Use Case Modeling

FEUP-MIEIC-ESOF-2018-19

J. Pascoal Faria

Contents

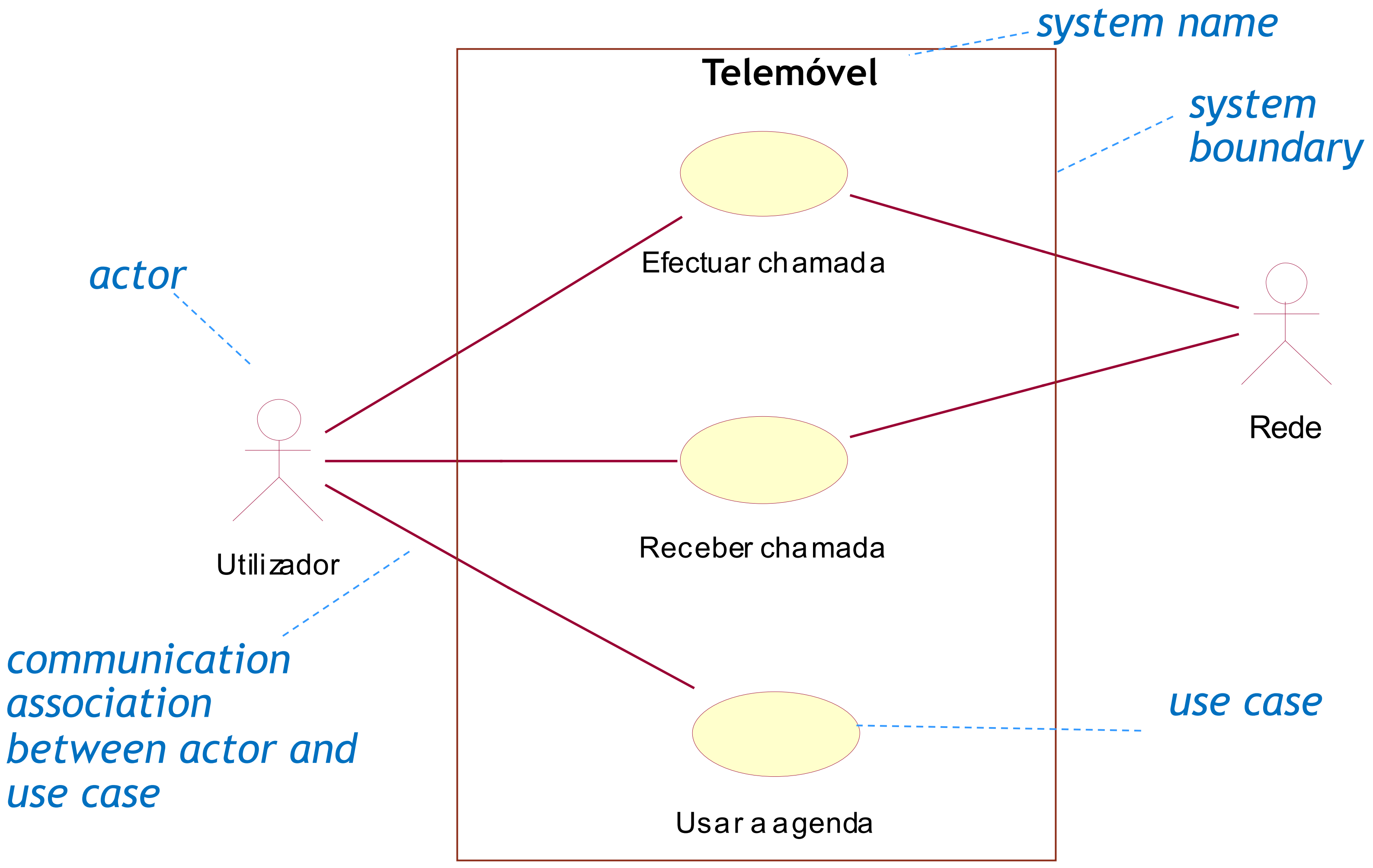
System models in requirements engineering

- A system model is a simplified representation of a system (as-is or to-be) from a certain perspective
- Models are used in many fields of engineering to tackle complexity through abstraction
- UML is the modeling standard in software engineering
- In requirements engineering, (semi-formal) models also help removing the ambiguity and lack of structure inherent to natural language descriptions
- The following are helpful in requirements engineering:
  - Use case model – for organizing functional requirements (in a way closer the structure of final system)
  - Domain model – for organizing the vocabulary and information requirements

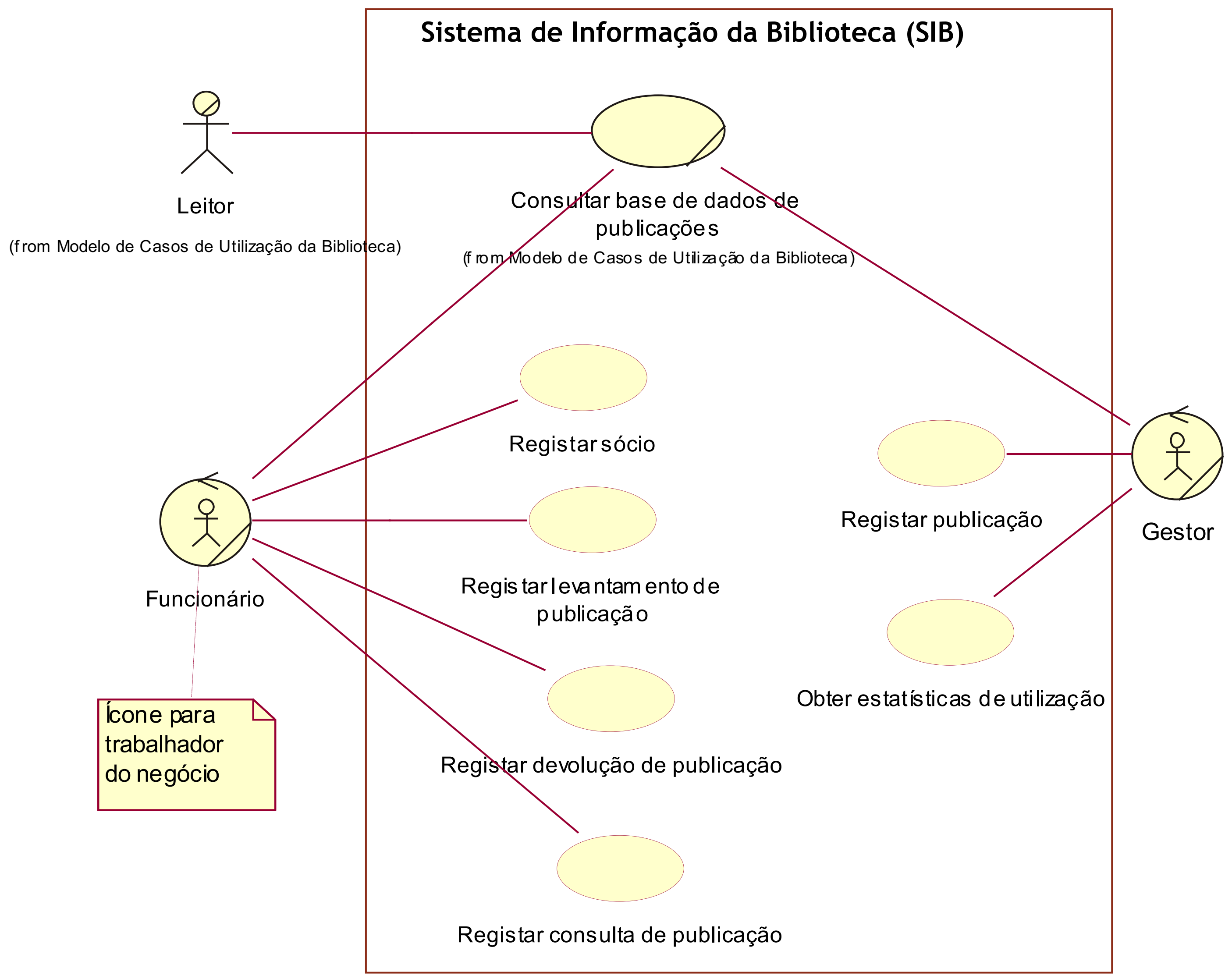
Use case diagrams

- Use case model = use case diagram(s) + associated documentation
- Show:
  - Actors: user roles or external systems
  - Use cases: system functionalities or services as perceived by users; types of interactions between actors and the system
  - Relationships between them
- Purpose:
  - show the system purpose and usefulness
  - capture functional requirements (through the use cases)
  - specify the system context (actors)
- Applicable not only to software systems
- Prepared by analysts and domain experts

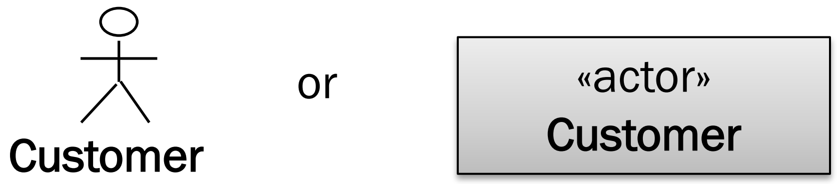
Example: equipment



Example: information system


Sistema de Informação da Biblioteca (SIB)
Leitor
(from Modelo de Casos de Utilização da Biblioteca)
Consultar base de dados de publicações
(from Modelo de Casos de Utilização da Biblioteca)
Registar sócio
Registar levantamento de publicação
Registar devolução de publicação
Registar consulta de publicação
Registar publicação
Obter estatísticas de utilização
Funcionário
Gestor
Ícone para trabalhador do negócio

Actors



- Actor = user role or external system

- Actor = role
  - An actor (in relation to a system) is a role that someone or something of the surrounding environment plays when interacting with the system

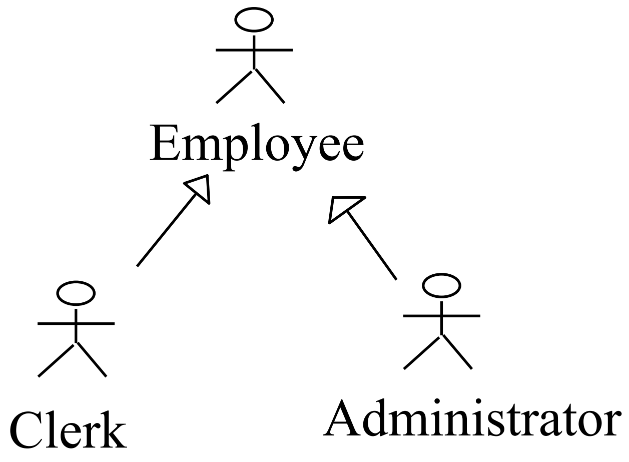
- Actor $\neq$ individual
  - the same individual can interact with the system in various roles (such as customer, supplier, etc.)

Use cases

Purchase tickets

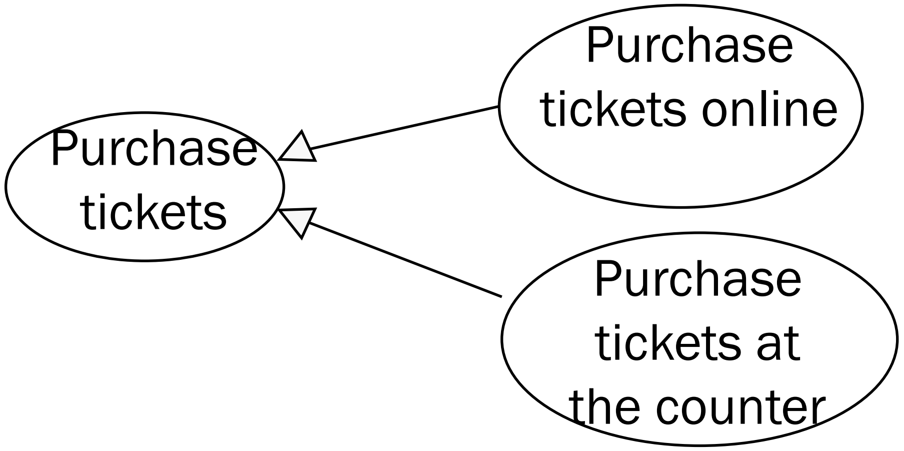
- Definitions:
  - Functionality or service as perceived by users
  - Type of interaction between actors and the system
  - Sequence of actions, including variants, resulting in an observable result with value for an actor
- Name:
  - Must show the purpose
  - Should be given from the perspective of the actor
- Granularity:
  - "Enter card" in ATM -> Not ok, has no value in isolation
  - “Withdraw money" -> Ok, has value for the cardholder
  - It includes preparatory and finalization actions: “Withdraw money: from the introduction of the card to the collection of the card, the receipt and the money"

Relationships: generalization (actors)



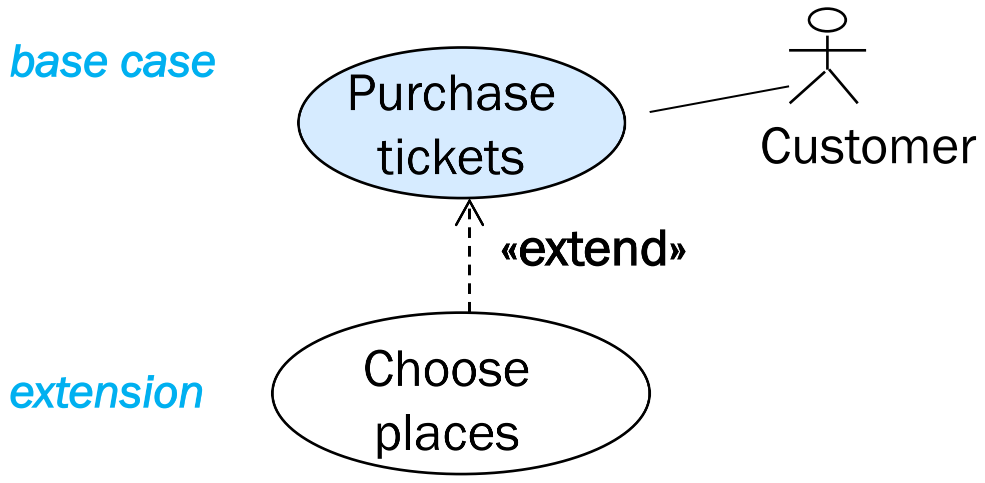
- Generalization relationship: between a more general concept and a more specialized concept
- It should be possible to read “is a (special case of)”
- Specialized actors inherit use cases of more generic actors
- Allows simplifying and structuring diagrams

Relationships: generalization (use cases)



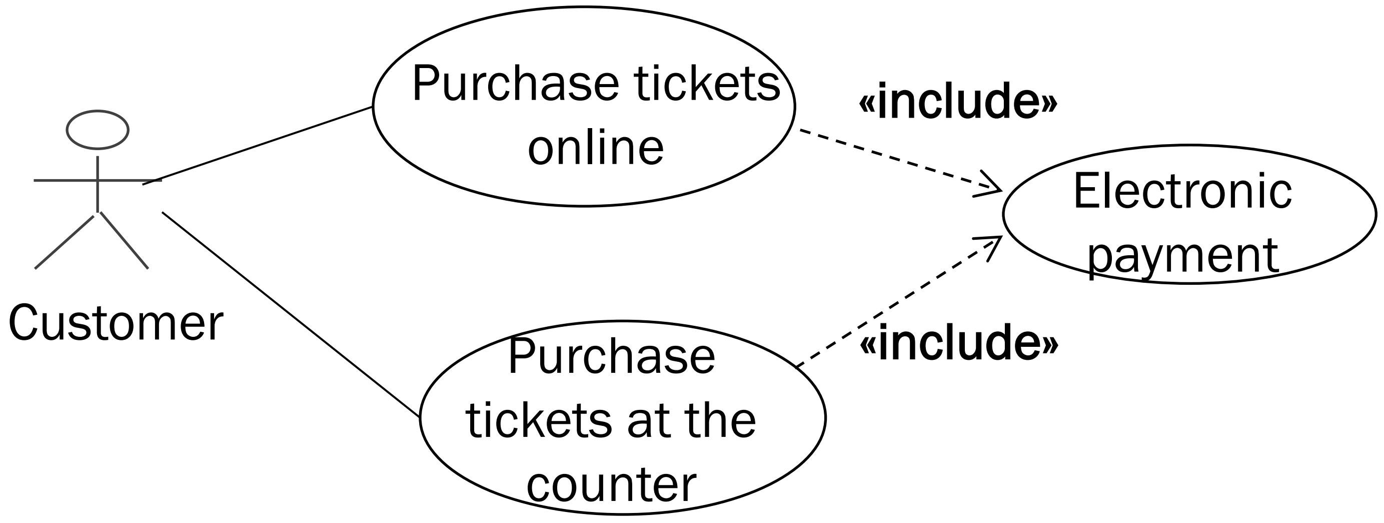
- The specialized use case inherits the behavior, meaning and actors from the generic use case, and may add behavior
- It should be possible to read "is a (special case of)"

Relationships: Extend



- Extensions to base cases indicate conditionally added behaviors
- They allow to highlight optional features and distinguish what is mandatory or essential from what is optional or exceptional
- Actors interact with the base case, which should make sense alone

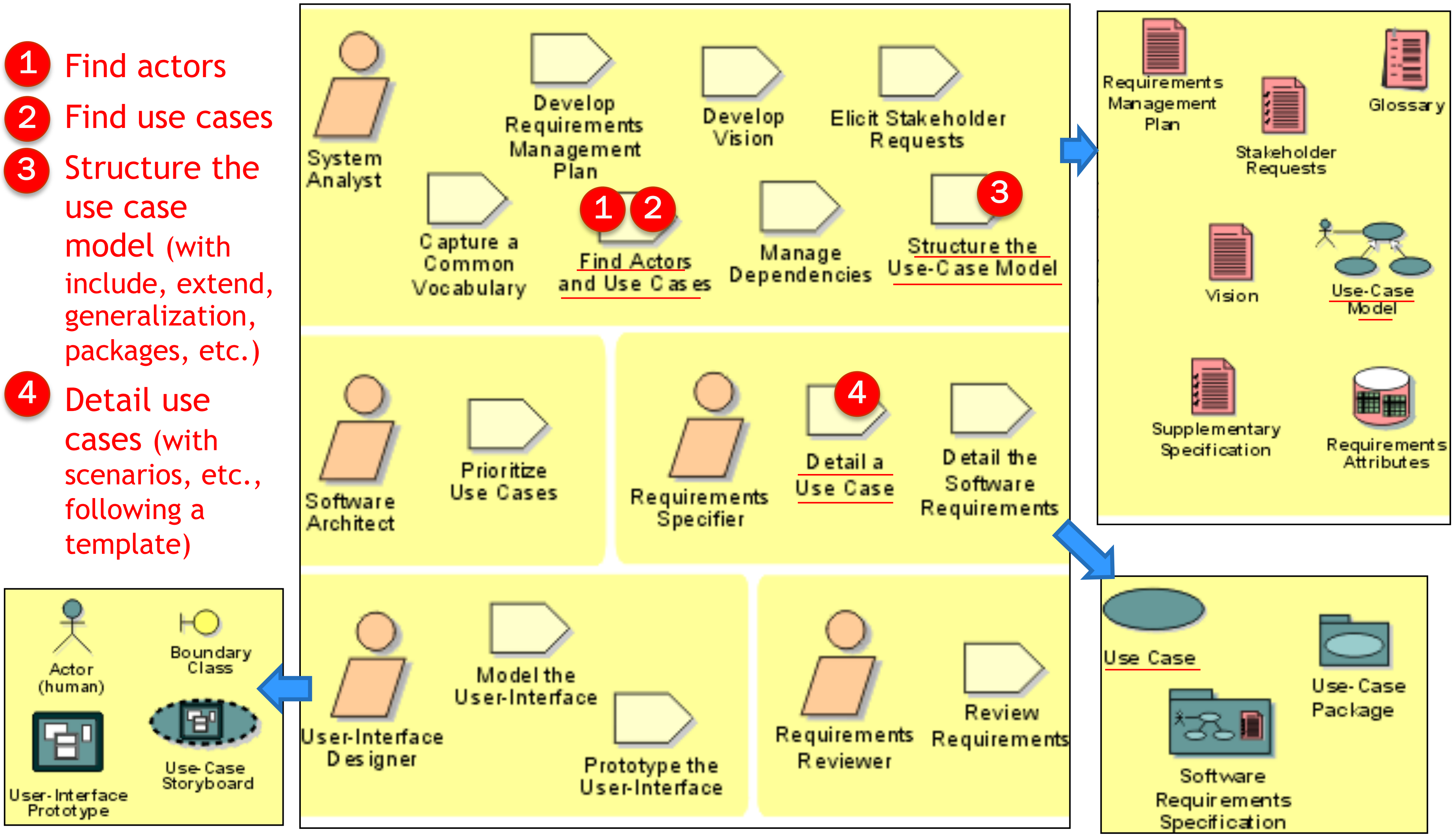
Relationships: Include

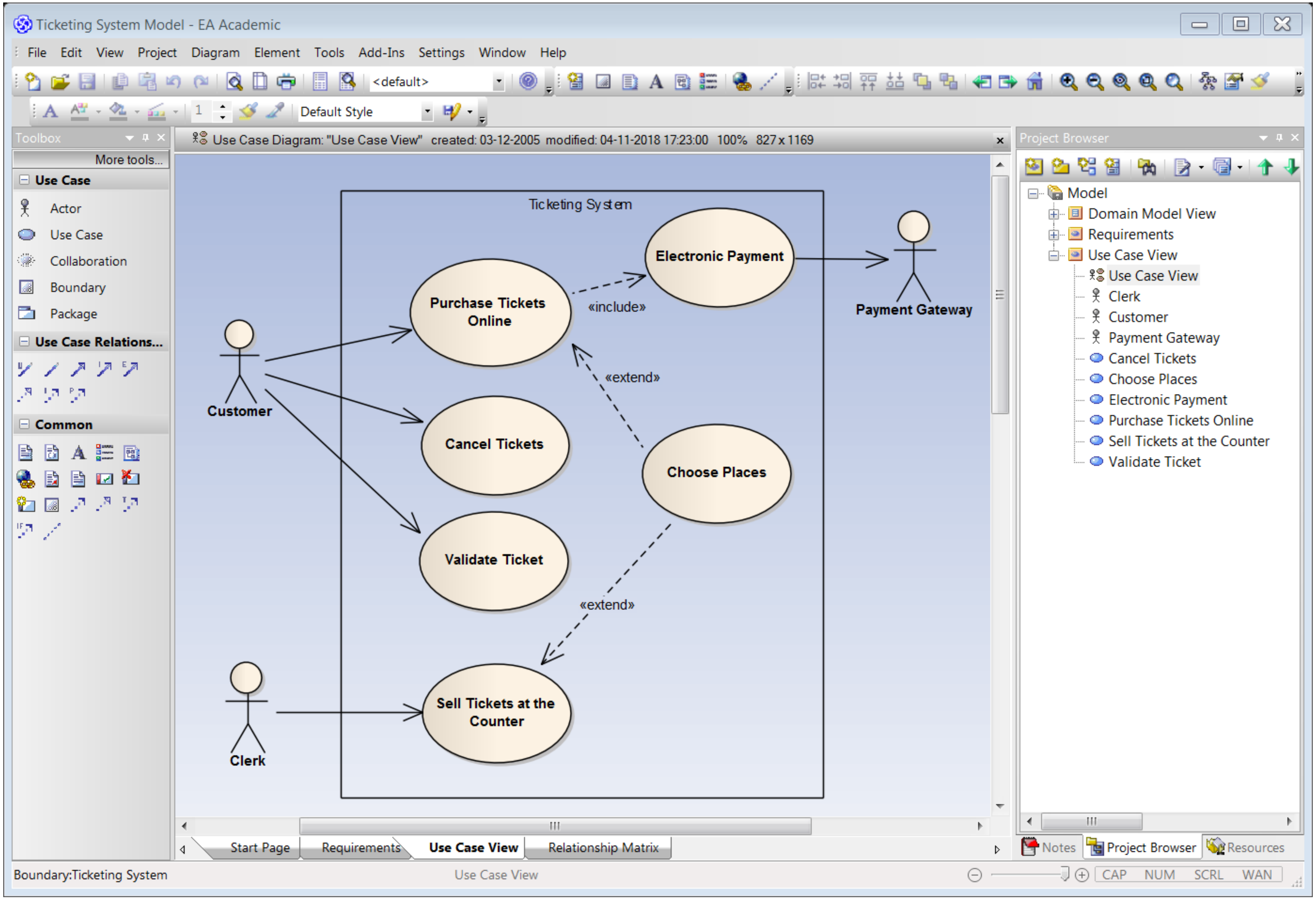


- When several use cases share some common behavior, that common behavior can be separated and described in a new use case which is included by the first ones
- Inclusion is mandatory
- In the textual description, write: include (Electronic payment)

Use-case driven requirements engineering in RUP

1. Find actors
2. Find use cases
3. Structure the use case model (with include, extend, generalization, packages, etc.)
4. Detail use cases (with scenarios, etc., following a template)



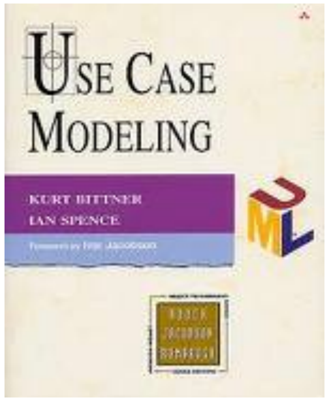


Use Case Diagram: "Use Case View" created: 03-12-2005 modified: 04-11-2018 17:23:00 100% 827 x 1169

Boundary:Ticketing System

References and further info

- Use Case Modeling,
  K. Bittner, I. Spencer,
  Addison-Wesley, 2003



- www.uml.org – especificações e recursos sobre UML
- Software engineering, 9th edition, Ian Sommerville, Chapter 5 – System Modeling
- http://www.sparxsystems.com.au/resources/uml2_tutorial/
- http://www.agilemodeling.com