004]

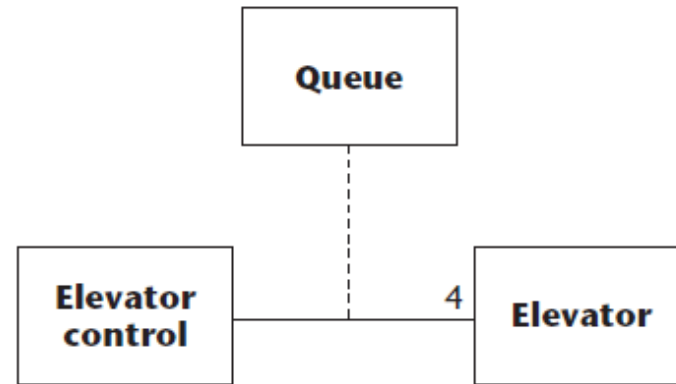
- Associações reflexivas

- Ocorre quando os objectos de uma classe podem desempenhar mais de um papel

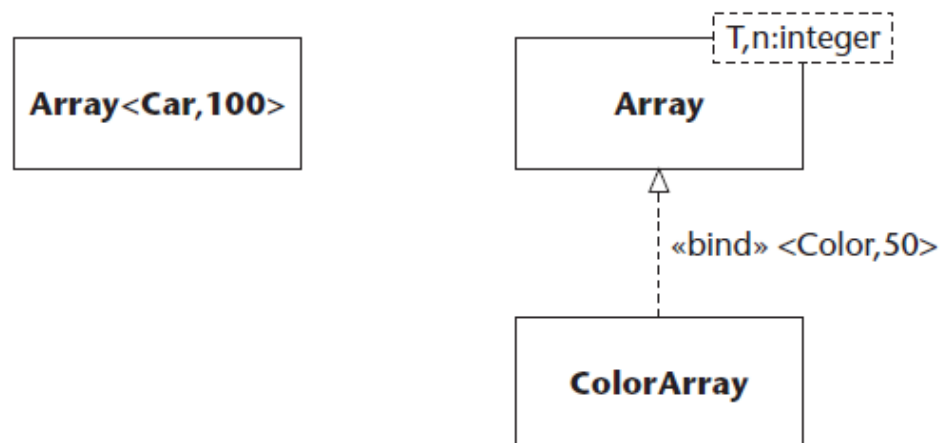


Diagramas de Classes

- Classes associativas
 - Atributos de associações
 - Podem estar associadas a outras classes



- Classes paramétricas

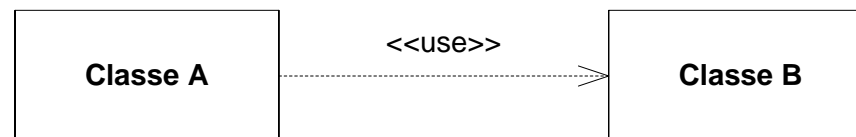


[Eriksson et al., 2004]

Diagramas de Classes

- Dependências

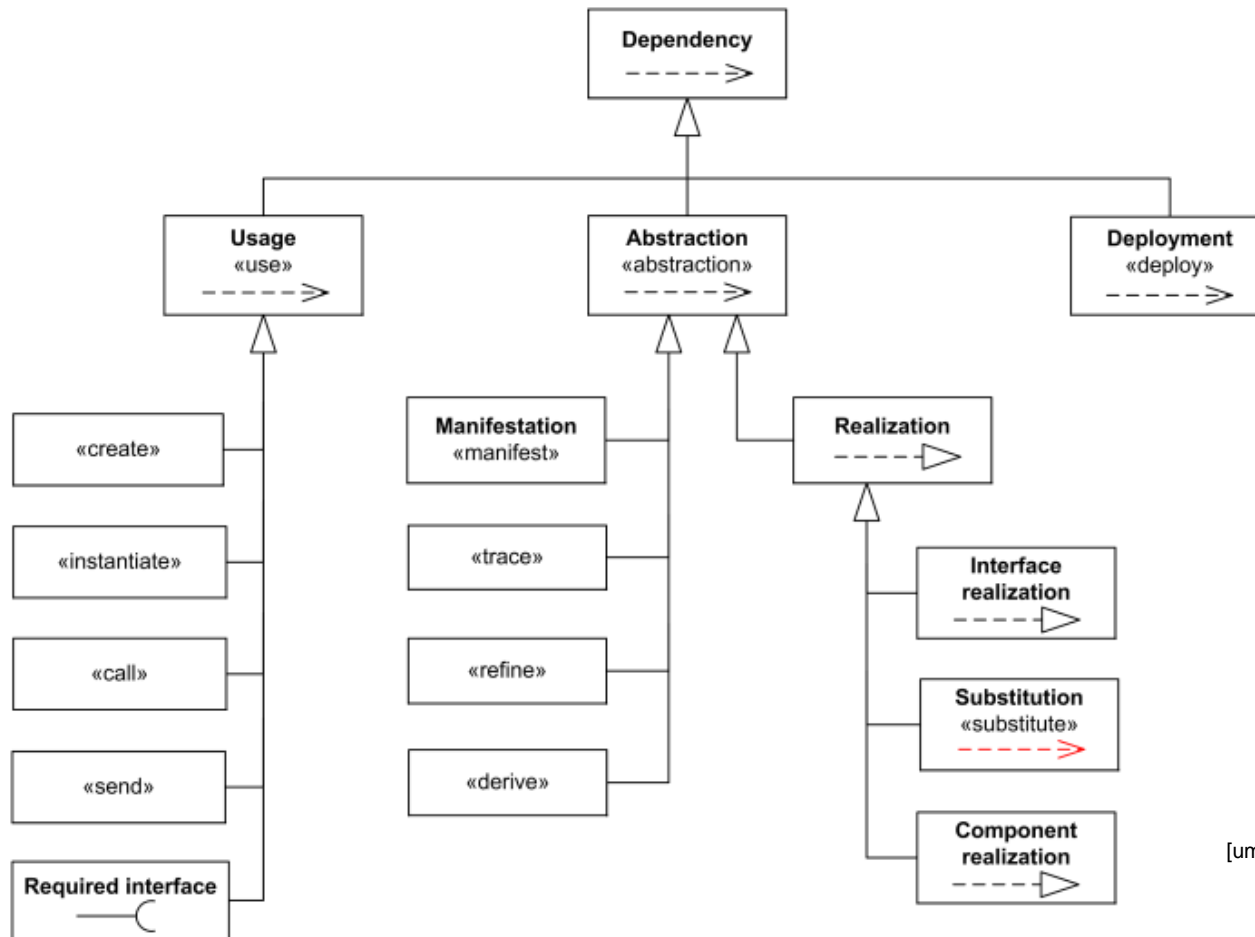
- Utilização quando uma classe não tem uma relação directa com outra classe mas depende dela de alguma forma
- Alterações numa classe podem implicar alterações noutra classe
- Definidas como estereótipos de associações
- Rastreabilidade
 - Alterações numa classe podem afectar outras classes
- Utilização de estereótipos



**Em sistemas complexos é importante
manter registo de dependências**

Diagramas de Classes

- Estereótipos de dependências
 - Especializam a relação geral de dependência



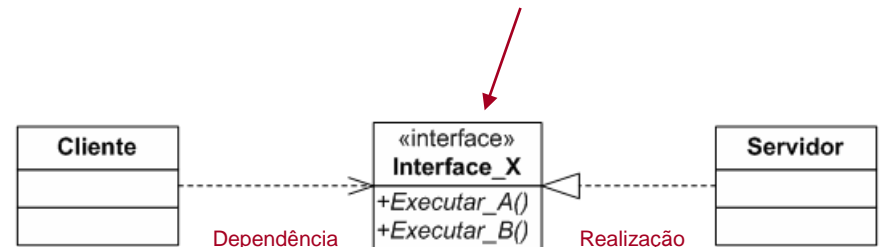
[uml-diagrams.org]

Diagramas de Classes

• Interface

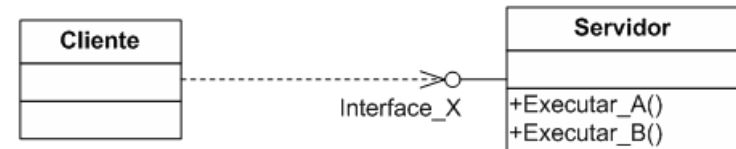
- Classificador que define as características visíveis de uma classe
- Conjunto **coeso** de características
- Define um contrato
 - Não implementa essas características
- **Encapsulamento**

A interface define um contrato de prestação de serviços independente da implementação



• Realização

- Representa implementação
- Relação entre uma interface e uma classe ou um componente
- Uma classe **realiza** as características de uma interface implementando-as
- Uma classe pode implementar mais que uma interface

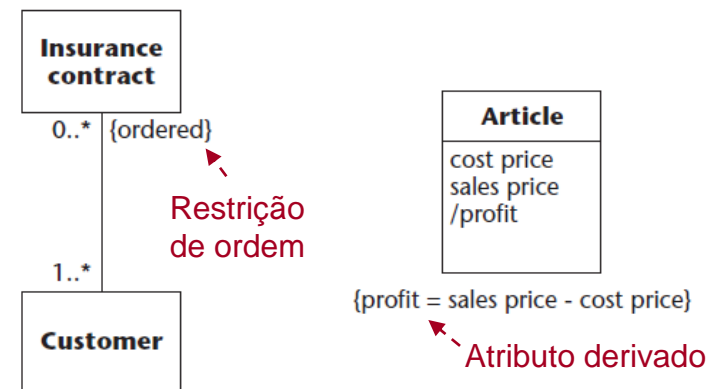
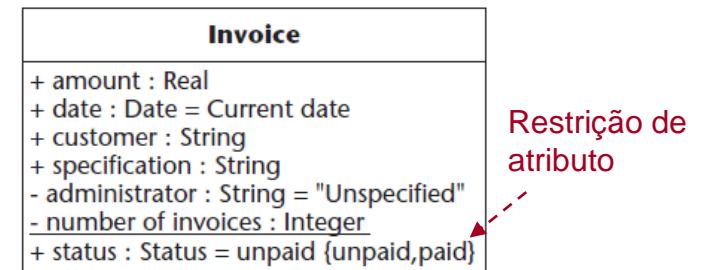
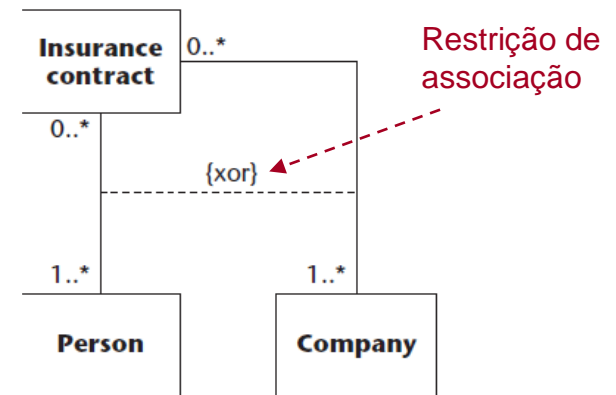


Diagramas de Classes

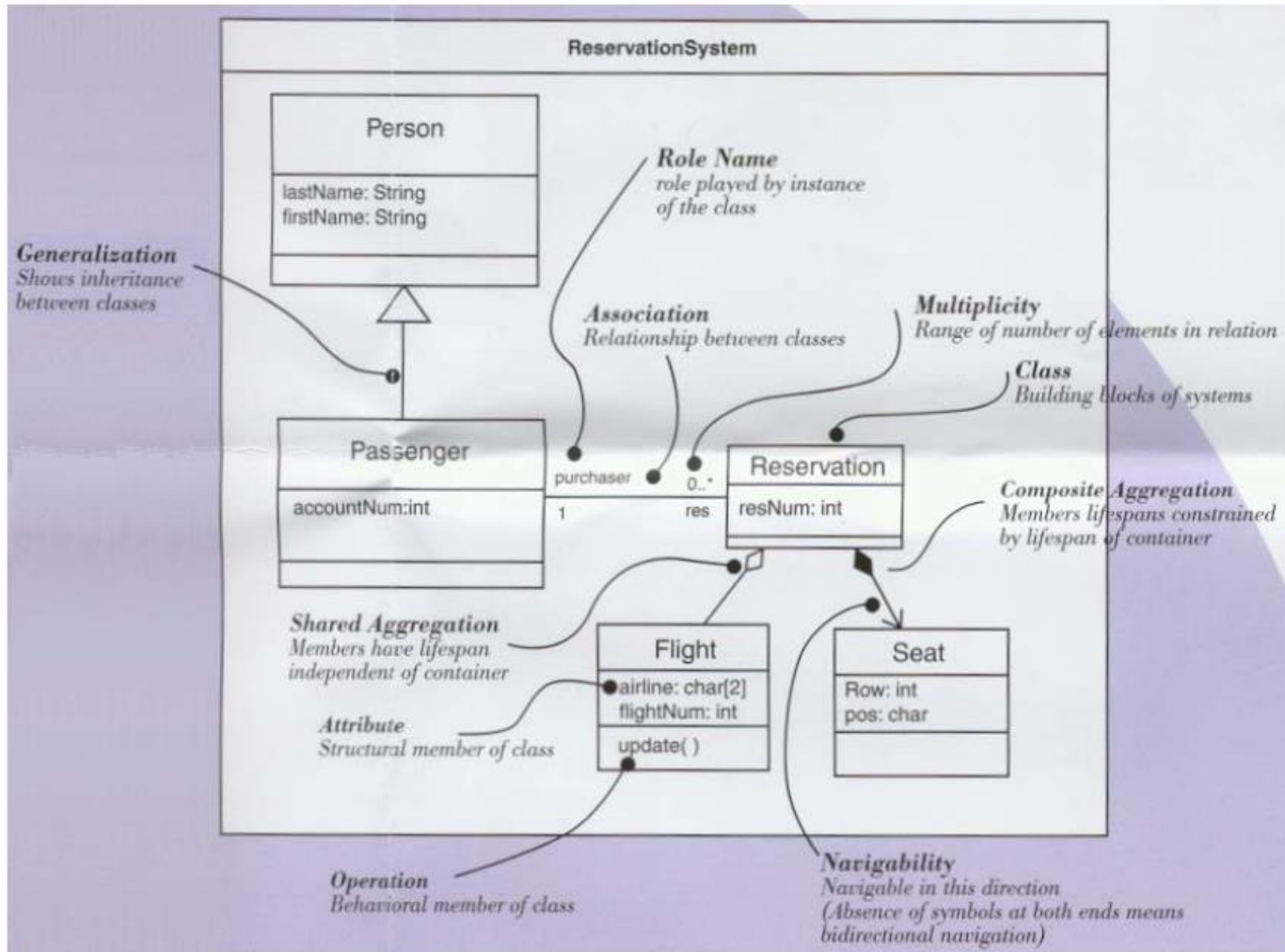
- Restrições

- Descrevem constrições aos elementos de um modelo
- Múltiplas formas
 - Por exemplo, multiplicidade
 - Pré-condições / Pós-condições
- Anexação de notas aos elementos do modelo
- Notação base UML
 - { Especificação da restrição }
 - Notas associadas aos elementos
- Especificação da restrição
 - Linguagem natural
 - Problema: **Ambiguidade**
 - OCL (*Object Constraint Language*)

Redução de multiplicidade



Diagramas de Classes



Class Models describe the basic building blocks of object-oriented systems and the relationship between them. [\[UML 2.0\]](#)

Diagramas de Objectos

- Descrevem a forma como instâncias de classes se relacionam ou operam em conjunto para realizar a funcionalidade do sistema
 - Elementos são instâncias de classes
 - Também designados diagramas de instâncias
 - Descrevem a concretização da estrutura num **instante específico de tempo**
- Utilização
 - Identificação de objectos e de relações entre objectos
 - Compreensão das formas de colaboração entre objectos para produzir a função do sistema

Diagrama de classes

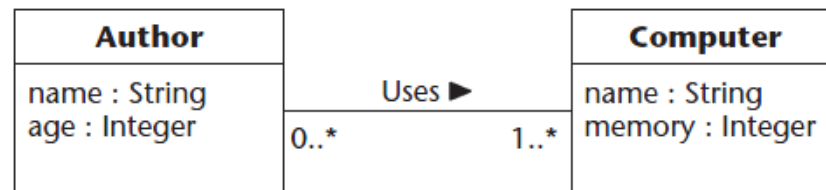
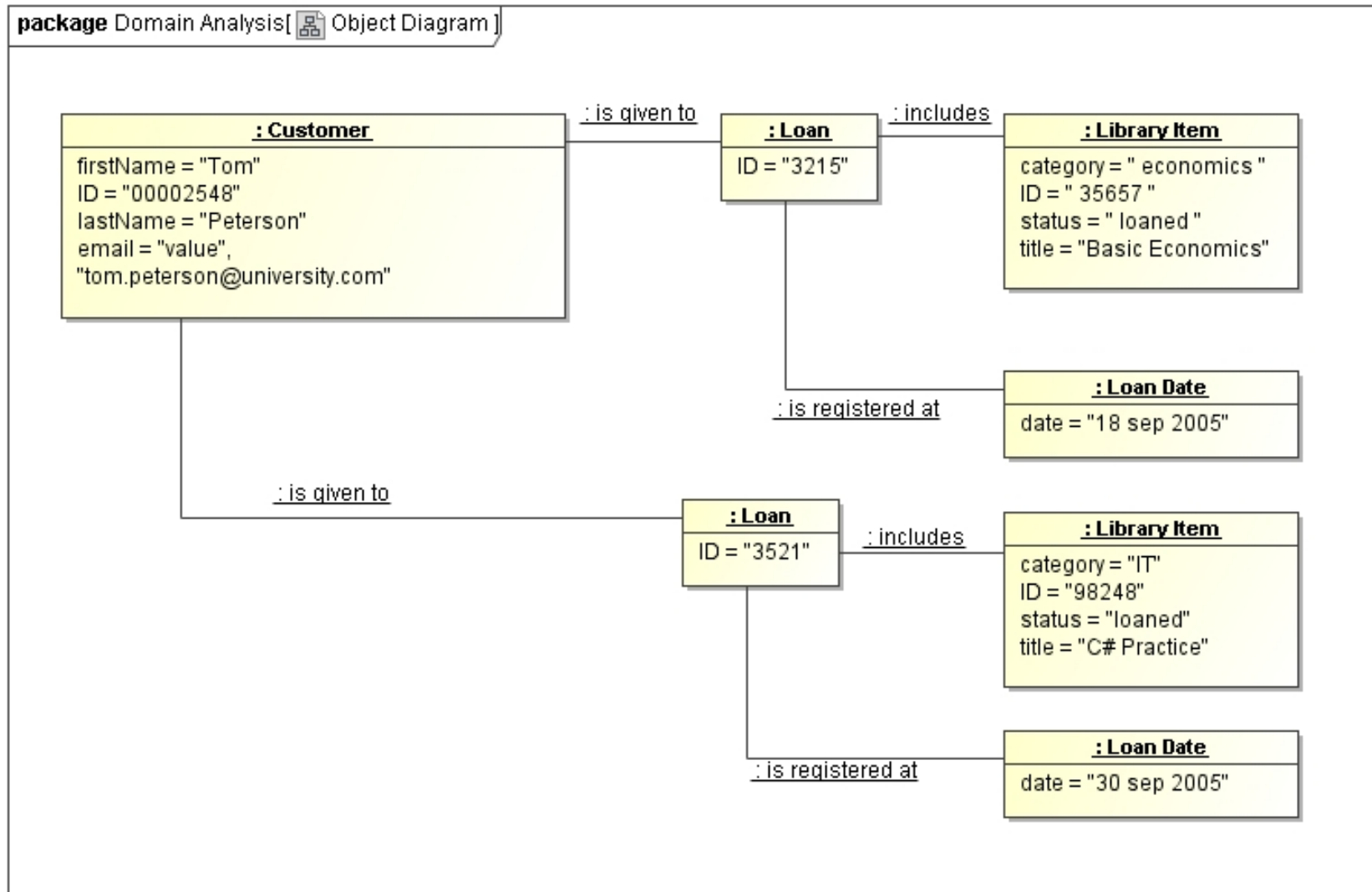


Diagrama de objectos



[Eriksson et al., 2004]

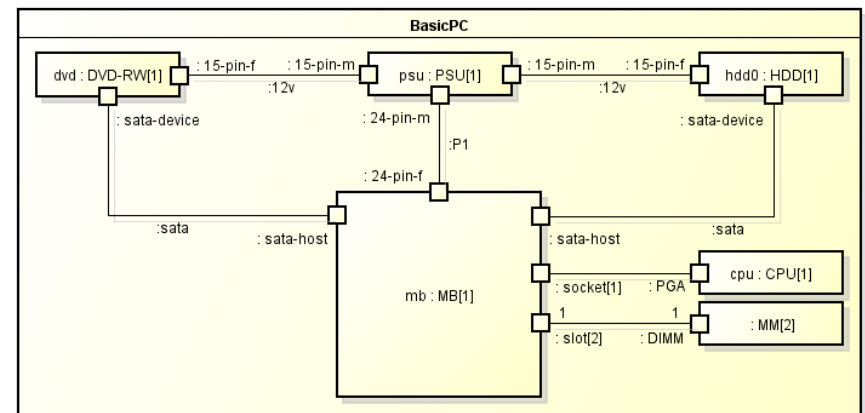
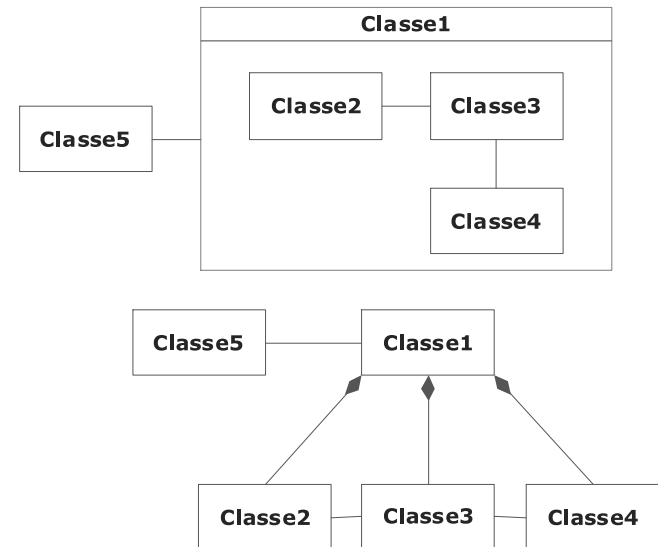
Diagramas de Objetos



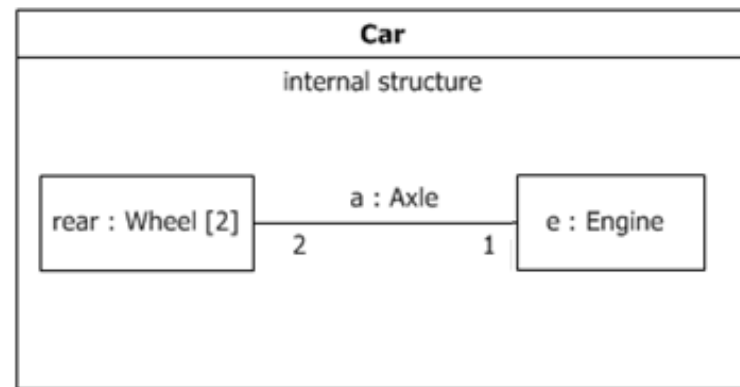
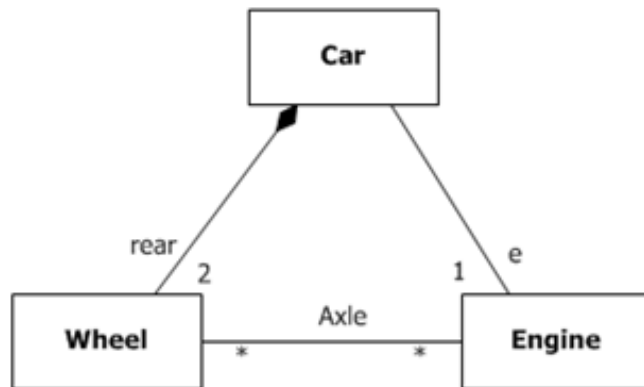
Diagramas de Estrutura Composta

- Focagem a diferentes níveis de detalhe
 - Essencial para lidar com a complexidade
 - Abstracção
 - Encapsulamento
- Subsistemas
- Partes
- Pontos de interacção (*port*)
 - Permite representar a ligação com a parte do classificador (classe, componente) que realiza determinada funcionalidade
 - Agrupam interfaces (uma determinada parte do classificador implementa ou utiliza uma interface)
- Interfaces
 - Disponibilizadas
 - Requeridas

Forma alternativa de representar composição

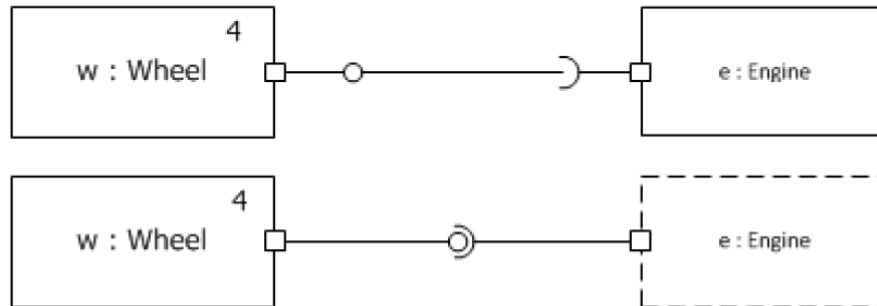


Diagramas de Estrutura Composta



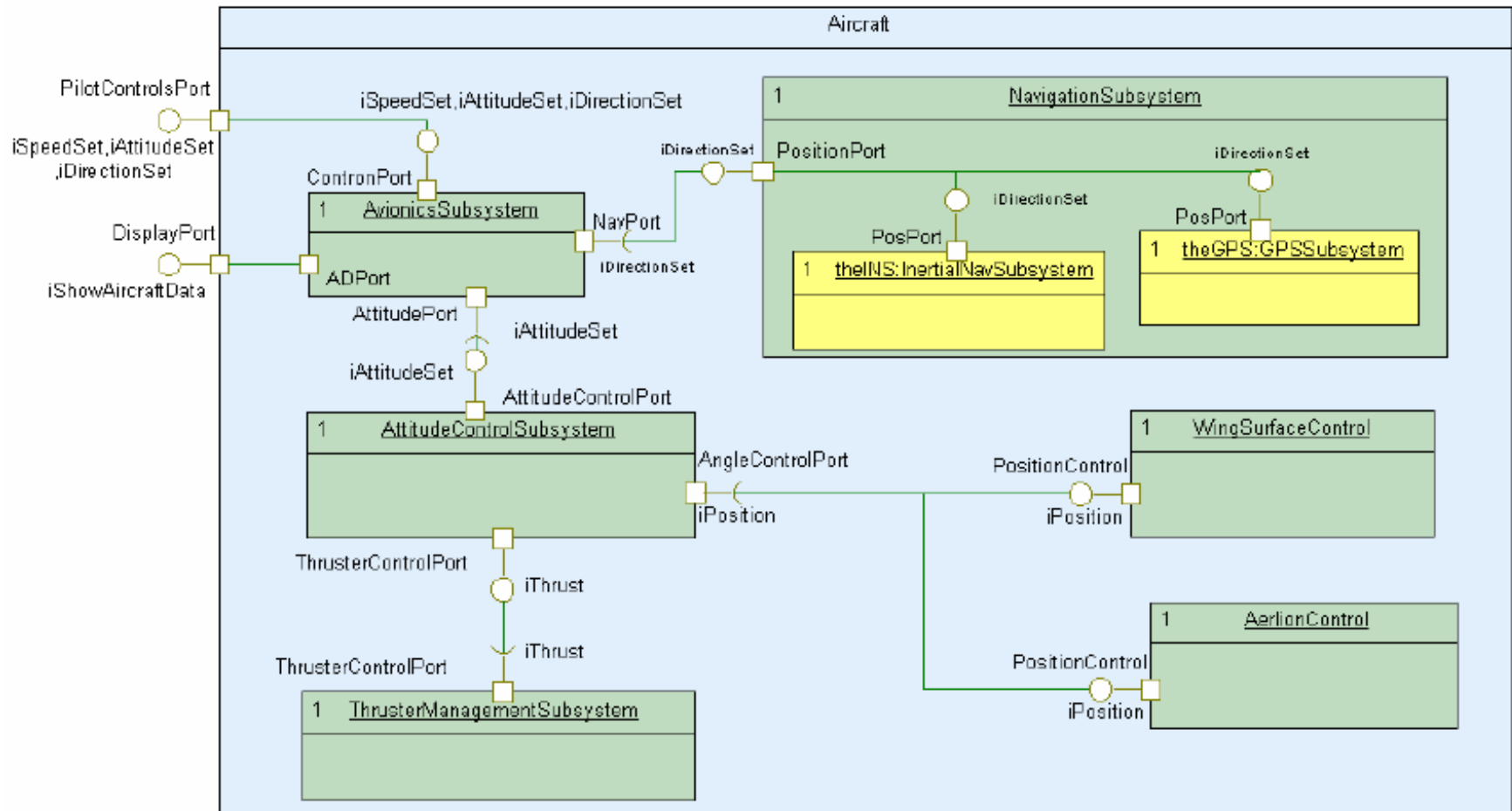
[OMG, 2020]

Diagramas de Estrutura Composta



[OMG, 2020]

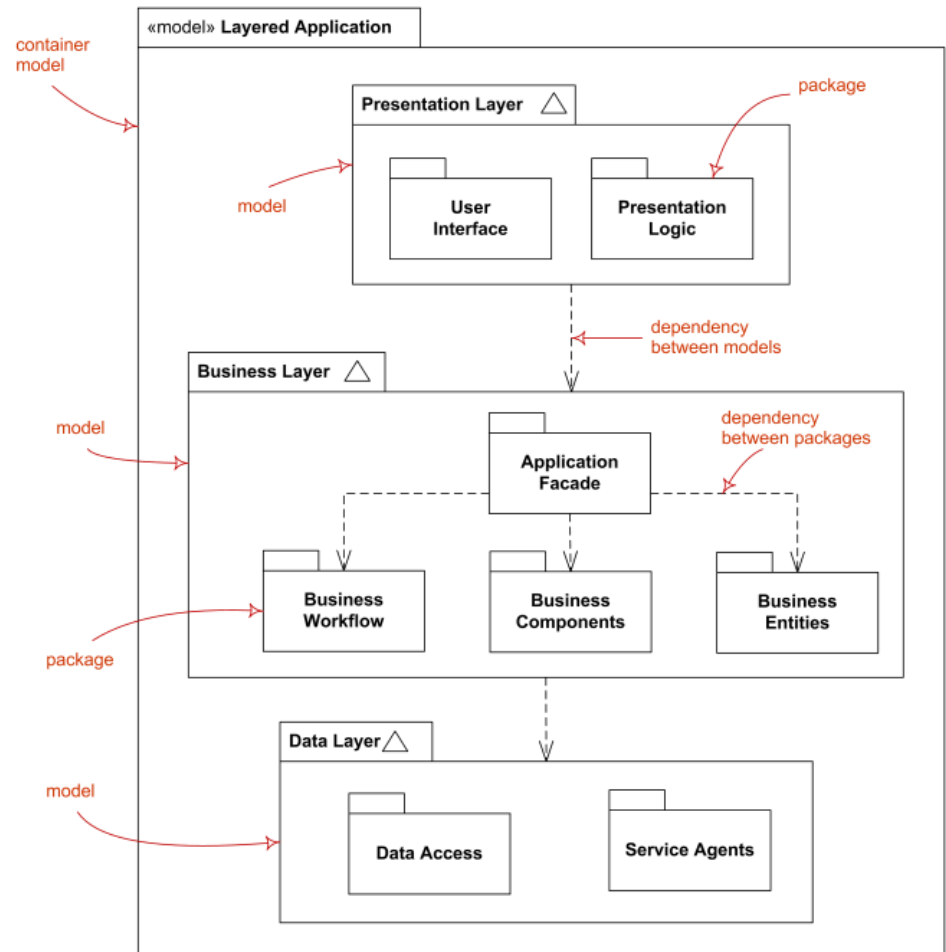
Diagramas de Estrutura Composta



[Douglass, 2006]

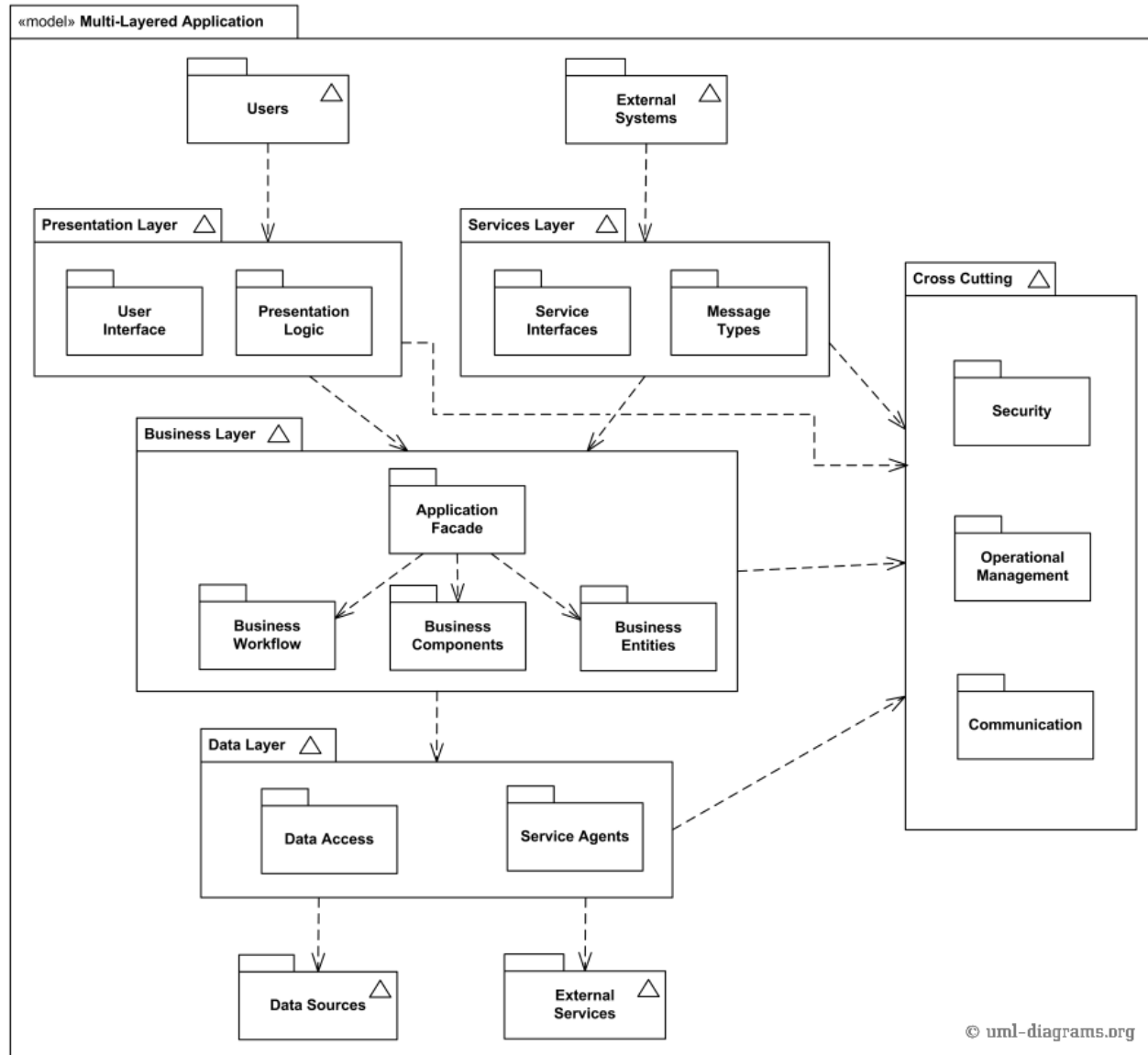
Diagramas de Pacotes (*Packages*)

- Organização hierárquica dos elementos do modelo
 - Gestão e organização de sistemas complexos
 - Foco nas dependências
 - Divisão de um sistema em subsistemas



[uml-diagrams.org]

Exemplo: Arquitetura Multi-Camada



Bibliografia

[Watson, 2008]

Andrew Watson, *Visual Modeling: past, present and future*, OMG, 2008.

[Meyer, 1997]

B. Meyer, *UML: The Positive Spin*, American Programmer - Special UML issue, 1997.

[Yelland et al., 2002]

Yelland, M. J., B. I. Moat, R. W. Pascal and D. I. Berry, *CFD model estimates of the airflow over research ships and the impact on momentum flux measurements*, Journal of Atmospheric and Oceanic Technology, 19(10), 2002.

[Selic, 2003]

B. Selic, *Brass bubbles: An overview of UML 2.0*, Object Technology Slovakia, 2003.

[Graessle, 2005]

P. Graessle, H. Baumann, P. Baumann, *UML 2.0 in Action*, Packt Publishing, 2005.

[Eriksson et al., 2004]

H. Eriksson, M. Penker, B. Lyons, D. Fado, *UML 2 Toolkit*, Wiley, 2004.

[USDT, 2005]

U.S. Department of Transportation, *Clarus: Concept of Operations*, Publication No. FHWA-JPO-05-072, 2005.

[Douglass, 2006]

B. Douglass, *Real-Time UML*, Telelogic, 2006.

[OMG, 2020]

Unified Modeling Language (Specification), OMG, 2020.