
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*

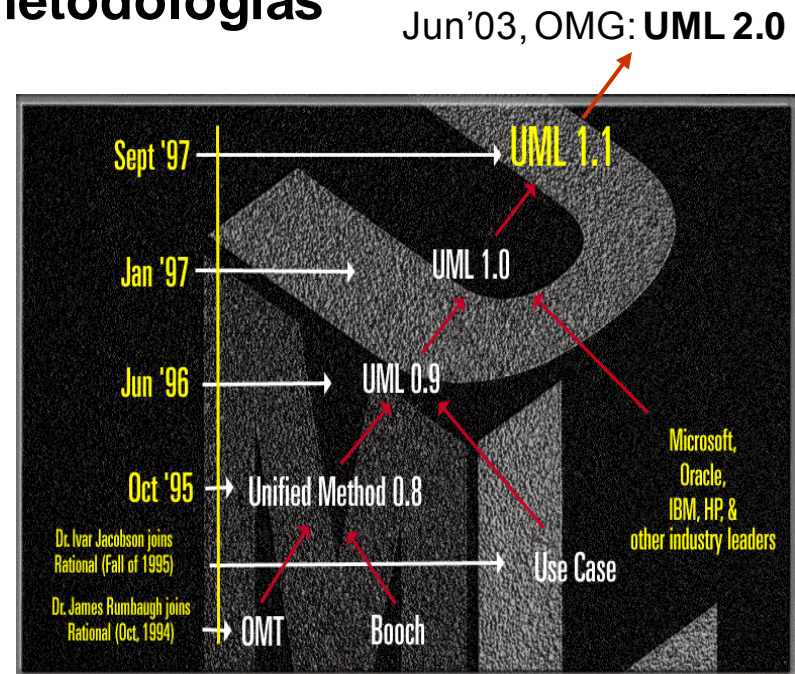
- **Resulta da combinação de várias metodologias** de modelação orientadas a objectos

- 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

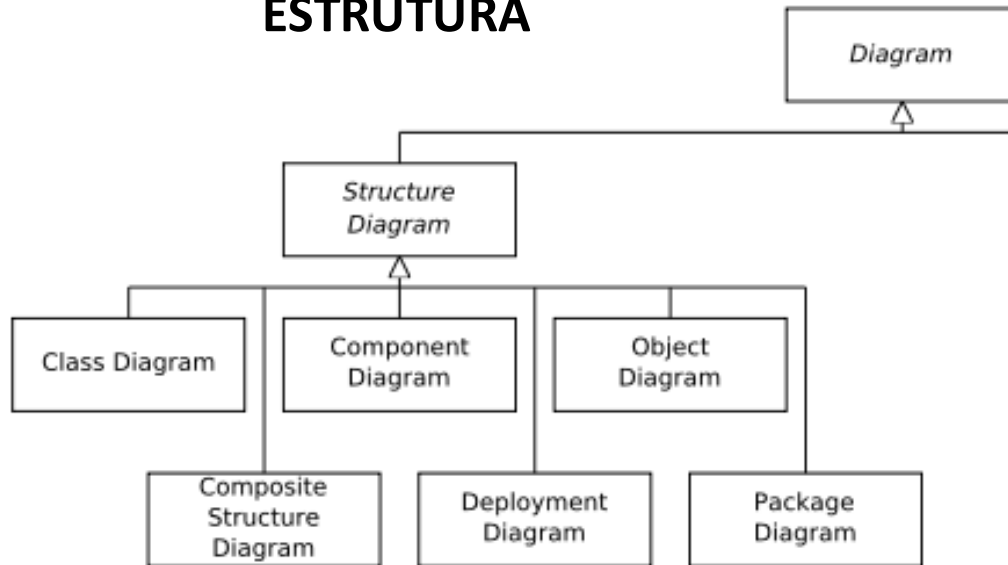
- **Normalizada no âmbito do OMG** (*Object Management Group*)
 - Especificações formais
 - Adequada a diferentes âmbitos de modelação de sistemas



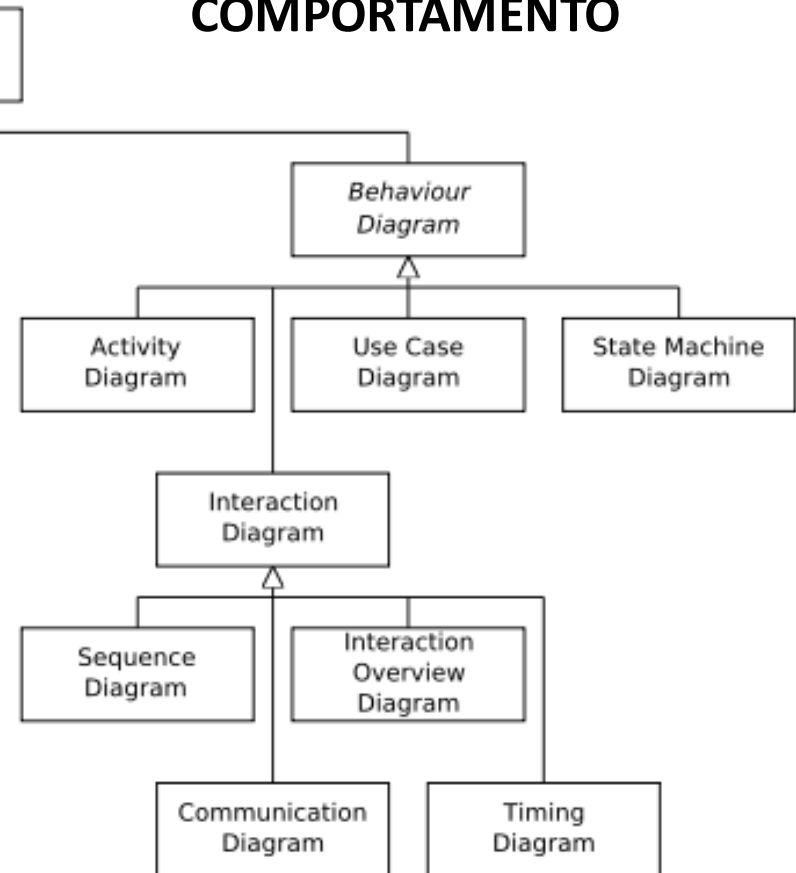
[Rational]

PERSPECTIVAS DE MODELAÇÃO

ESTRUTURA



COMPORTAMENTO



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 Interação (*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**

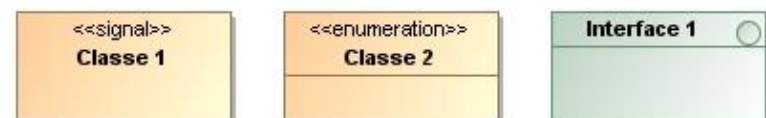
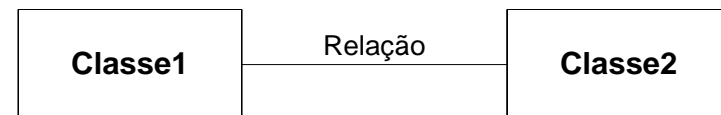
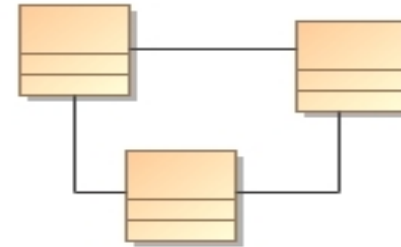
- Diagramas de Implantação

- **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**

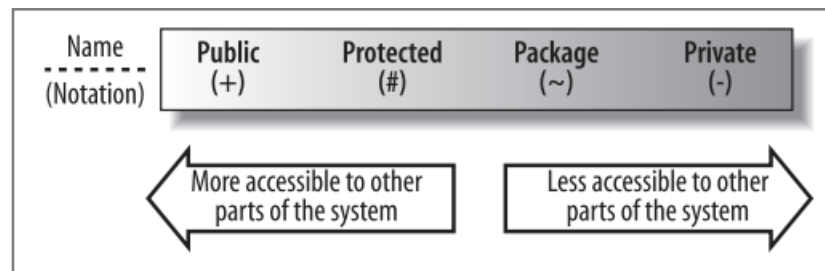
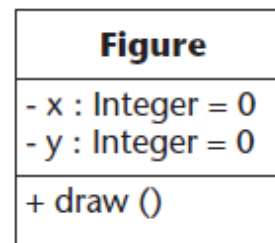
- Representação de estrutura
- Caracterizados por:
 - Designação
 - Tipo
 - Visibilidade
 - Público (+)
 - Privado (-)
 - Protegido (#)
 - Pacote (~)

- Sintaxe:

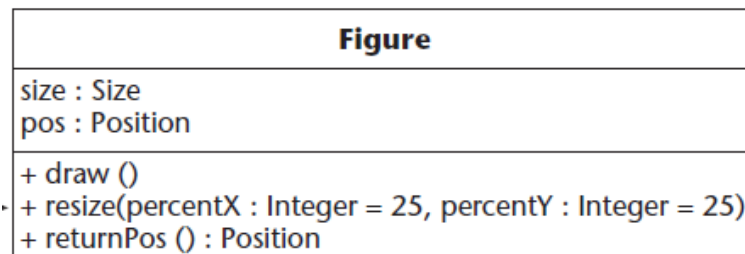
- visibility name:type = init_value
{property_string}

- **Operações**

- Representação de comportamento



[Miles & Hamilton, 2006]



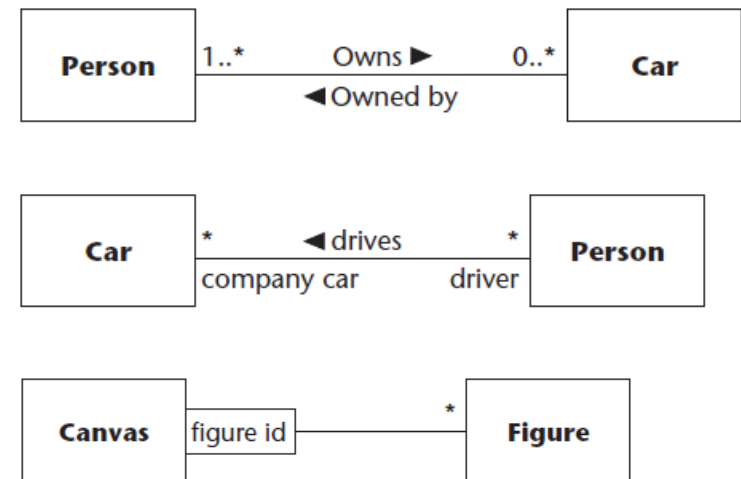
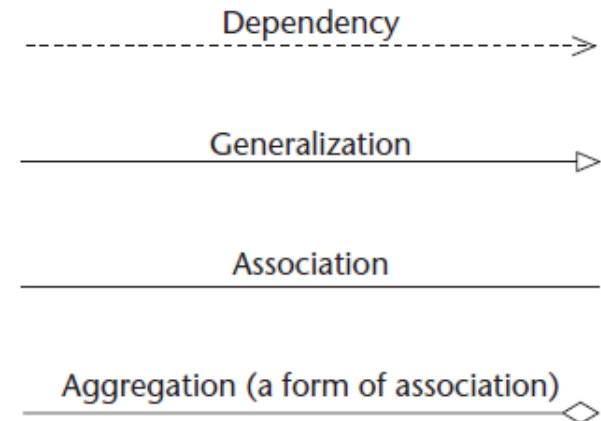
Diagramas de Classes

- **Relações entre classes**

- Dependência
- Associação
- Agregação
- Composição
- Generalização

- **Propriedades das relações**

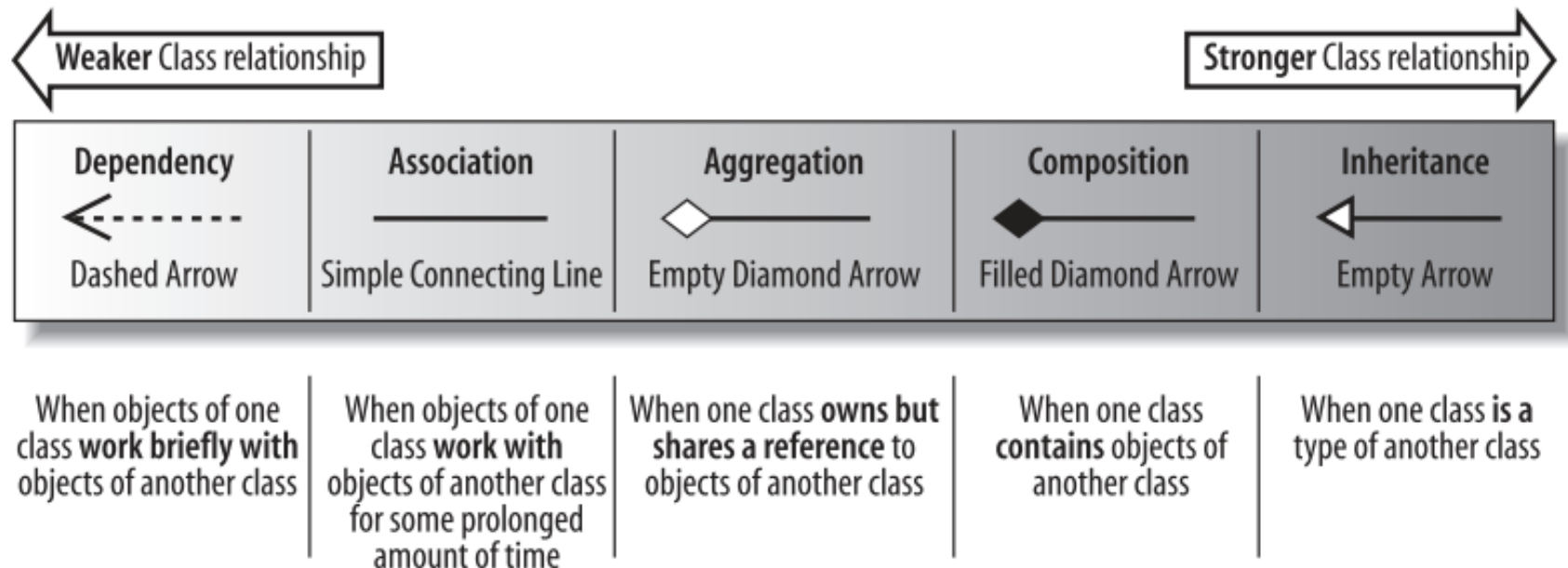
- Direcção
- Multiplicidade
- Papéis
- Qualificação
 - O qualificador de uma associação designa uma chave utilizada para obter um item da colecção respectiva



[Eriksson et al., 2004]

Diagramas de Classes

Relações entre classes e nível de acoplamento



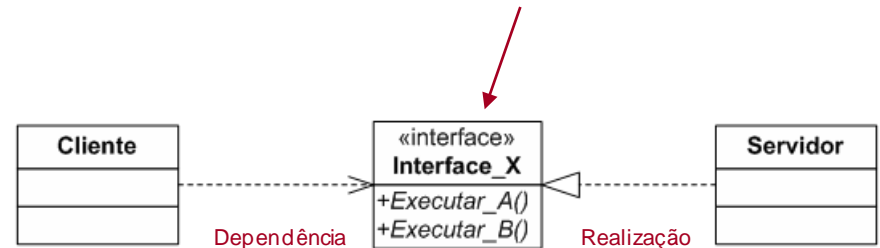
[Miles & Hamilton, 2006]

Diagramas de Classes

• Interface

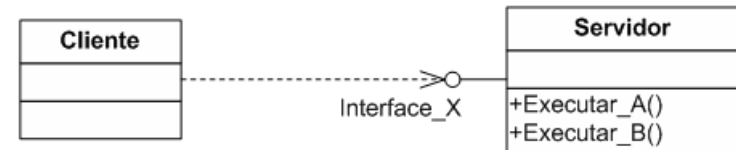
- Classificador que define as características visíveis de uma classe
- Conjunto **coeso** de características
- Define um contrato
 - Não implementa essas características
- **Encapsulamento**

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

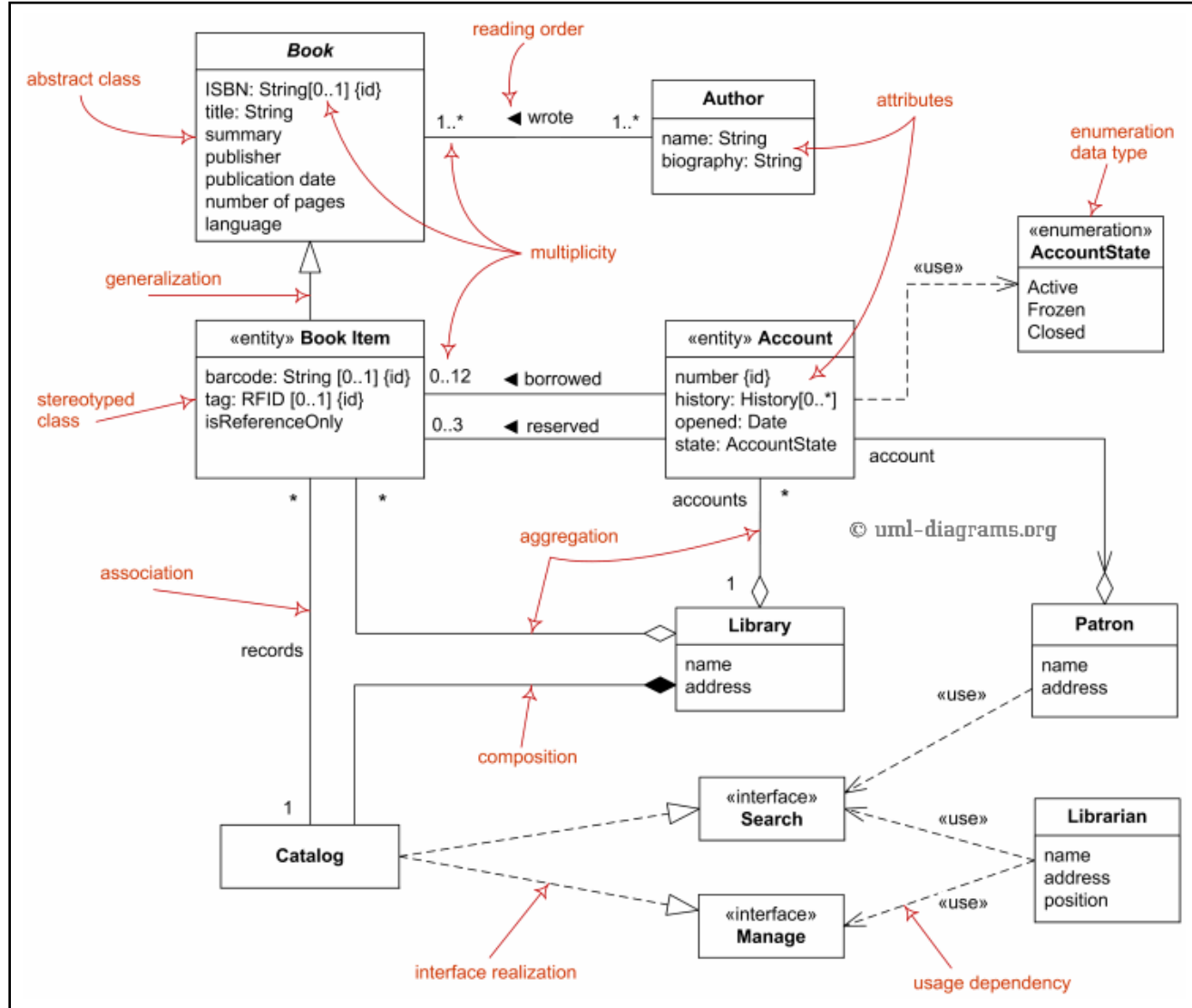


• Realização

- 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



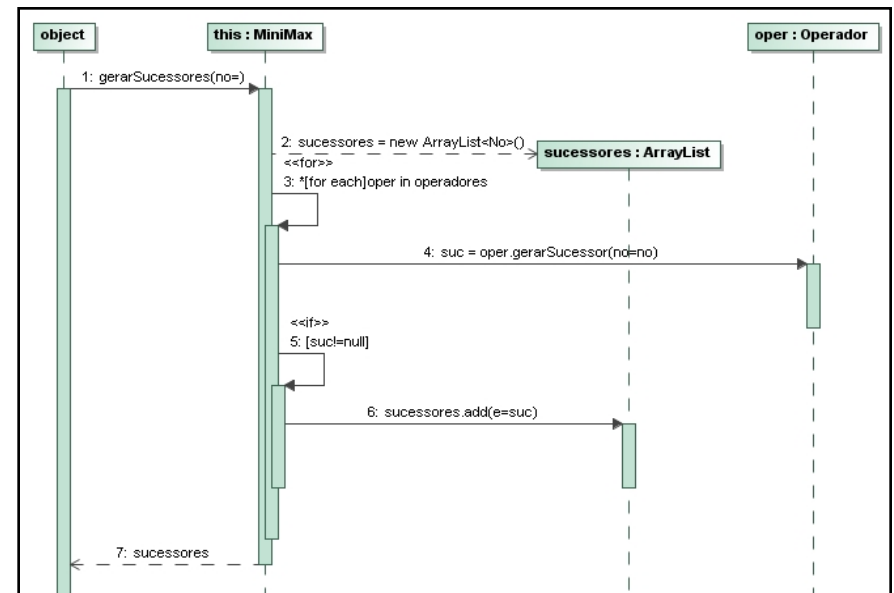
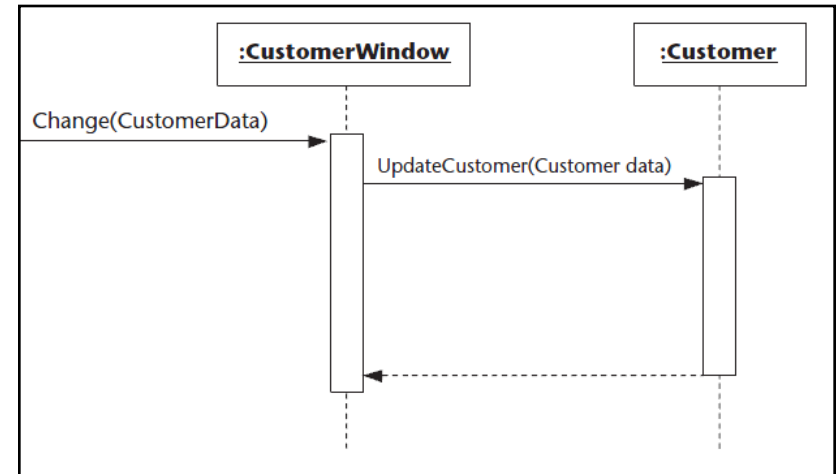
Class Models describe the basic building blocks of object-oriented systems and the relationship between them. [UML 2.0]

Diagramas de Interação

- **Descrevem a interação entre partes do sistema e/ou com o exterior do sistema**
 - **Diagramas de sequência**
 - Ênfase no tempo (sequência de interação)
 - **Diagramas de comunicação**
 - Ênfase na estrutura (ligações entre objectos)
- **Utilização**
 - Actividade de concepção
 - Compreensão e descrição de estrutura e comportamento
 - Compreensão de código legado
 - Actividade de teste
 - Análise de padrões de execução

Diagramas de Sequência

- Representação de interação
 - Descrevem a comunicação entre partes do sistema e/ou com o exterior
 - Ênfase na sequência temporal de interação
- Organização bidimensional
 - Tempo – vertical
 - Estrutura (partes) – horizontal
- Elementos de modelação
 - Linha de vida (*lifeline*)
 - Representa evolução temporal
 - Foco de activação (*activation bar*)
 - Representam execução de operações
 - Mensagem
 - Partes trocam mensagens
 - Operador



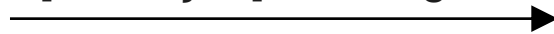
Diagramas de Sequência

- Mensagens

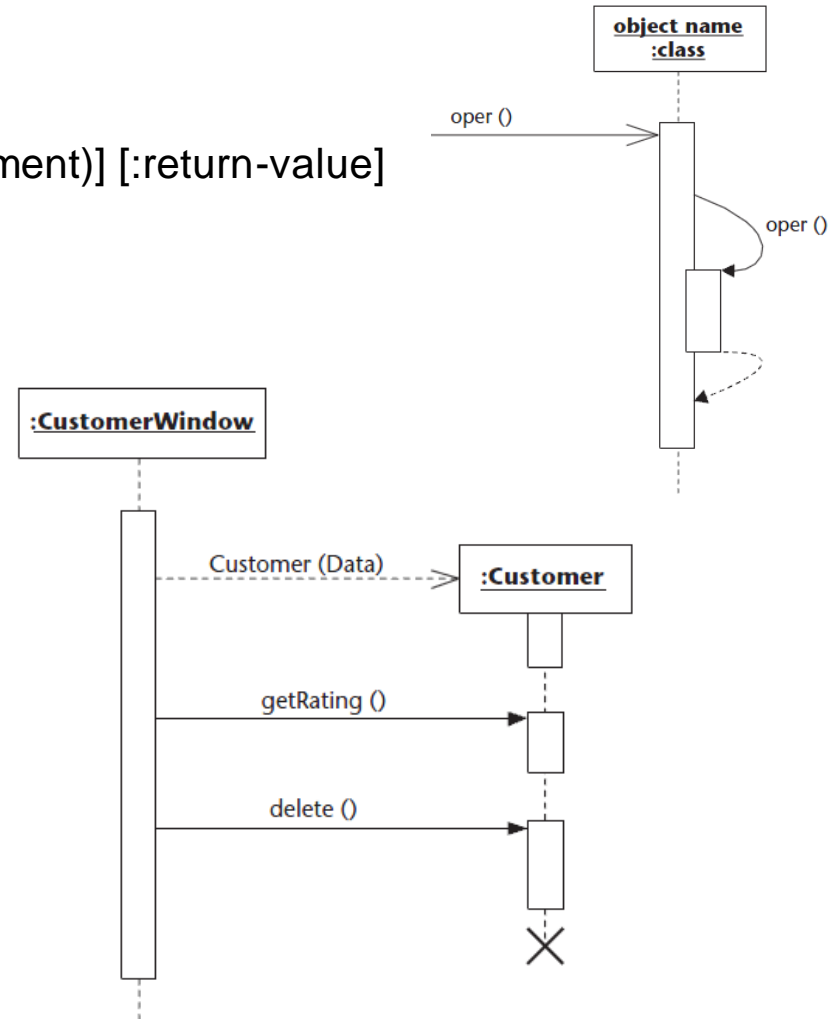
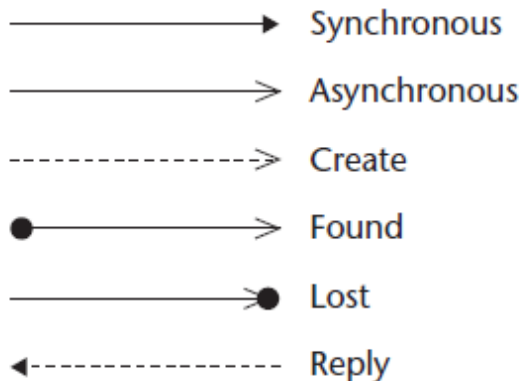
- Sintaxe

- [attribute=] message-name [(argument)] [:return-value]

[Condição] Mensagem



- Tipos de mensagens



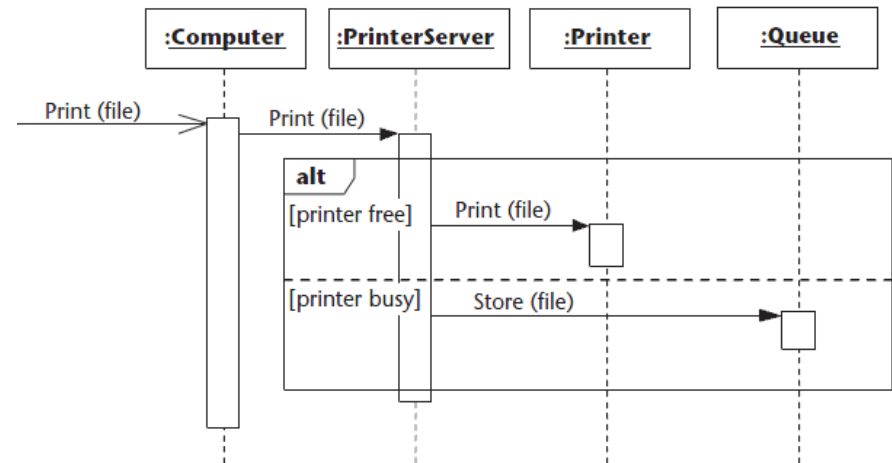
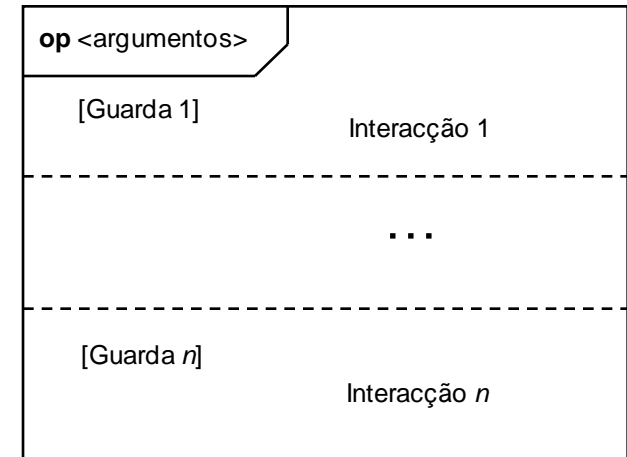
Diagramas de Sequência

- **Operador**

- Fragmento de interação com semântica específica

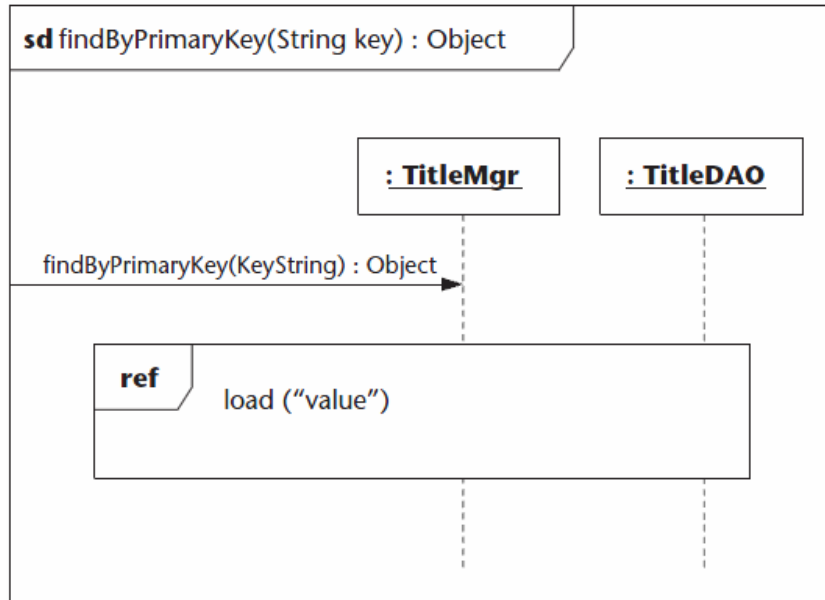
- **Tipos de operadores**

- **ref**: referência a fragmento de interação
- **loop**: repetição de fragmento de interação
- **break**: fim de repetição de fragmento de interação
- **alt**: selecção de fragmento de interação
- **par**: regiões concorrentes (paralelas)
- **assert**: fragmento de interação requerido
- **opt**: fragmento de interação opcional
- **neg**: especificação negativa (não pode acontecer)
- **region**: região crítica (não são permitidas outras mensagens)

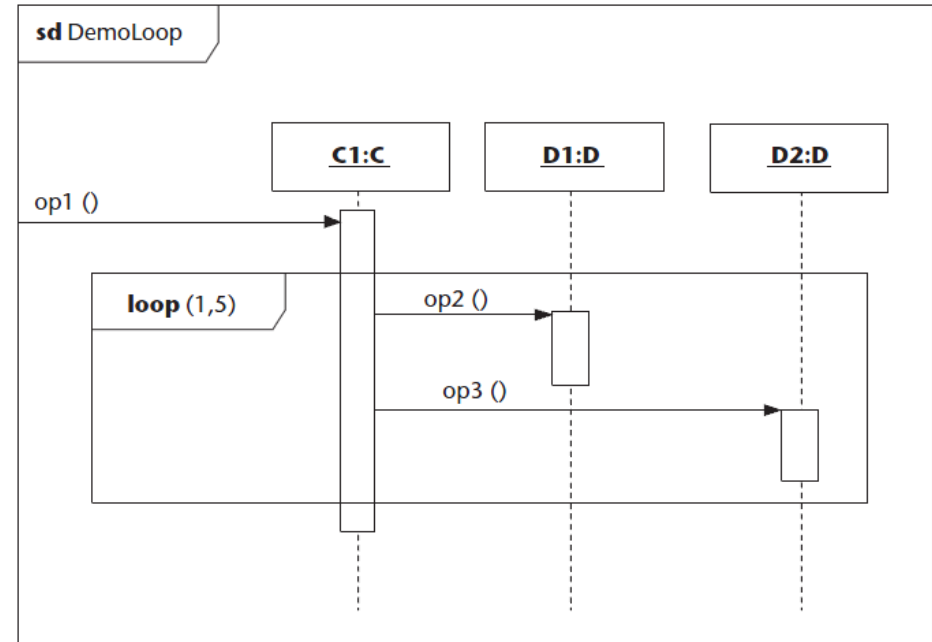


Diagramas de Sequência

Operador **ref**

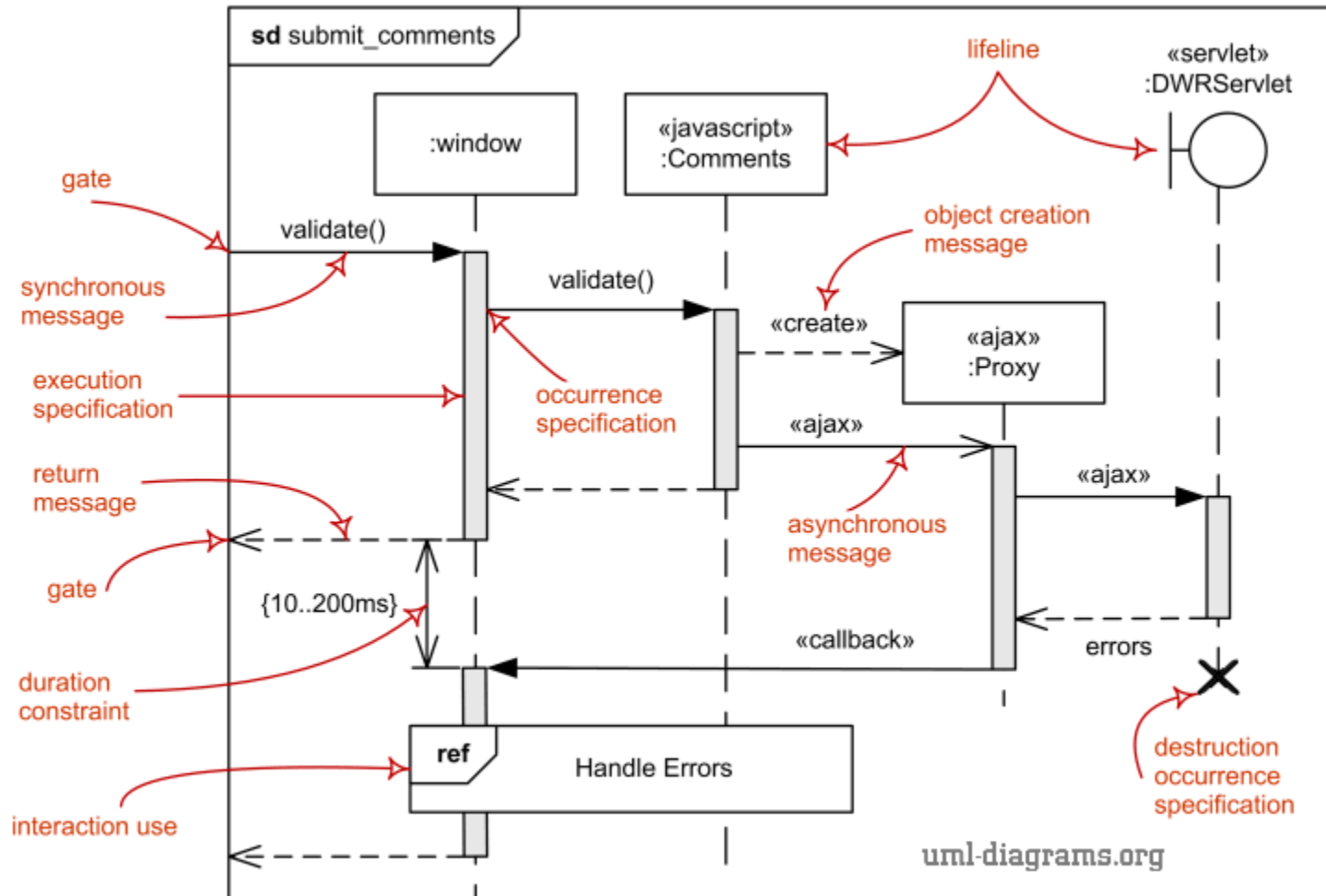


Operador **loop**



[Eriksson et al., 2004]

Diagramas de Sequência

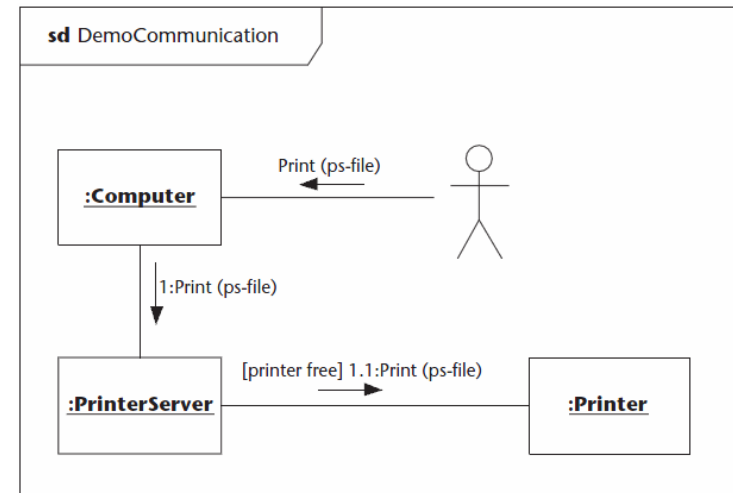


Sequence Diagrams show the communication behavior between parts of a system. A sequence diagram can be a behavior specification or a trace of actual runtime communication. [UML 2.0]

Diagramas de Comunicação

- Representação de interacção

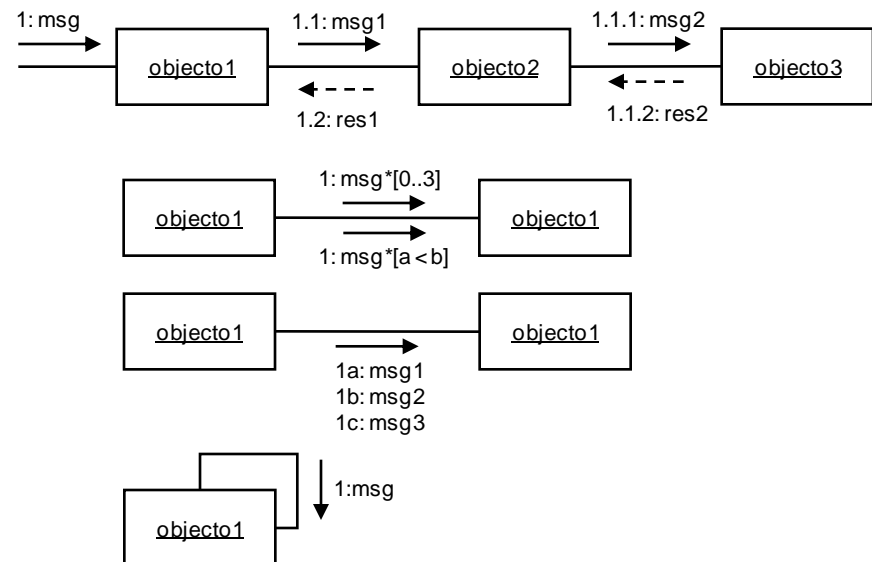
- Descrevem a comunicação entre partes do sistema e/ou com o exterior
- Ênfase nas ligações entre partes (estrutura)
- Semanticamente equivalentes aos diagramas de sequência
- São uma extensão dos diagramas de objectos



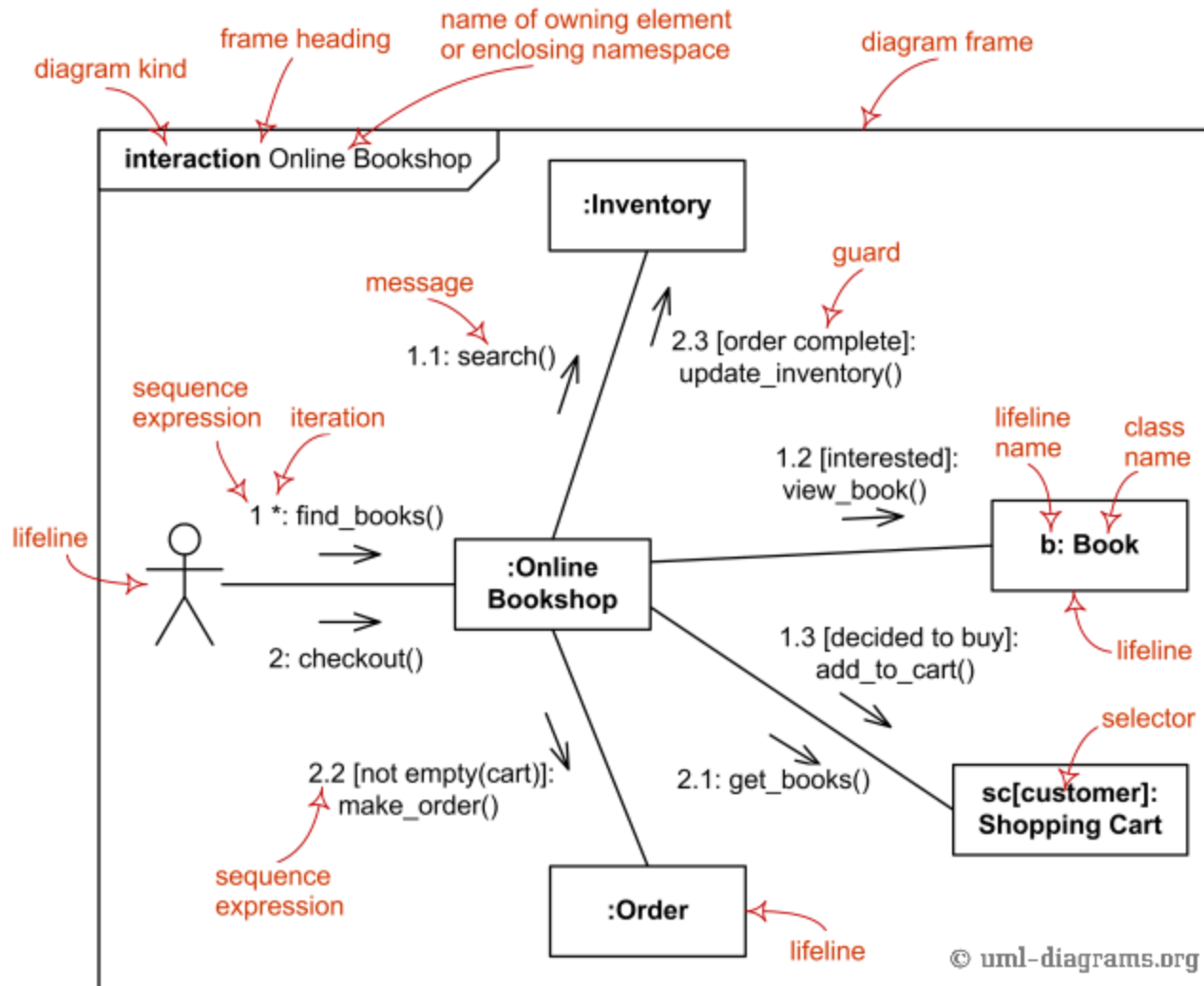
[Eriksson et al., 2004]

- Notação das mensagens

- Mensagens encadeadas
- Mensagens iterativas
- Mensagens paralelas
- Mensagens próprias



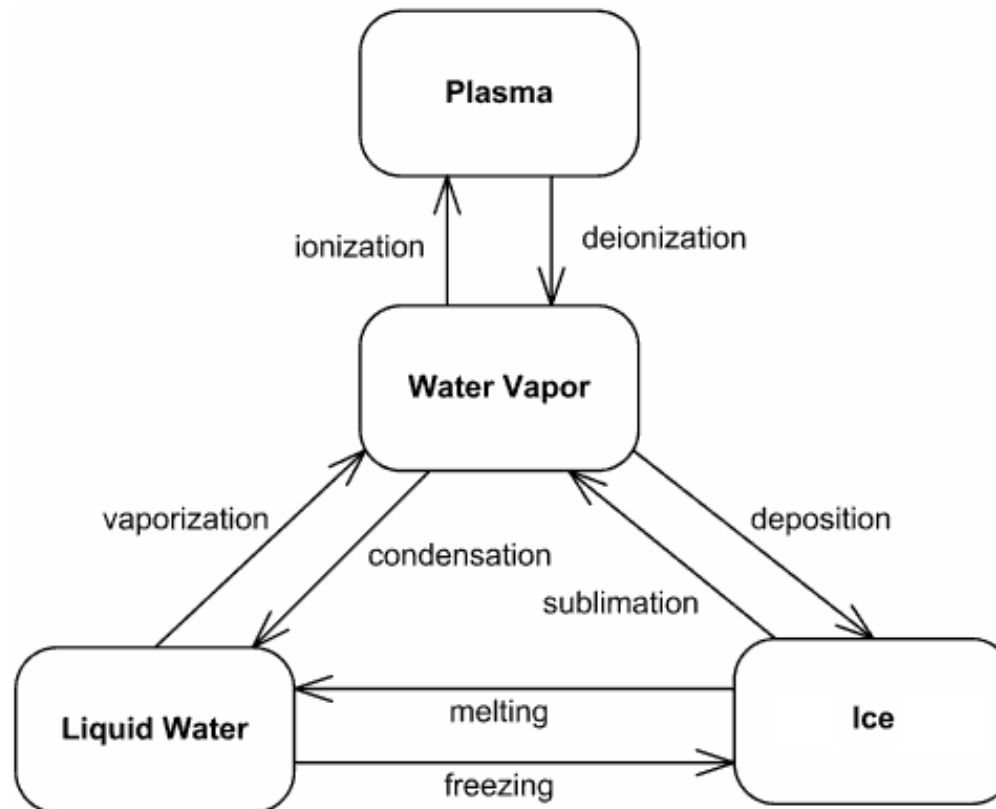
Diagramas de Comunicação



Communication Diagrams show communications in a system, emphasizing the structure and communication paths. [UML 2.0]

Modelos de Dinâmica de um Sistema

- **Evolução da configuração e propriedades** das partes e relações entre partes de um sistema em função do **tempo** (espaço) e da **interacção** com o exterior
 - Exemplo: variáveis que se alteram ao longo do tempo com a evolução do sistema



Diagramas de Transição de Estado

- **Estado**

- Configuração da estrutura de um sistema discriminadora do comportamento do sistema
 - Representa situações relevantes da evolução de um sistema

- **Transição**

- Acontecimento através do qual o sistema evolui do estado actual para um novo estado

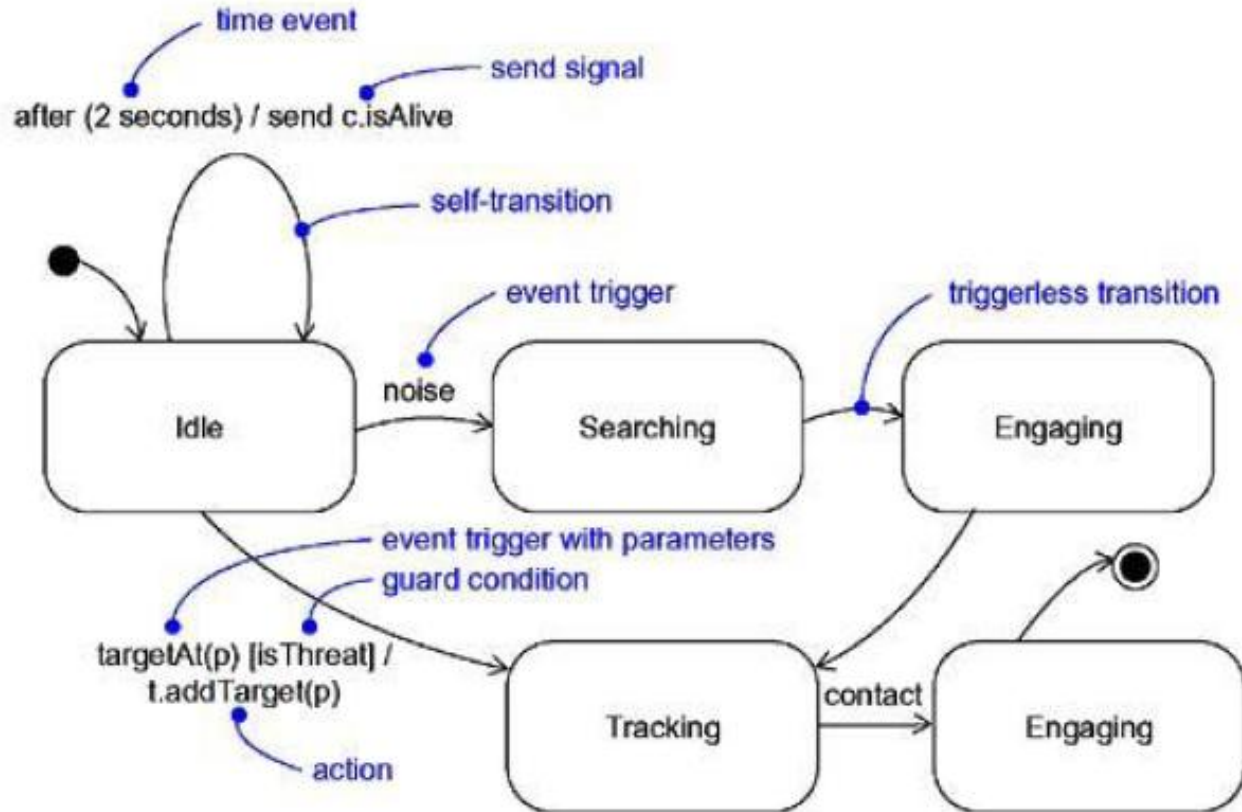
- **Evento**

- Ocorrência no tempo e no espaço com significado para o sistema

- **Acção**

- Resposta do sistema a acontecimentos
 - Diferentes tipos: afectação de variáveis, evocação de funções, geração de eventos, etc.

Diagramas de Transição de Estado



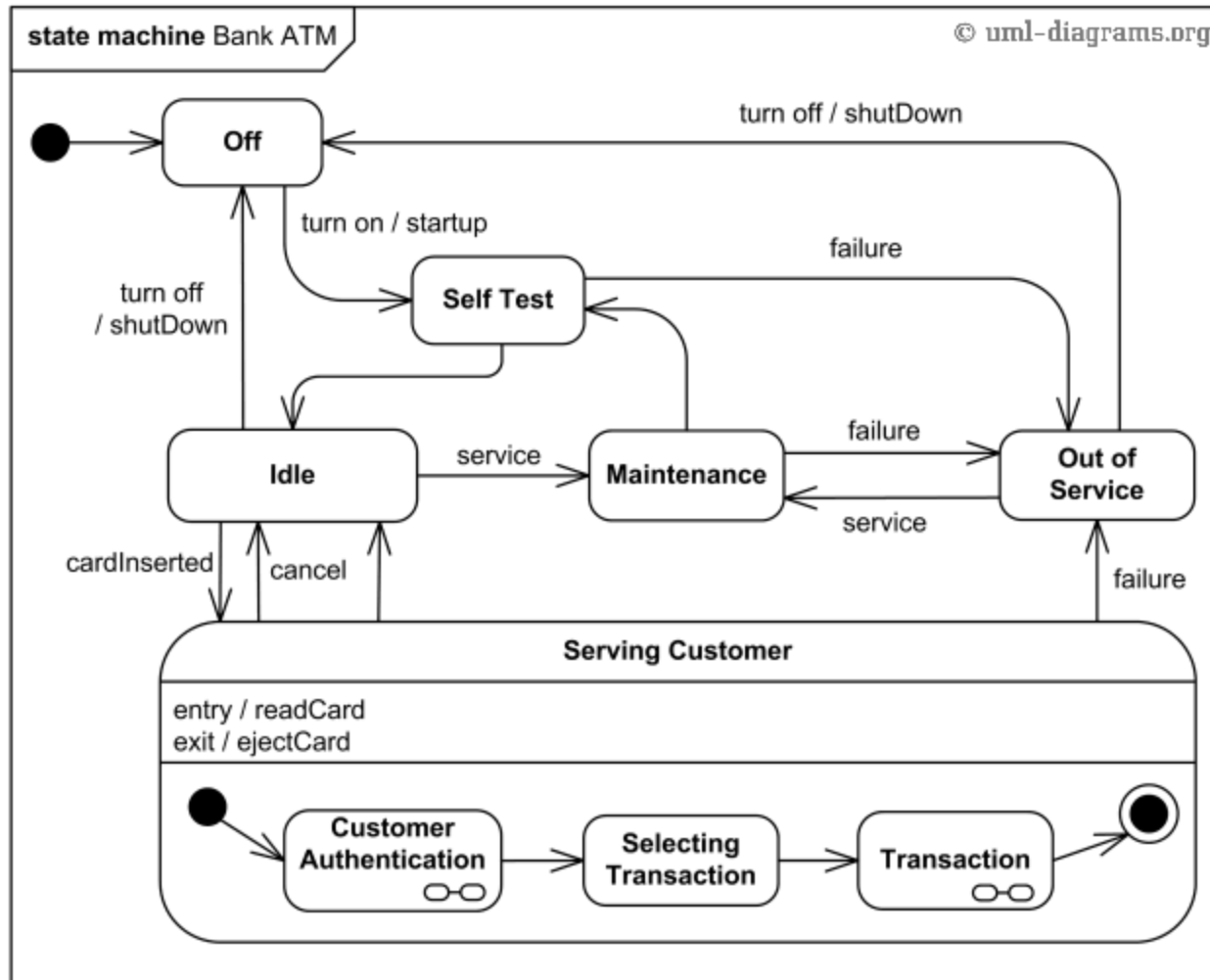
Event Trigger: an event is the specification of a significant occurrence that has a location in time and space.

Guard Condition: a guard condition is a Boolean expression that determines if the state transition occurs for some event.

Action: an action is an executable atomic computation. Actions may include operation calls (to the object that owns the state machine, as well as to other visible objects), the creation or destruction of another object, or the sending of a signal to an object.

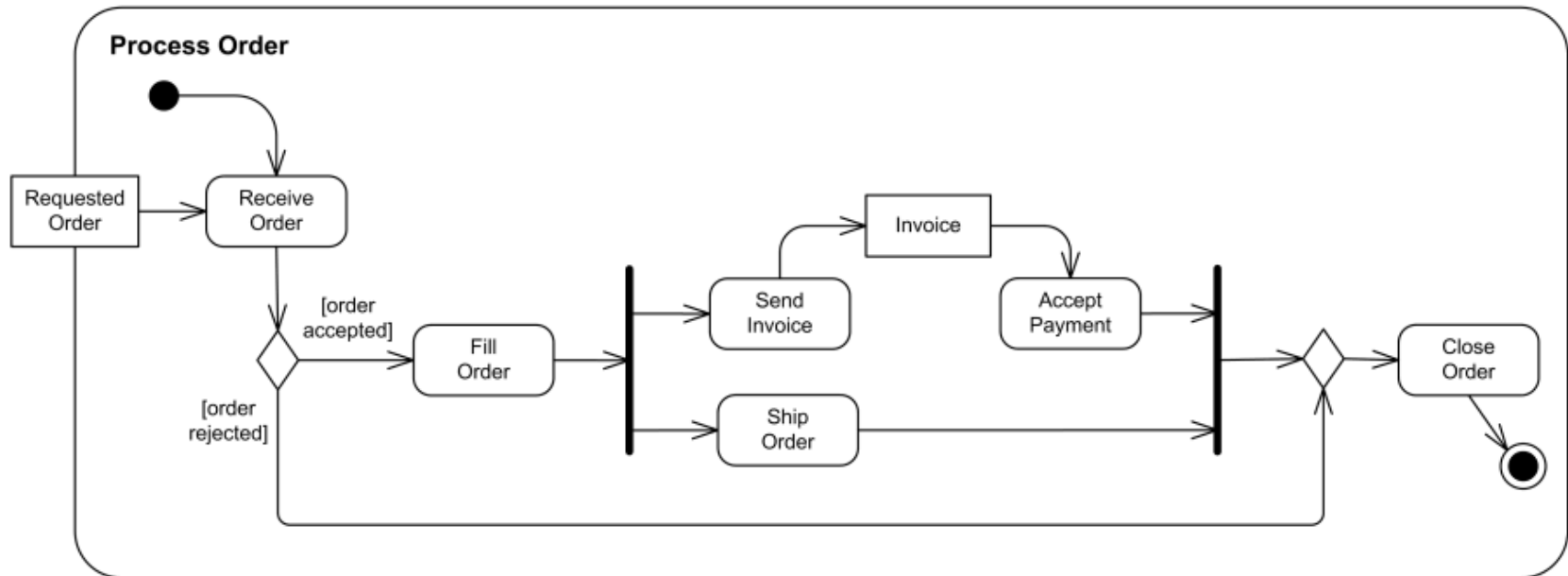
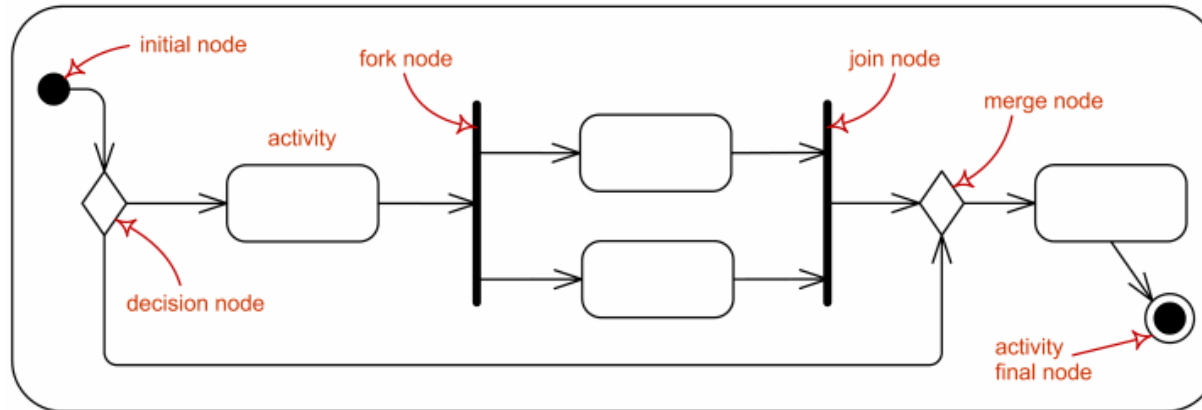
[UML User Guide, Booch *et al.* 1998]

Diagramas de Transição de Estado



State Machines model discrete behavior through finite-state transition systems, at any level of granularity (classes, subsystems, etc.)[\[UML 2.0\]](#)

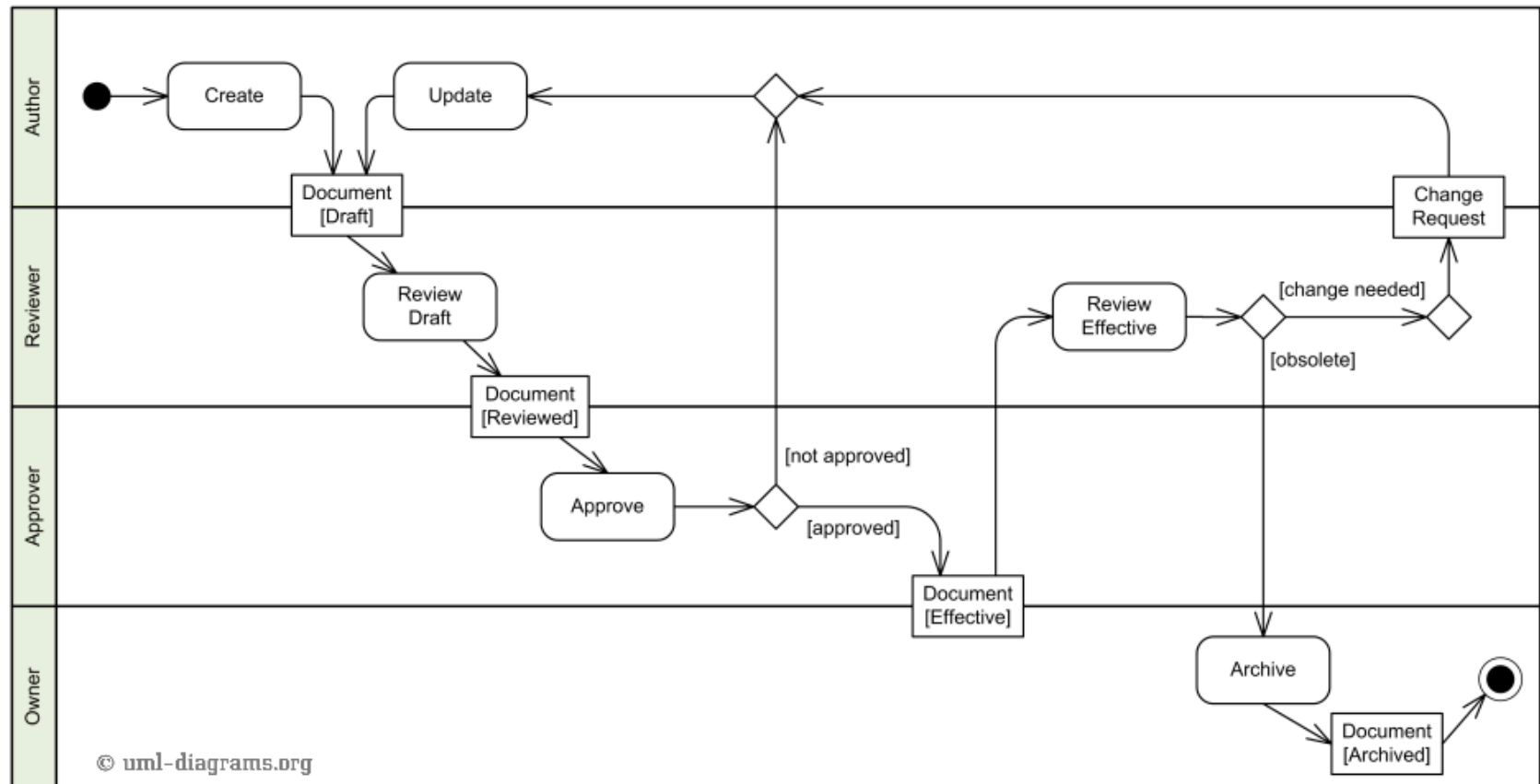
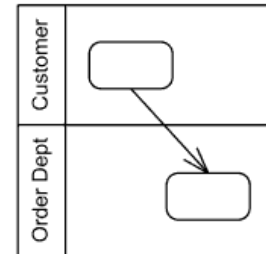
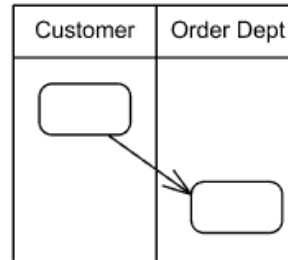
Diagramas de Actividade



Activity Modeling emphasizes the sequence and conditions for coordinating lower level behavior, rather than whose parts own those behaviors. [UML 2.0]

Diagramas de Actividade

Partições



© uml-diagrams.org

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.