41951- ANÁLISE DE SISTEMAS

Visual Modelling with UML

Ilídio Oliveira | v2022/03/08

universidade de aveiro
departamento de eletrónica,
telecomunicações e informática

deti

Learning objectives for this lecture

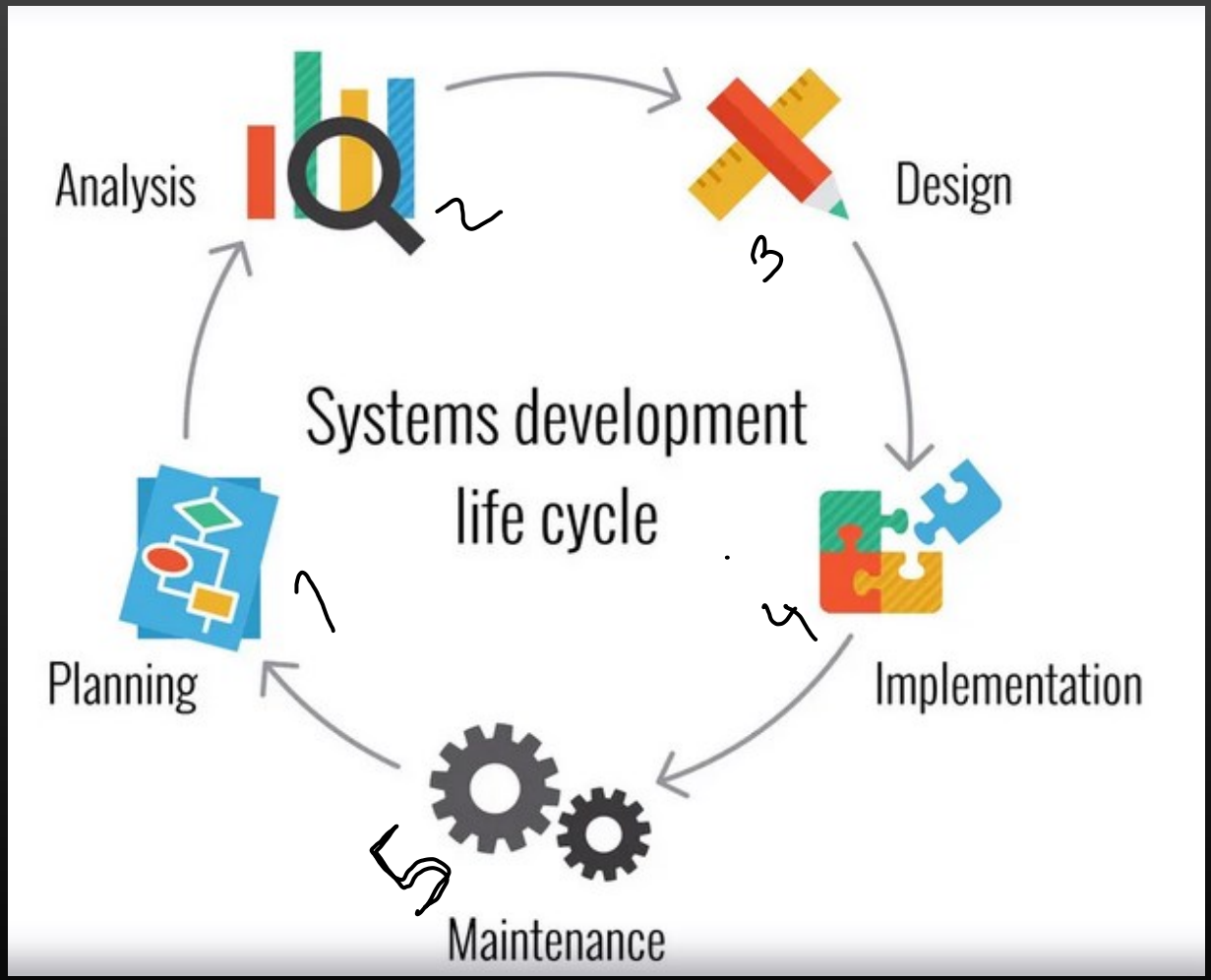
Justify the use of models in systems engineering

Enumerate advantages of visual models

Explain the organization of the UML

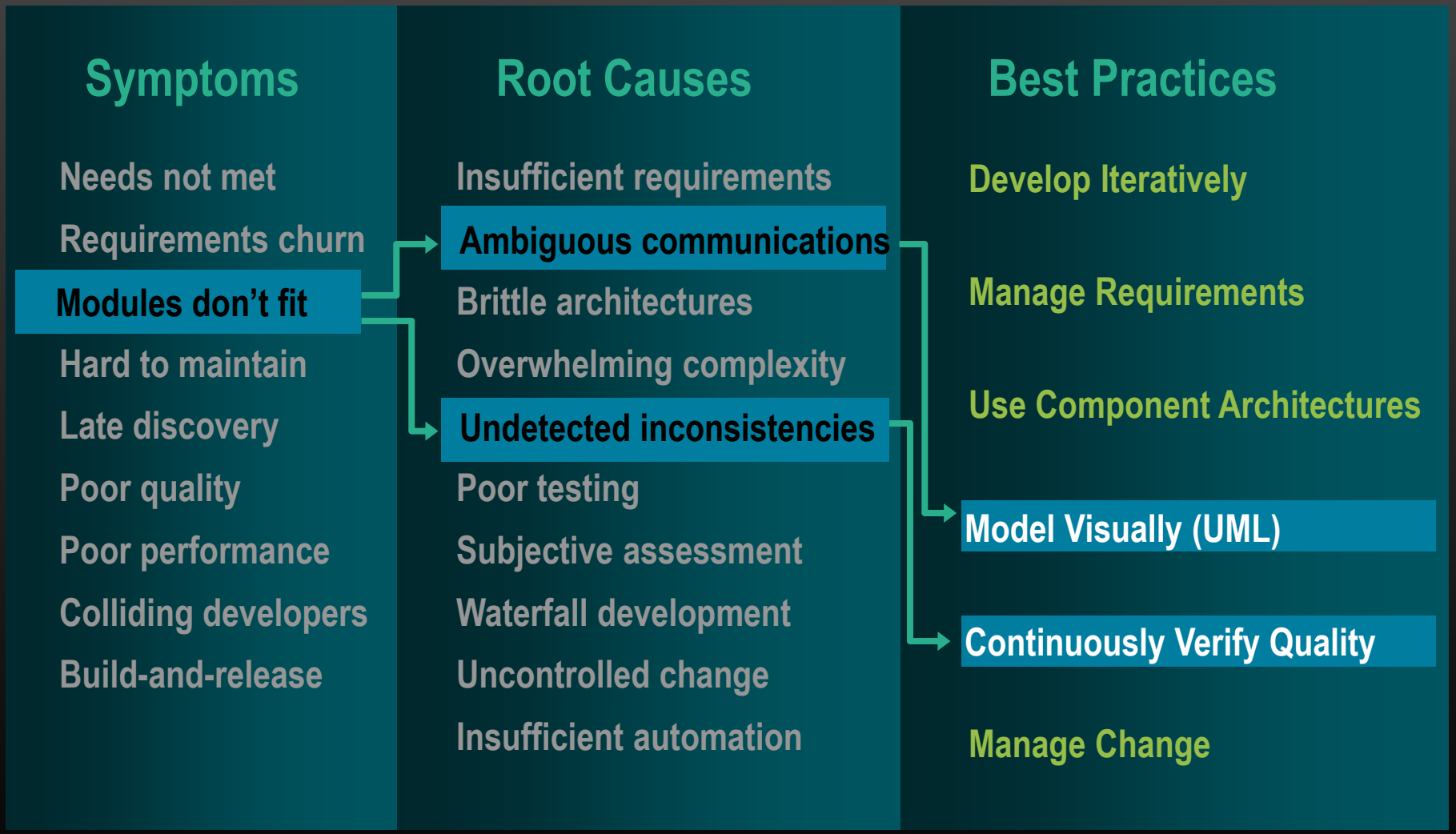
Identify the main diagrams in UML and their modeling viewpoint

Systems development lifecycle (SDLC)



Problems and solutions in the SDLC (a Rational Unified Process perspective)

problemas e soluções podem ser combinadas



Modeling

UML as a visual specification language

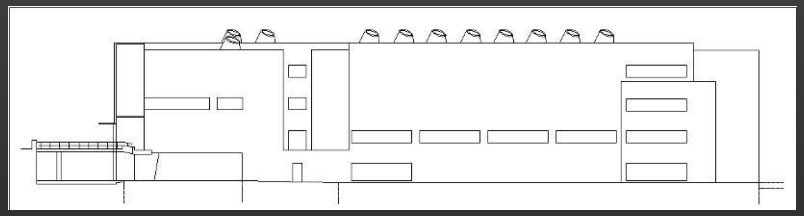
universidade de aveiro
departamento de eletrónica,
telecomunicações e informática

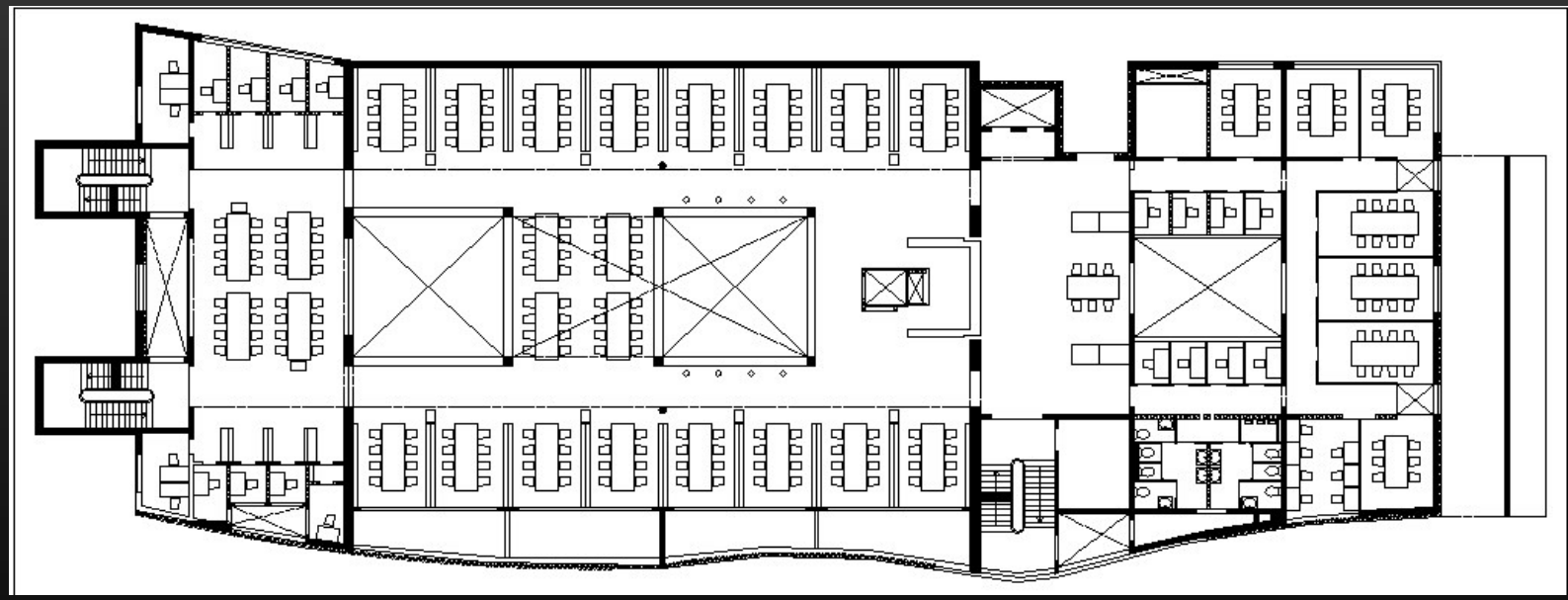
deti

Usamos modelos visuais para captar partes do mundo/realidade

- Uma linguagem comum (escrever, ler)
- Especificações visuais são mais inteligíveis
- Compor: aplicar talento e disciplinas técnicas
- Orquestra: a prova que os modelos funcionam!

Um modelo é uma simplificação da realidade





Os modelos ajudam a gerir a complexidade

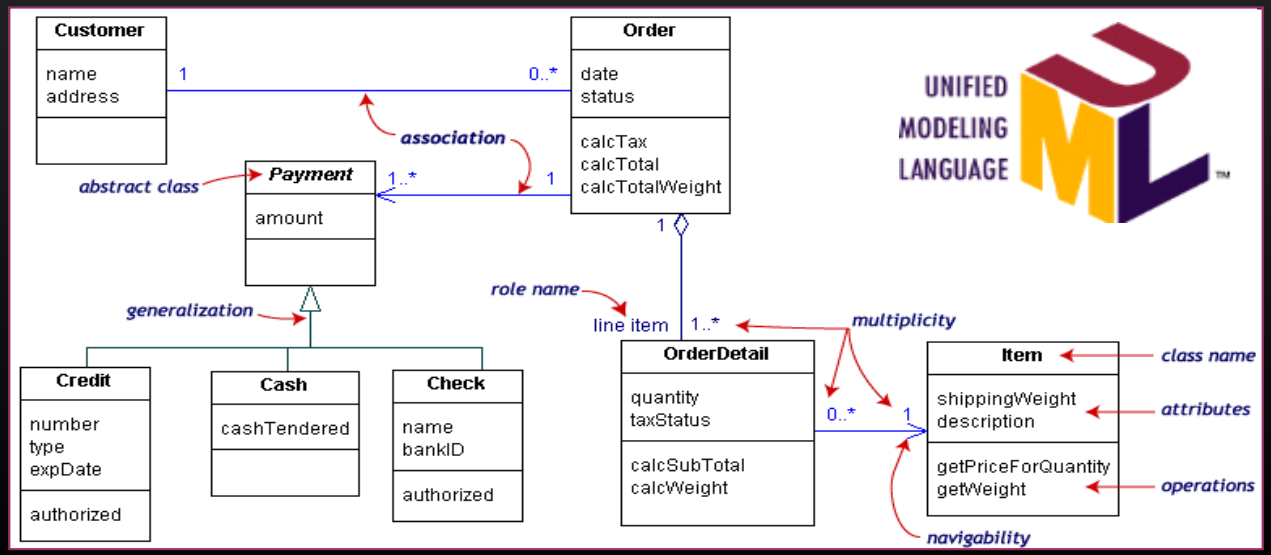
4 razões para usar modelos (G. Booch):

Ajudar a **visualizar** um sistema (*high-level*)

Especificar/documentar a **estrutura e o comportamento** do sistema (antes de implementar)

Serve como **referência** para orientar construção (“planta”)

Documentar as decisões (de desenho) que foram feitas



Modelação visual no desenvolvimento

UML 2: Unified Modeling Language

Linguagem de modelação normalizada

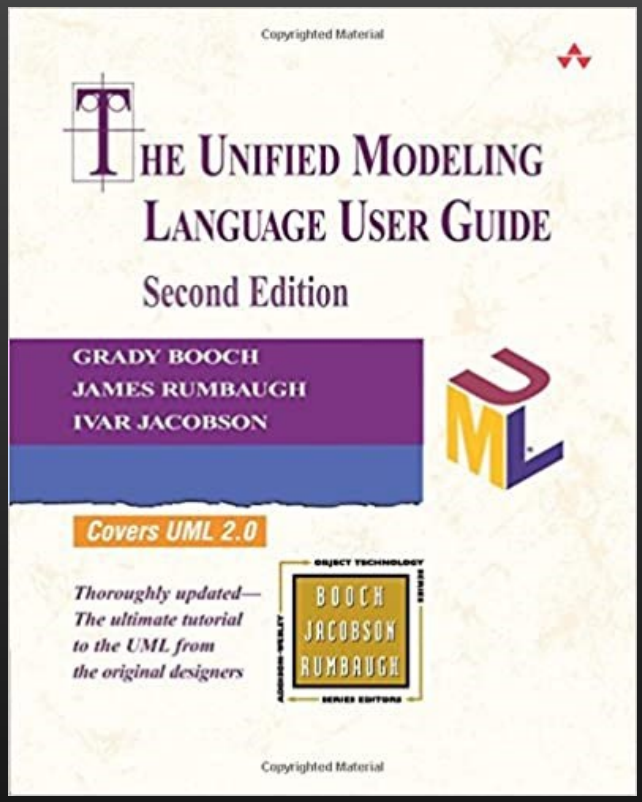
Benefícios

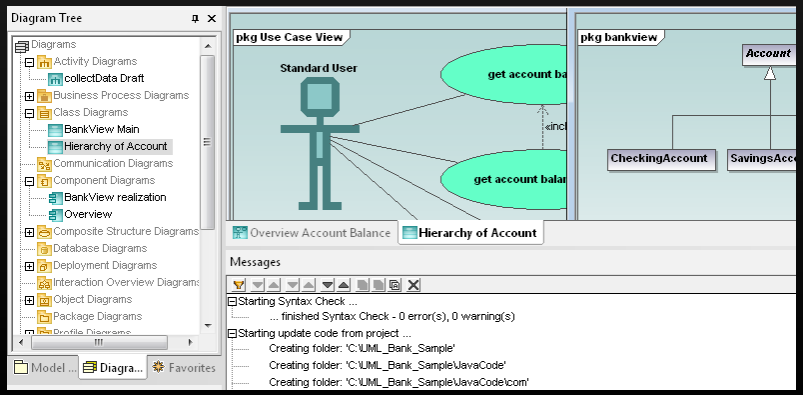
Promover a comunicação mais clara e sucinta

Manter o desenho (planeamento) e a implementação (construção) coerentes

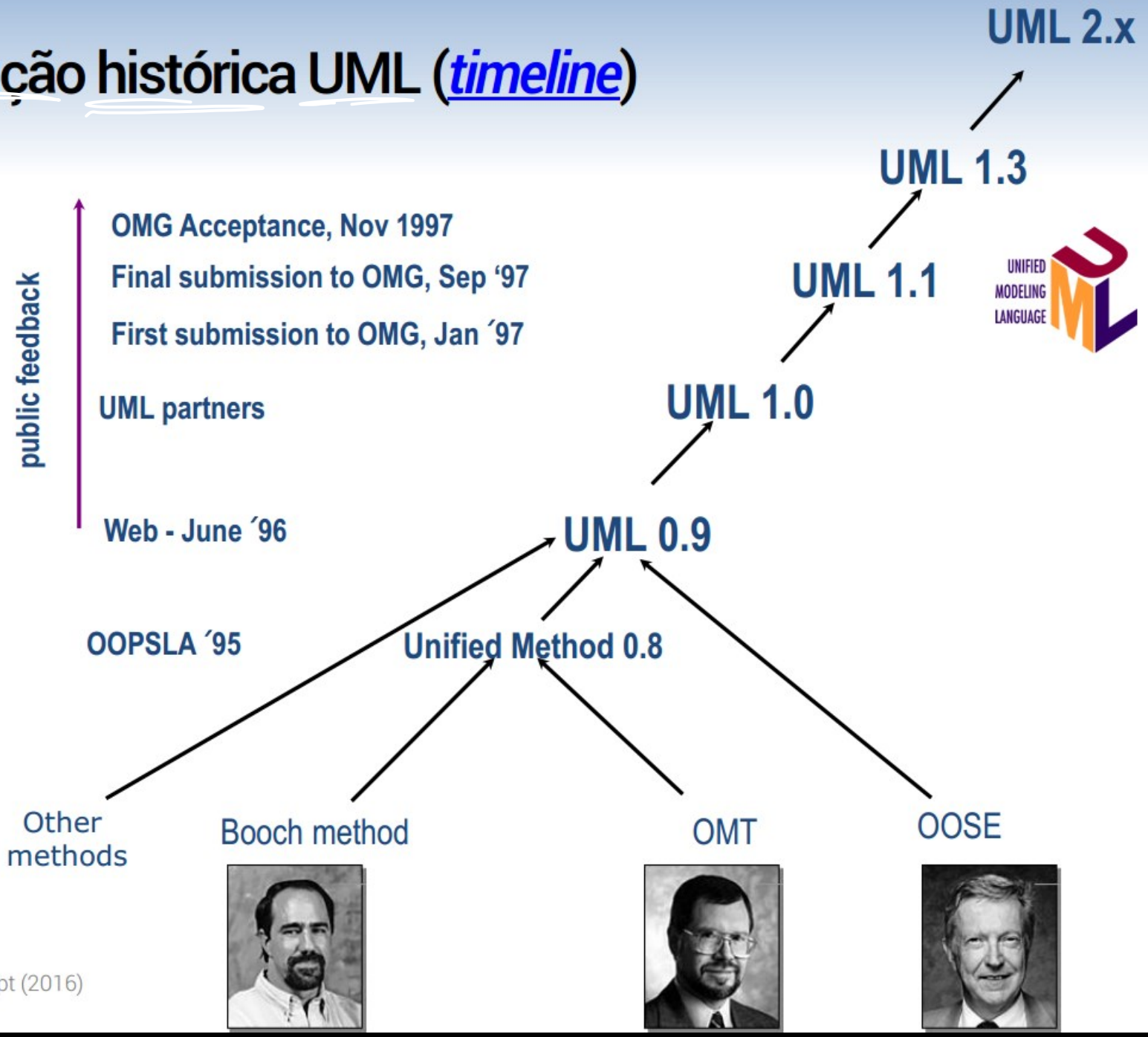
Mostrar ou esconder diferentes níveis de detalhe, conforme apropriado

Pode suportar, em parte, processos de construção automática (gerar a solução a partir do modelo)





Evolução histórica UML (*timeline*)

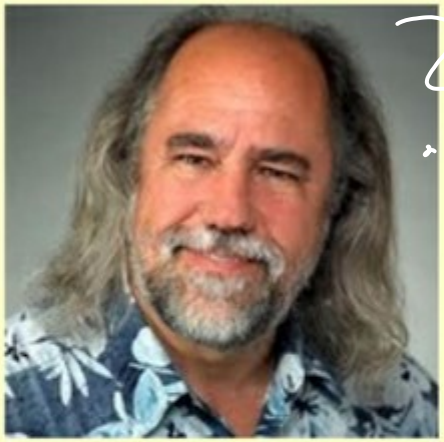


"Three Amigos"

I. Jacobson: thinking in use-cases...

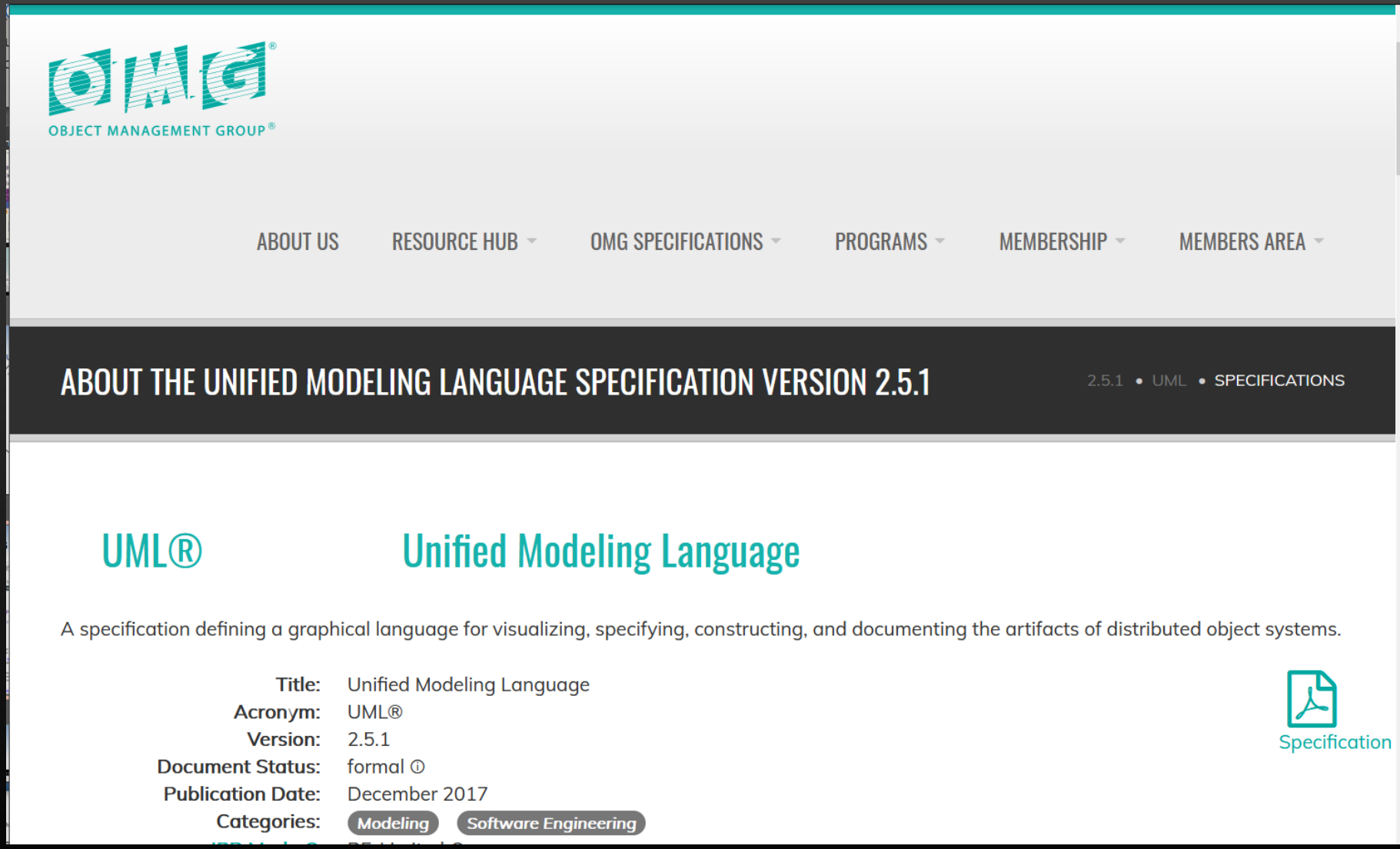
G. Booch: thinking in OO development process...

J. Rumbauch: thinking in classes modeling...



The UML effort started officially in October 1994, with the version 0.8 draft being released in October 1995. The Three Amigos, as they are sometimes referred to (Booch, Rumbauch, and Jacobson) had successfully unified semantics and notation, ultimately meaning that users could focus on their own work and worry less about the specifics of a given method.

UML é uma especificação do OMG



Também reconhecida como um standard internacional ISO

UML está adotada globalmente

Français | Русский Members area

Standards About us Standards Development News Store

Search ISO

Standards catalogue Online collections Graphical symbols

ISO Store › Store › Standards catalogue › By TC › JTC 1 Information technology › SC 7

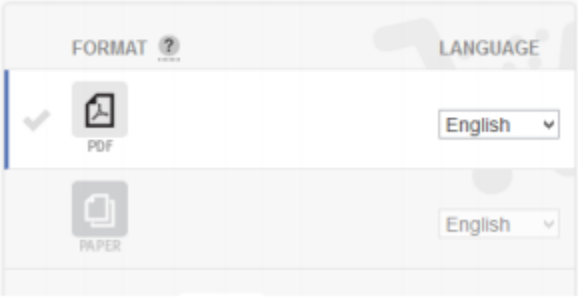
ISO/IEC 19505-1:2012

Information technology -- Object Management Group Unified Modeling Language (OMG UML) -- Part 1: Infrastructure

Abstract

Preview ISO/IEC 19505-1:2012

ISO/IEC 19505-1:2012 defines the Unified Modeling Language (UML), revision 2. The objective of UML is to provide system architects, software engineers, and software developers with tools for analysis, design, and implementation of software-based systems as well as for modeling business and similar processes.



Aplicações principais da UML

Análise e desenho de sistemas de software

Estrutura e comportamento de sistemas baseados em software

- Elementos do modelo representam entidades do mundo do software

Especialmente adequada para o desenvolvimento por objetos (*object-oriented*)

Domínio do problema (processos de trabalho,...)

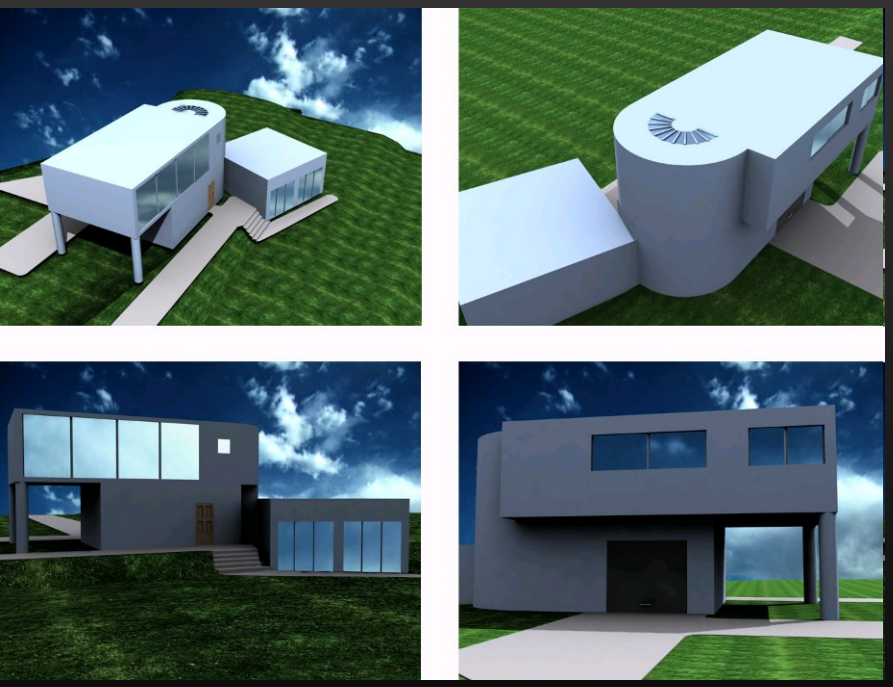
Especificar ou documentar o domínio de aplicação/negócio

- Elementos do modelo representam entidades do negócio

Não implica ou assume uma implementação em software

Domínio: área de aplicação

Não há uma vista única, mas várias e complementares



Para que serve o sistema?

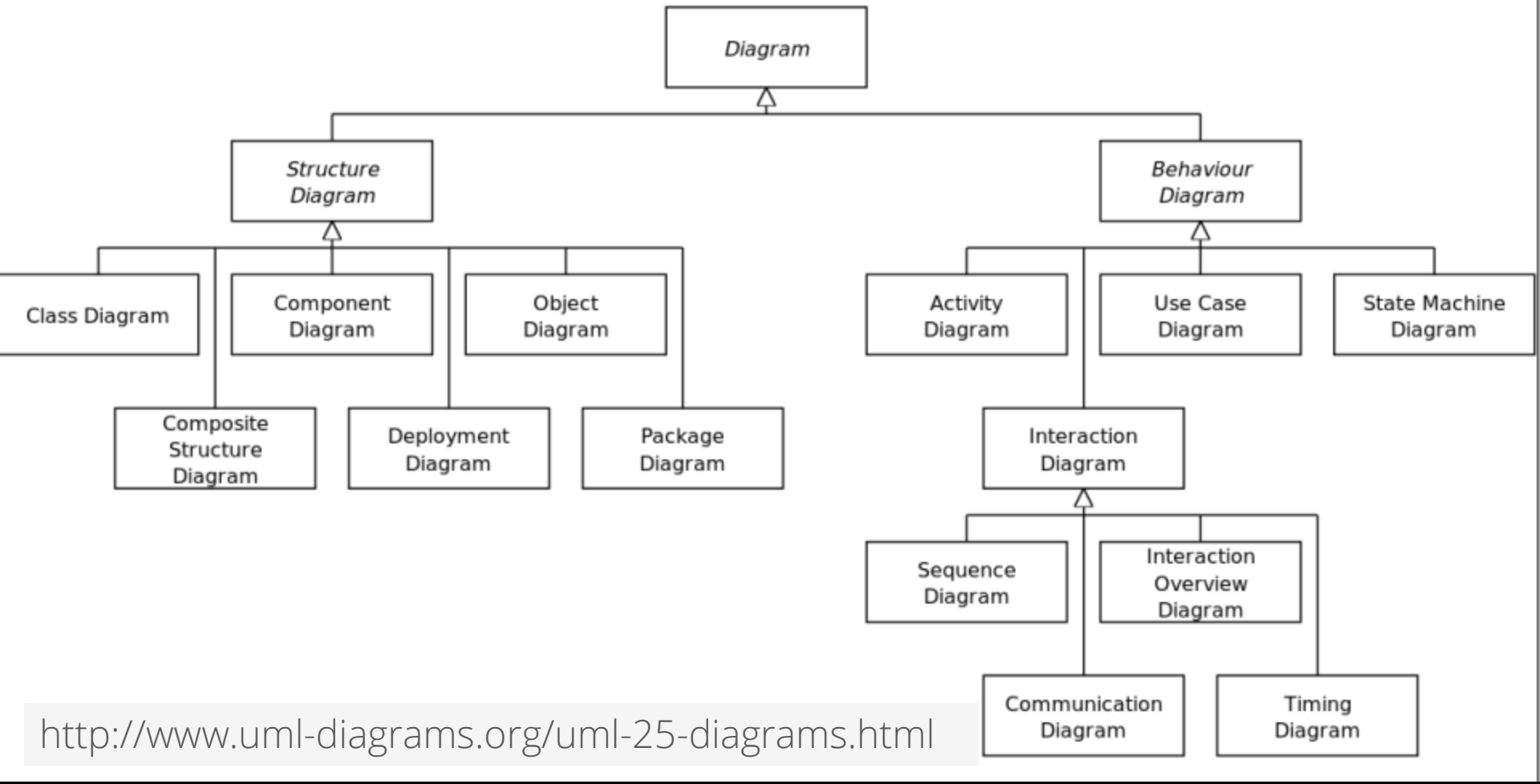
Quais são as estruturas de informação?

Decomposição funcional de atividades complexas

Visualizar a organização do software em partes e as suas interações

Etc.

Diagramas da UML 2.x

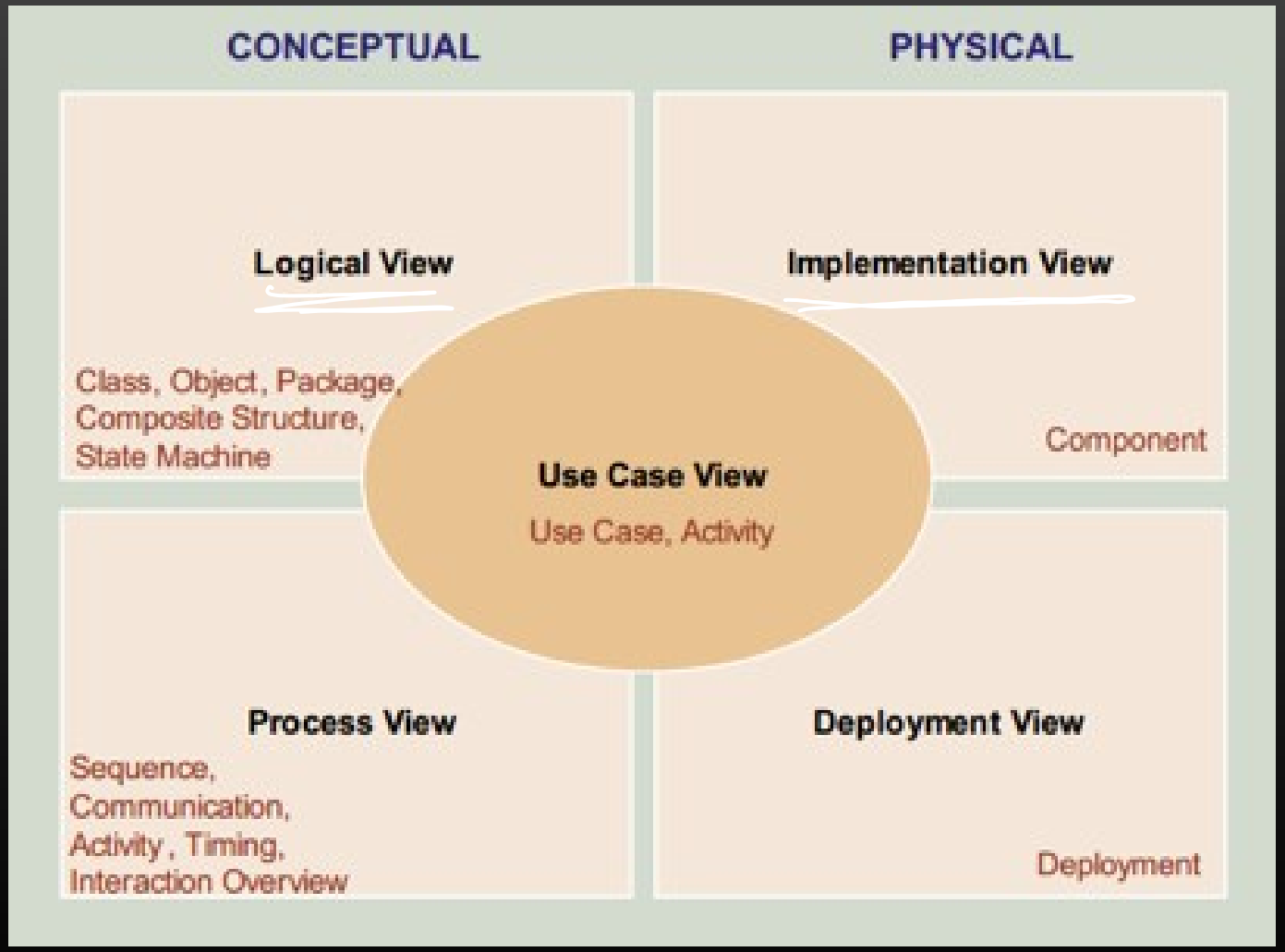


http://www.uml-diagrams.org/uml-25-diagrams.html

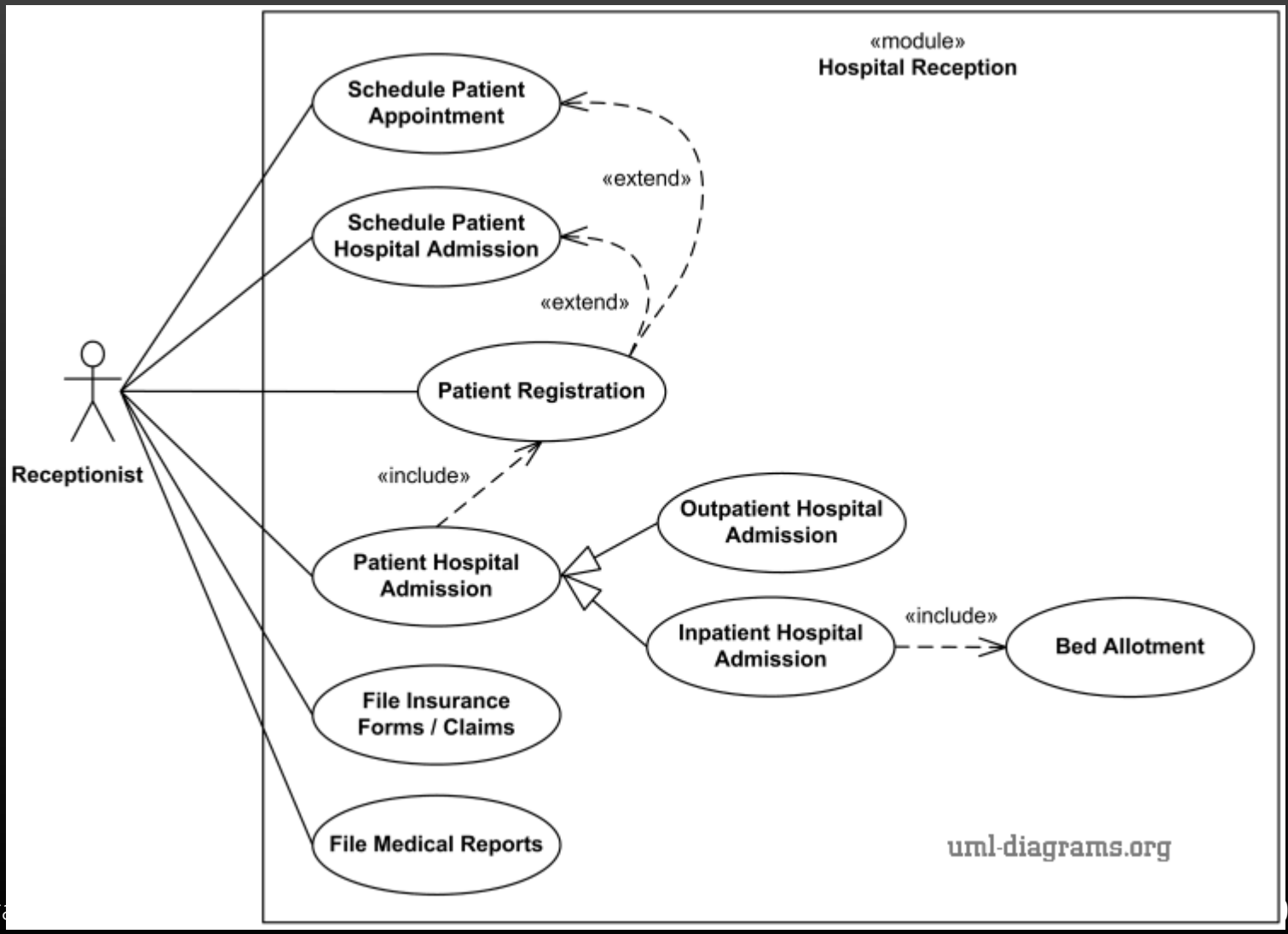
APLICAÇÃO DA UML AO LONGO DO PROCESSO DE DESENVOLVIMENTO

ILÍDIO OLIVEIRA ico@ua.pt
v2017-06-02

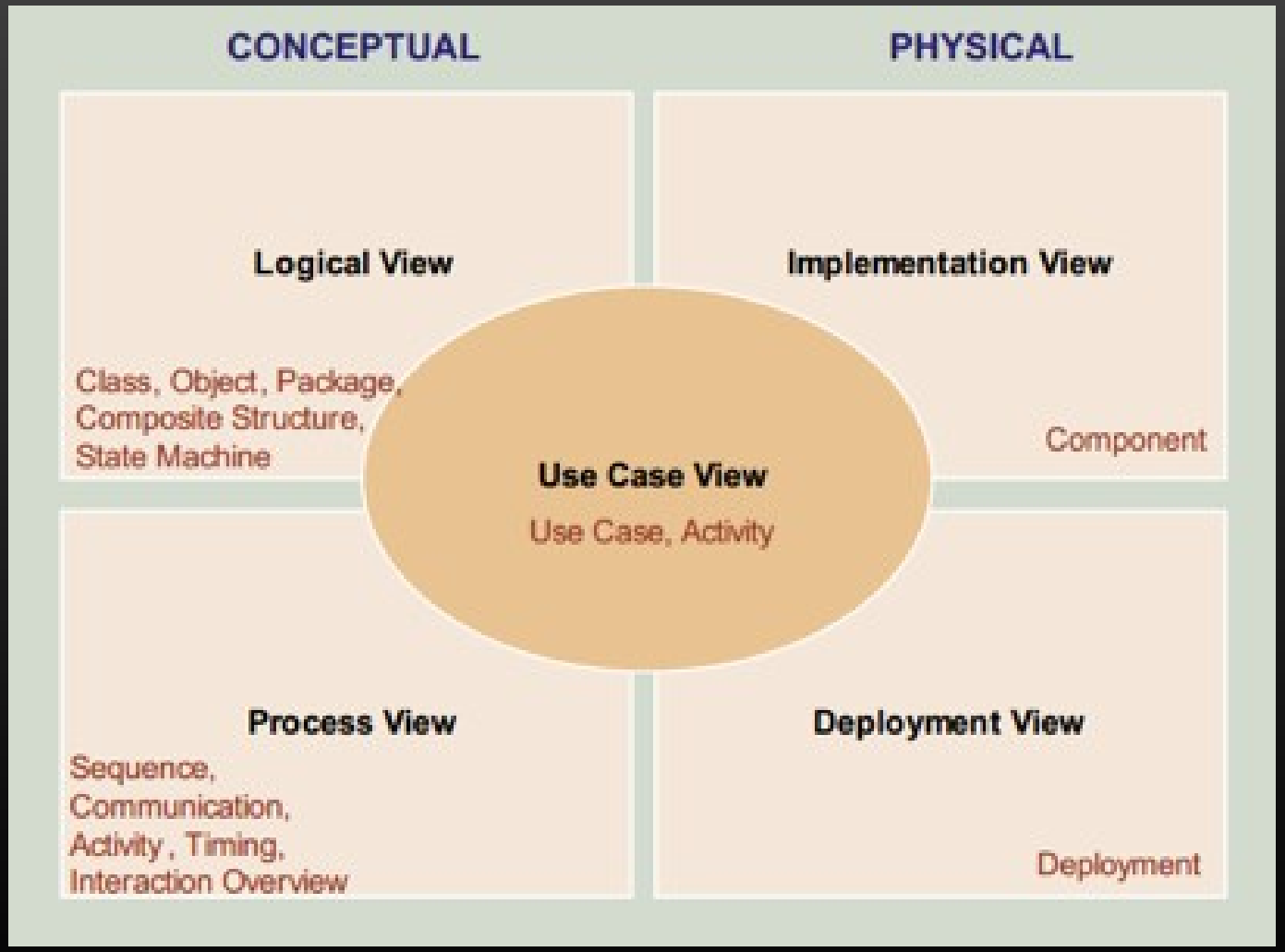
Diversos diagramas para abranger diferentes perspetivas de análise



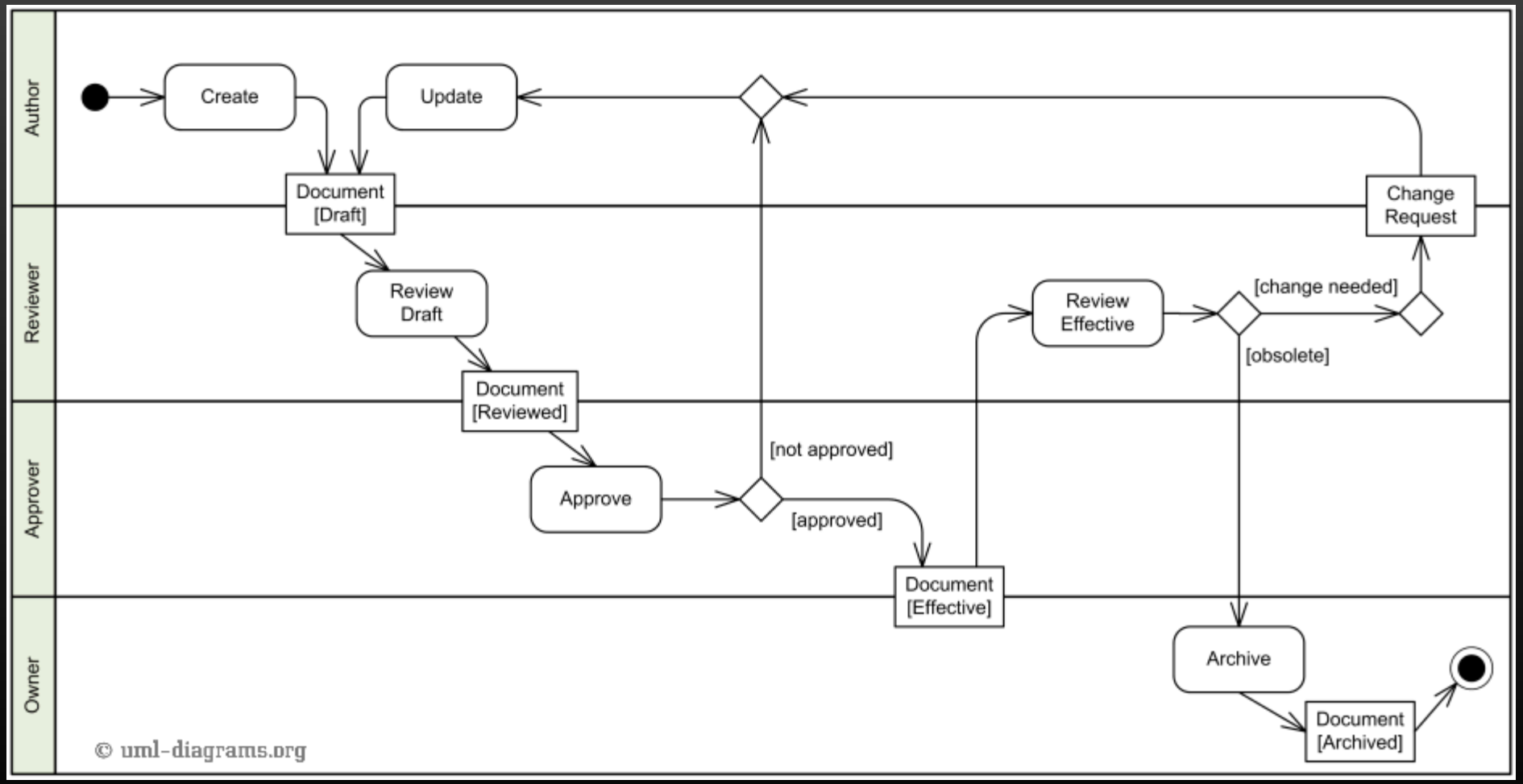
CaU do Sistema: organizar a funcionalidade do sistema em episódios de utilização



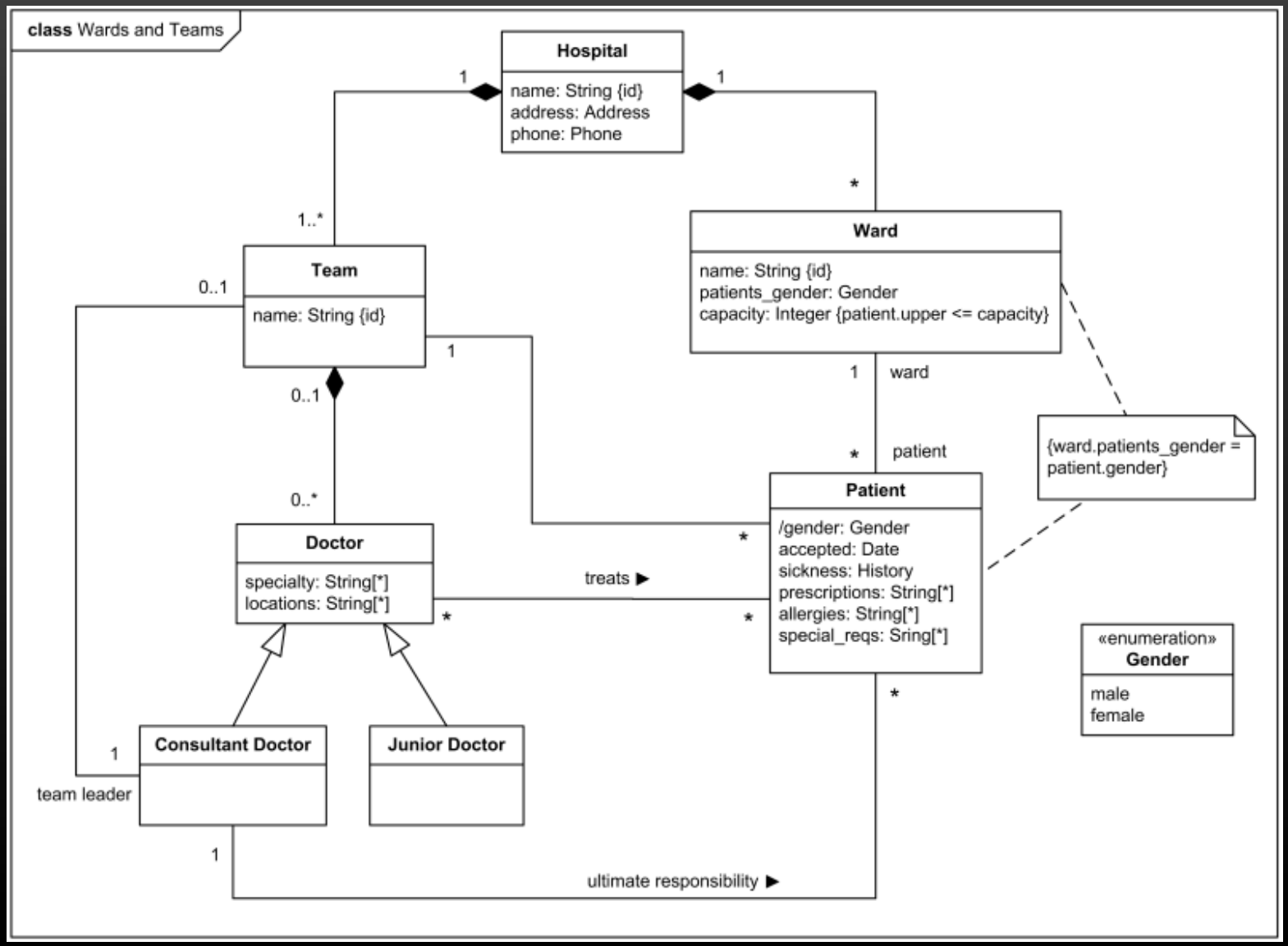
Diversos diagramas para abranger diferentes perspetivas de análise



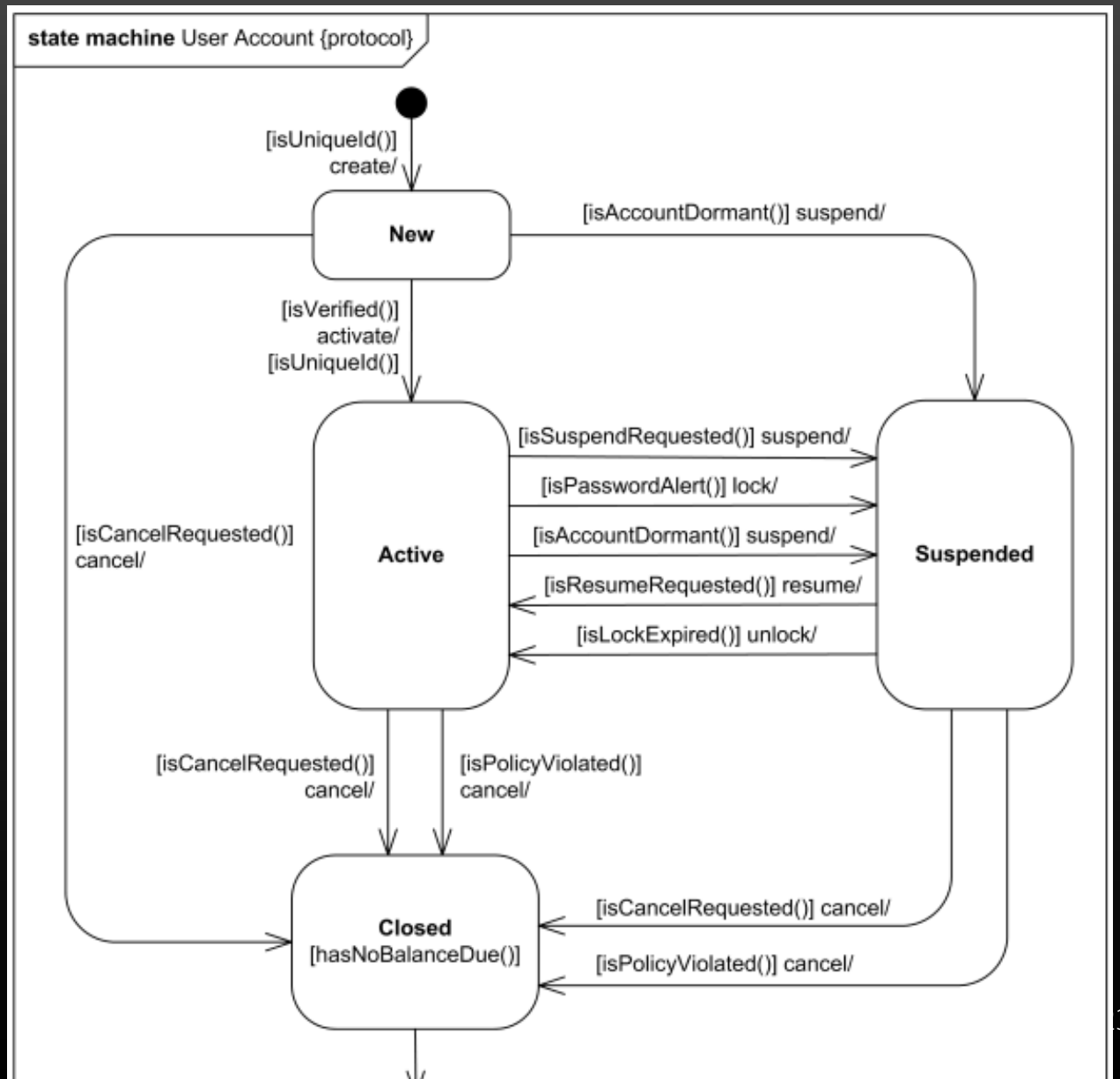
Diagramas de atividades para explicar procedimentos do domínio



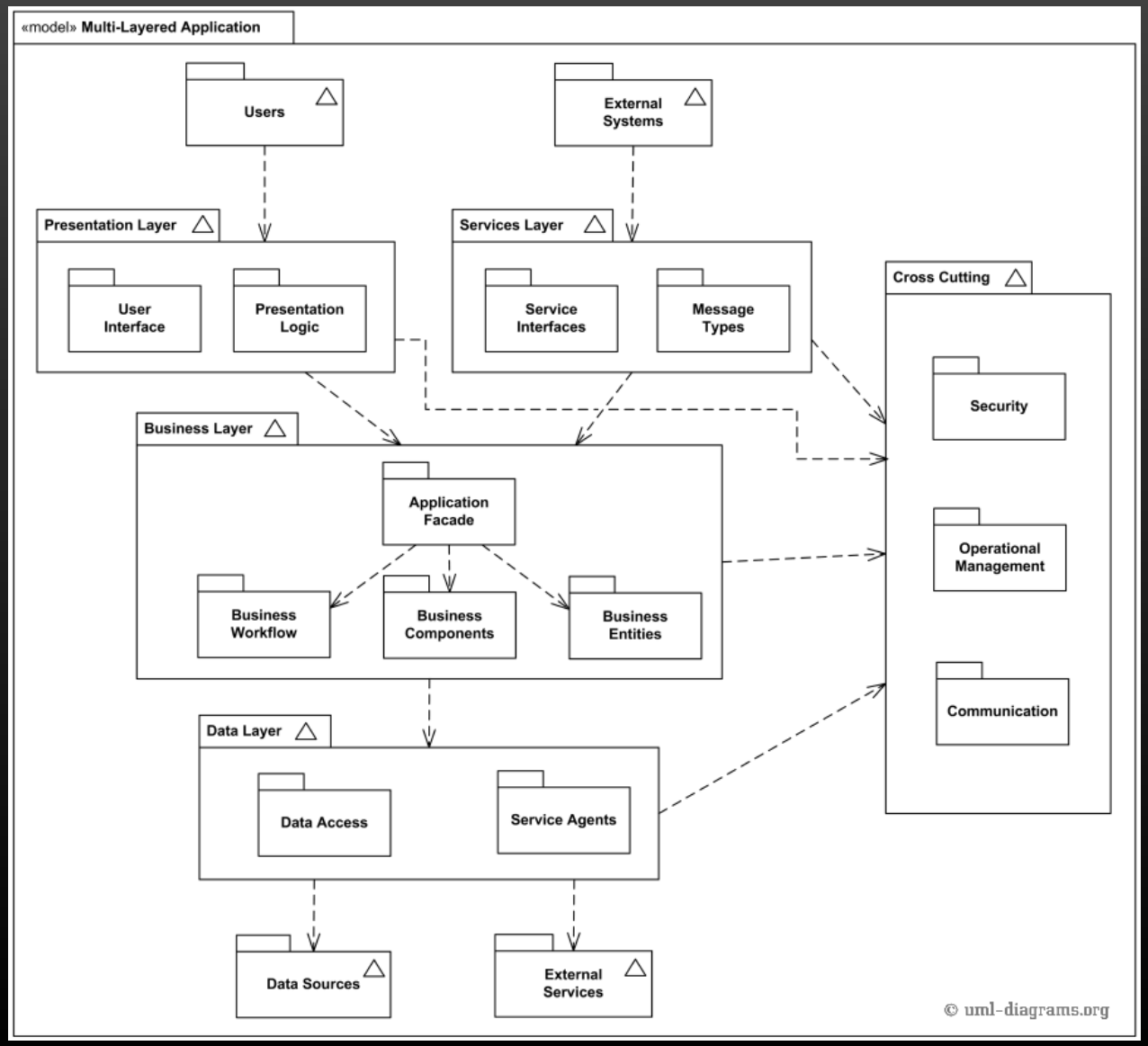
Classes para representar os conceitos da área do problema (modelo do domínio)



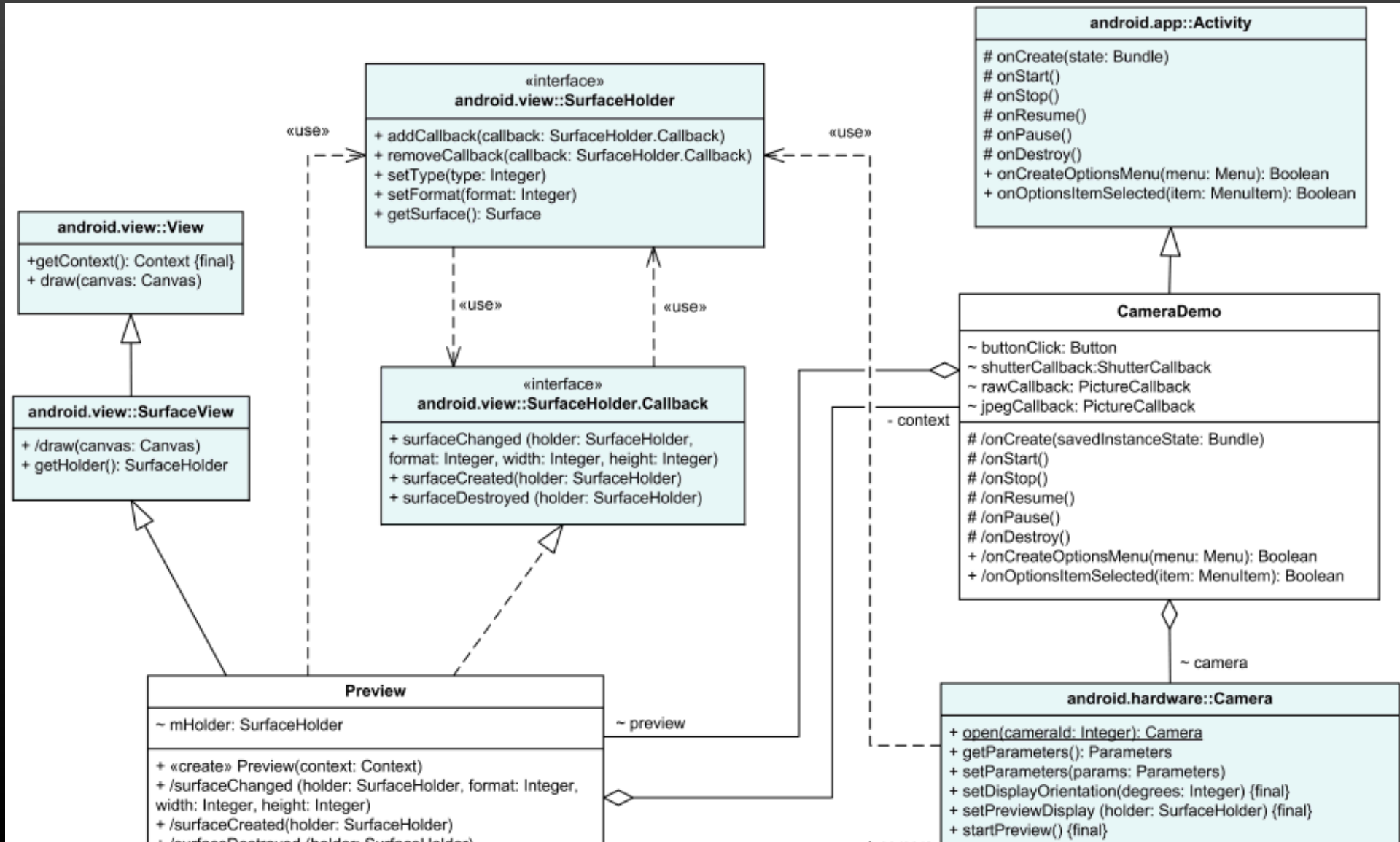
Máquina de estados de entidades/objetos



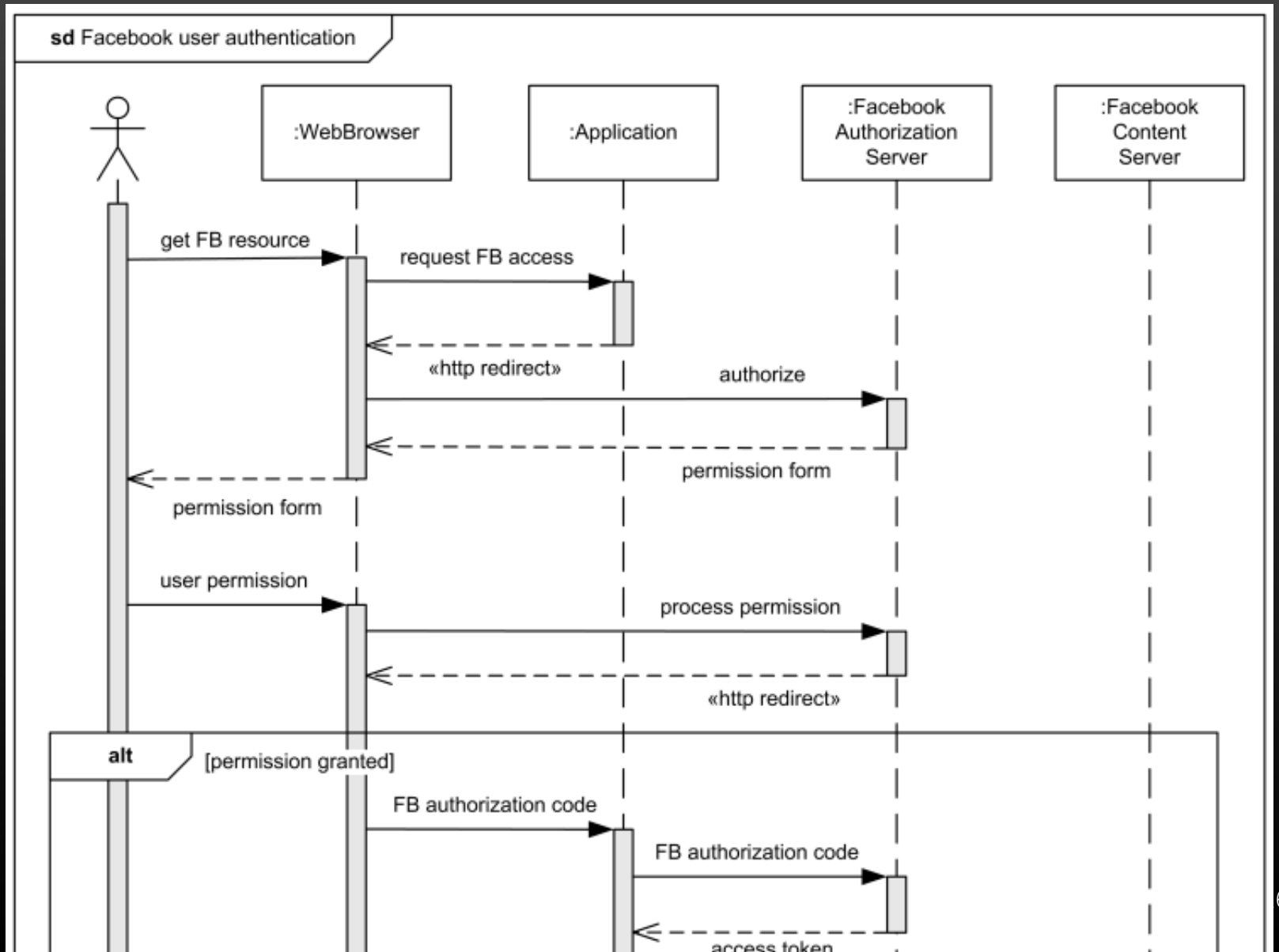
Visualizar a arquitetura lógica com D. Pacotes



Classes para visualizar objetos de um linguagem de programação



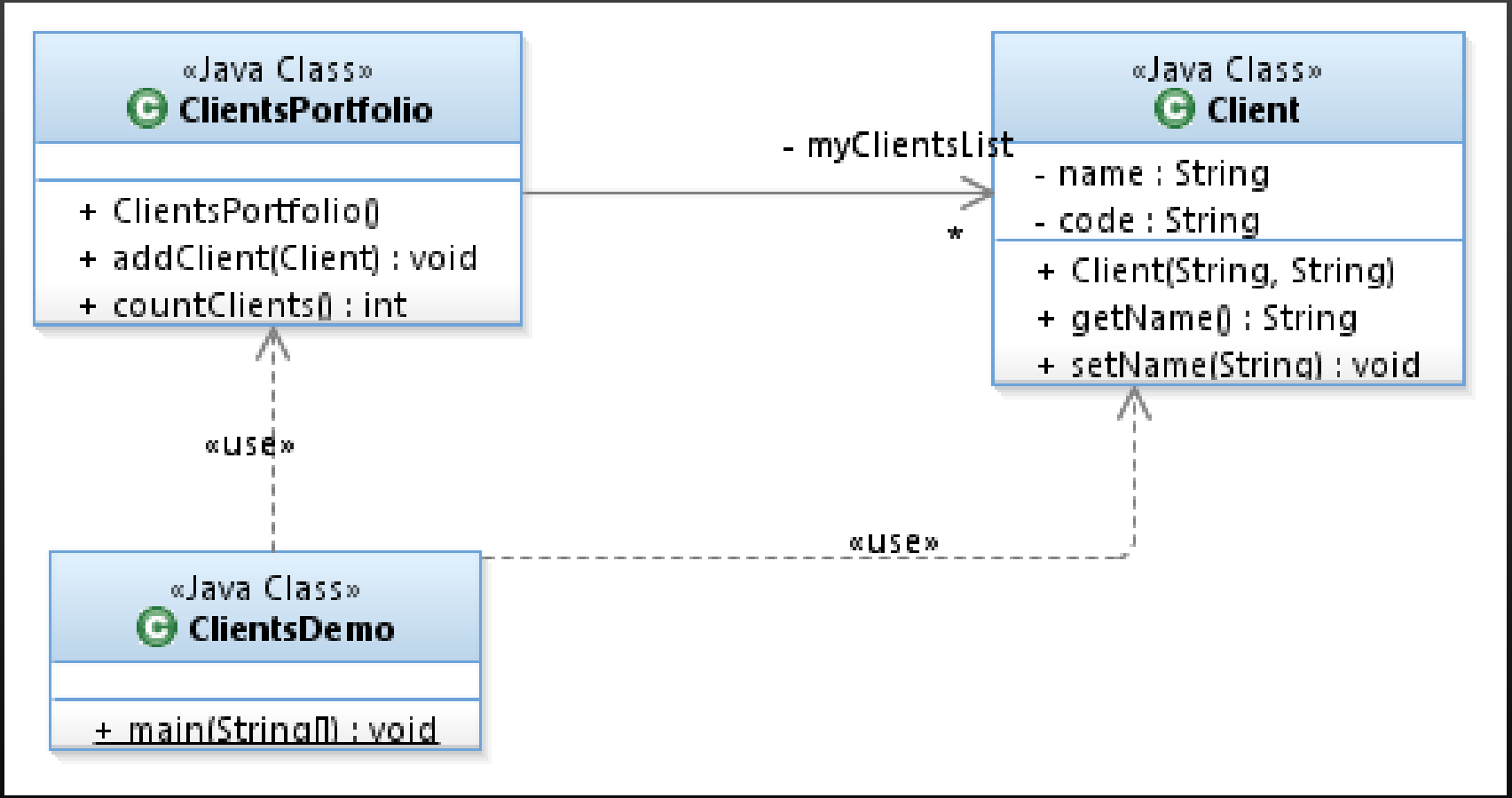
Interações entre componentes do software



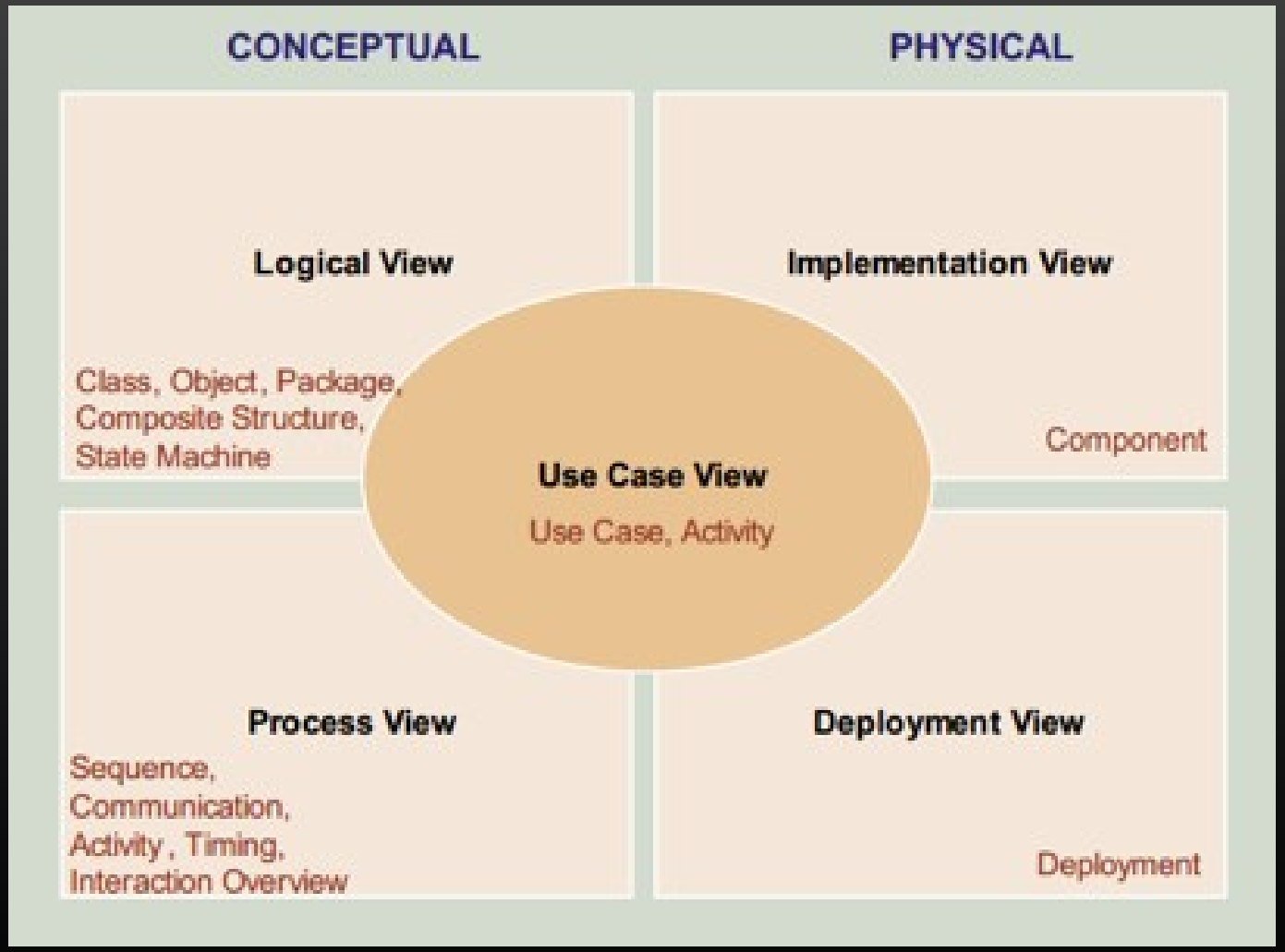
Objetos em código

```
public class ClientsPortfolio {
   private ArrayList<Client> myClientsList;

   public ClientsPortfolio() {
      myClientsList =new ArrayList<>();
   }
   public void addClient(Client newClient) {
      this.myClientsList.add(newClient);
   }
   public int countClients() {
      return this.myClientsList.size();
   }
}
```


«Java Class»
ClientsPortfolio
+ ClientsPortfolio()
+ addClient(Client) : void
+ countClients() : int
- myClientsList
*
«Java Class»
Client
- name : String
- code : String
+ Client(String, String)
+ getName() : String
+ setName(String) : void
«use»
«use»
«Java Class»
ClientsDemo
+ main(String[]) : void

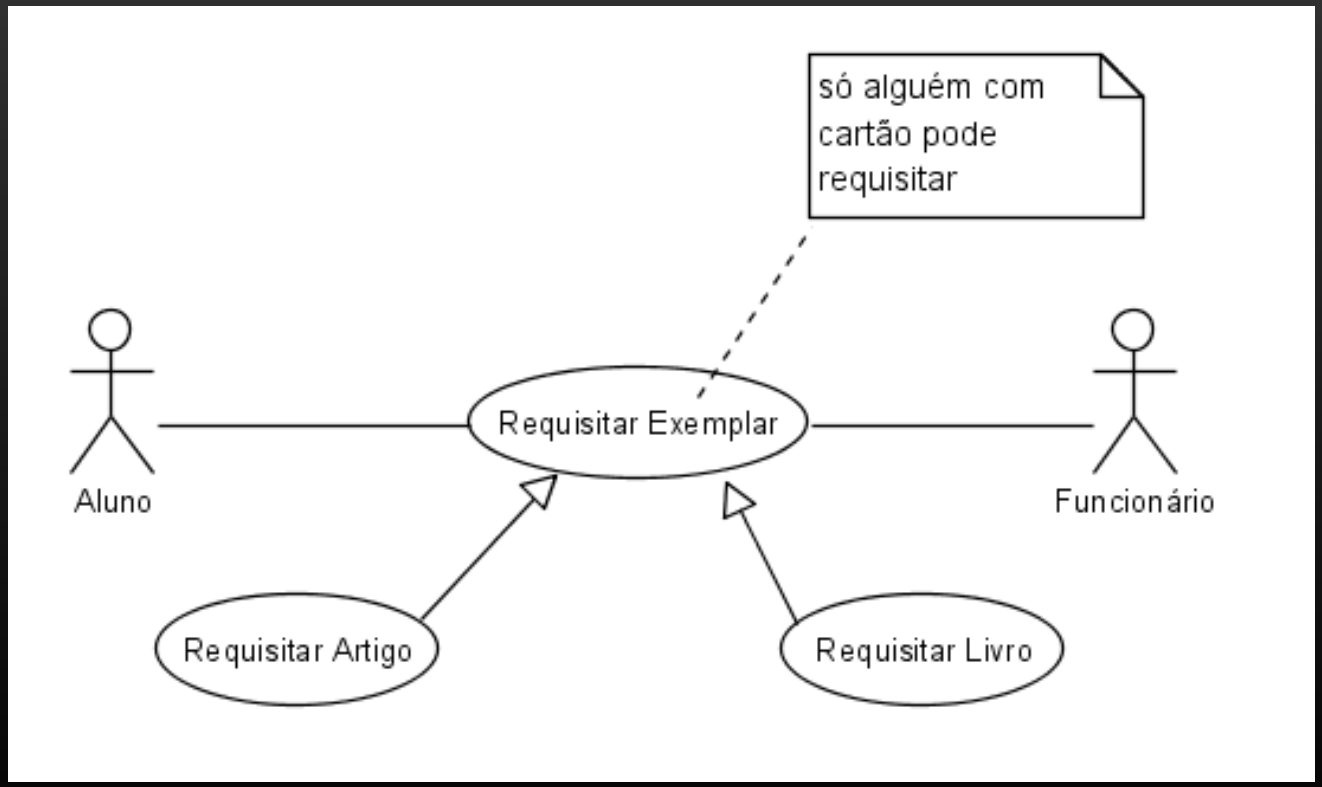
Diversos diagramas para abranger diferentes perspetivas de análise



Elementos comuns

Anotações

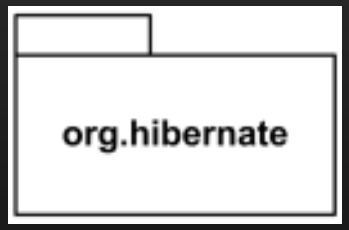
Um comentário que pode ser usado para anotar qualquer elemento

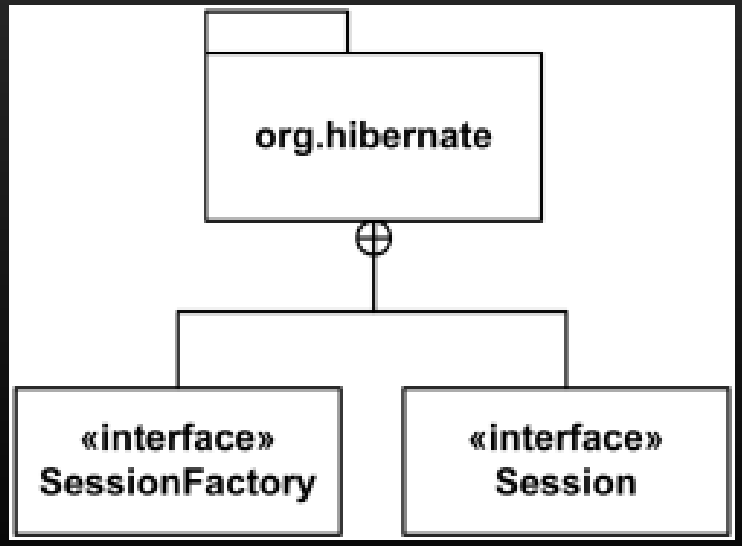


Pacotes

um mecanismo para dividir um modelo em partes

serve como mecanismo genérico para fazer agrupamentos

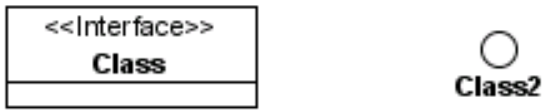


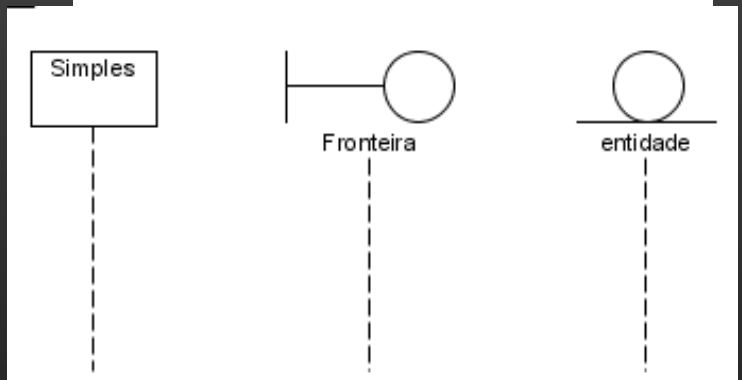


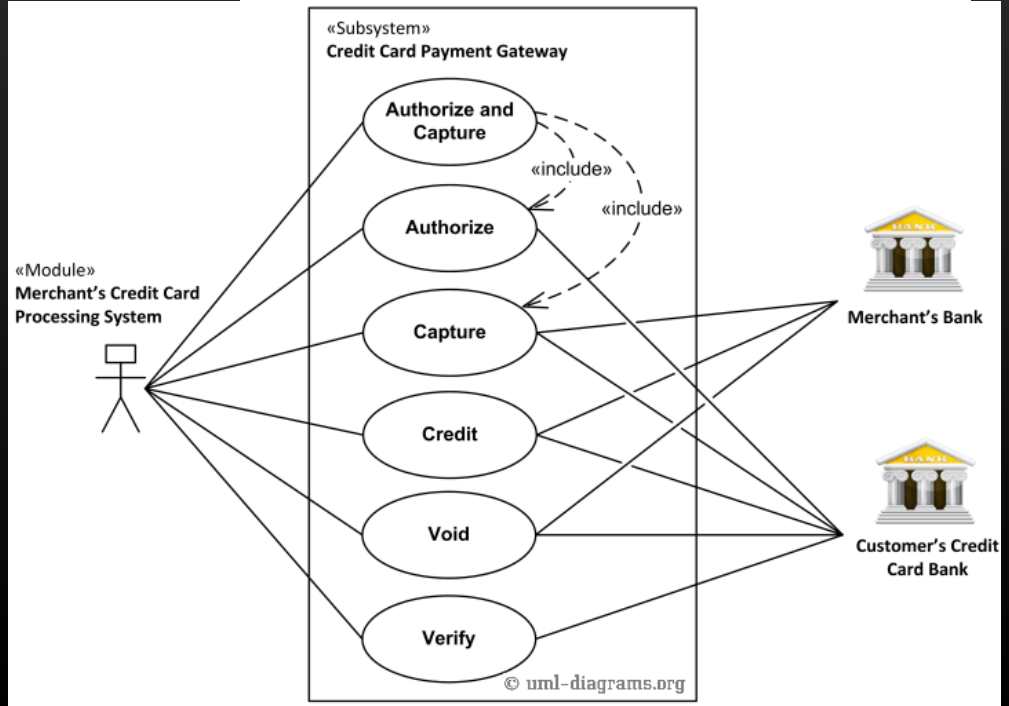
Estereotipo (*stereotype*)

uma especialização da semântica de um elemento do modelação

marcada com «...» ou com a alteração da decoração







Valores etiquetados (*tagged values*)

Estender elementos do modelo com uma linguagem "computável" (pares atributo/valor)

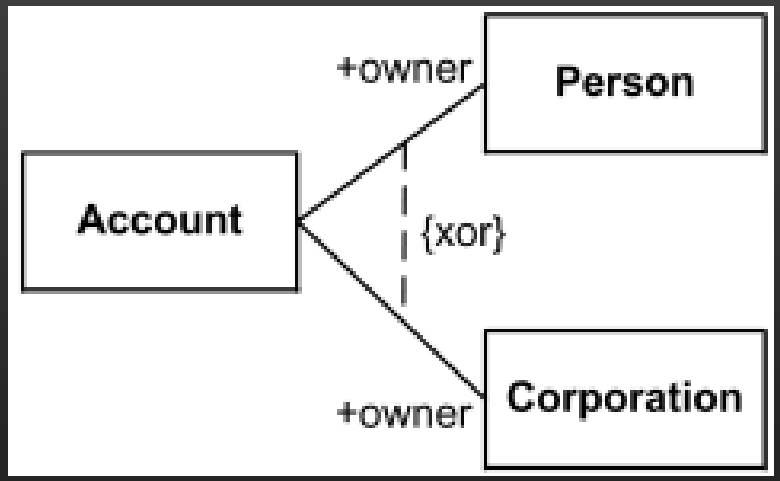
«Computer»
{Vendor = "Acer",
CPU = "AMD Phenom X4",
Memory = "4 GB DDR2"}
Aspire X1300

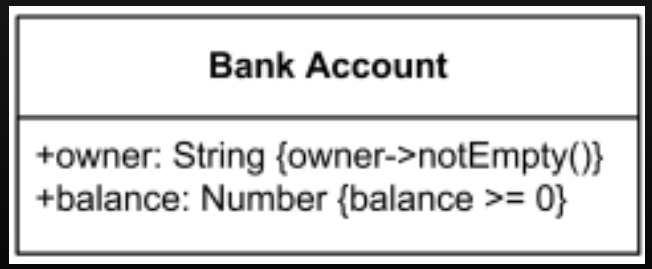
Restrições

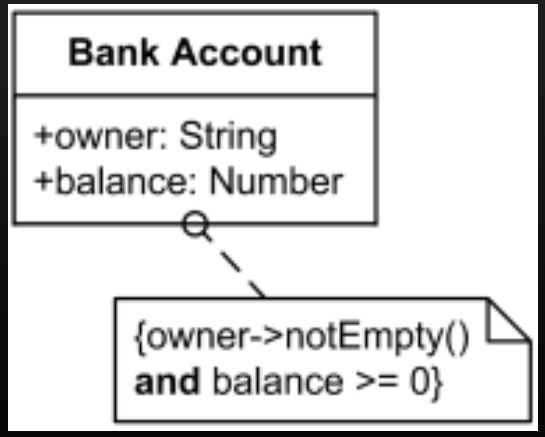
Adicionar regras ao modelo ou condicionar a sua interpretação

condição ou restrição relacionada com um ou mais elementos

Linguagem própria para declarar restrições (OCL)







Readings & references

Core readings	Suggested readings
• [Dennis15] – Chap. 1	[LAR'12] Larman, C. (2012). *Applying UML and Patterns: An Introduction to Object Oriented Analysis and Design and Interative Development*. Pearson Education. → chap. 10, chap. 15.