UNIFIED
MODELING
LANGUAGE

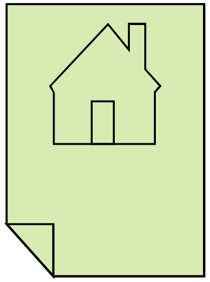
UML language - 1

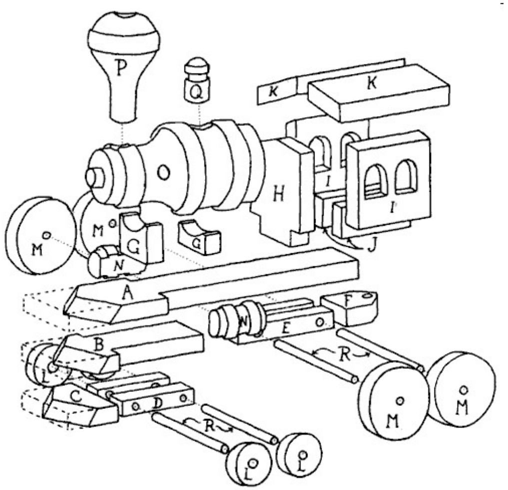
Lydie du Bousquet

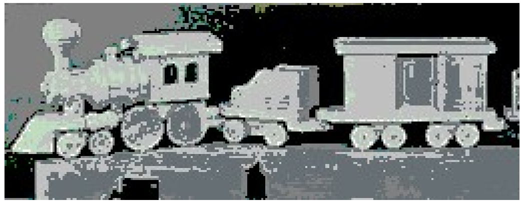
Lydie.du-bousquet@univ-Grenoble-alpes.fr

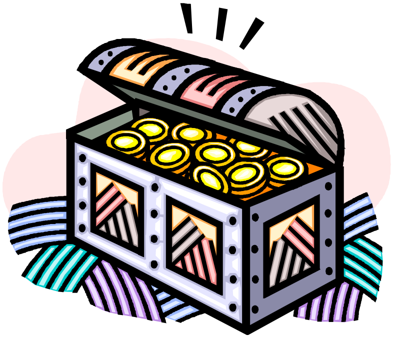
En collaboration avec J.-M. Favre, I. Parissis, Ph. Lalanda

Need of representations to discuss, organize, build, document…









In SE, models are used

- As a starting point to
  - abstract and to understand
  - support the discussion
  - organize, plan
- To design and detail
- As support at the end of development
  - To test
  - To document
  - To maintain

UML = Unified Modeling Language

- A **language**
- For **modeling**
  - at the analysis and the design stages (Object-oriented)
- **Unified**
  - To cover as many **domains** as possible
  - To cover as many **notions** as possible

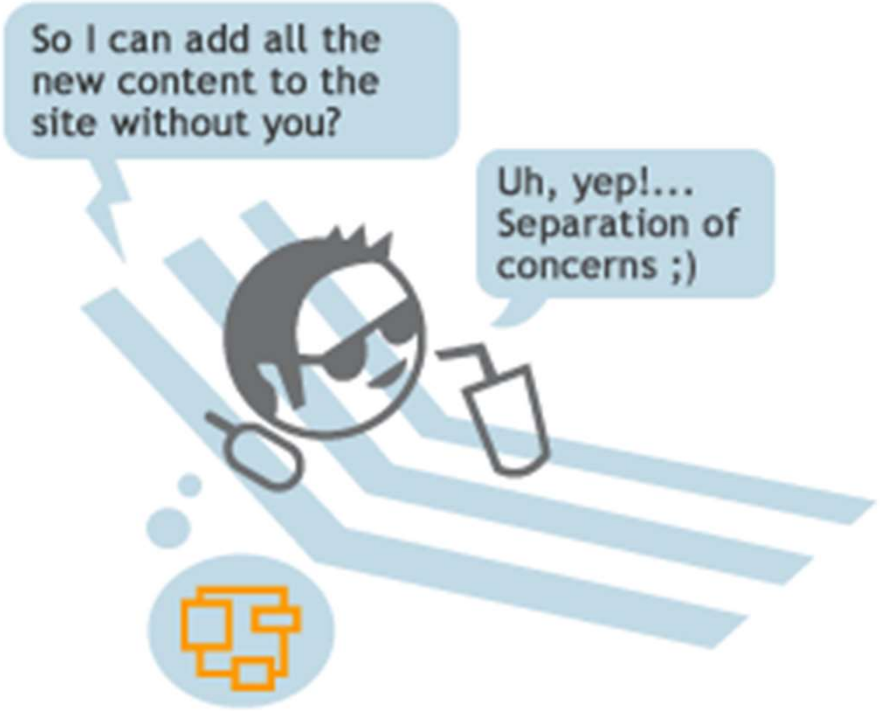
- Objective: different analysts can
  - Have a common language to discuss
  - Common tools

UML = standard

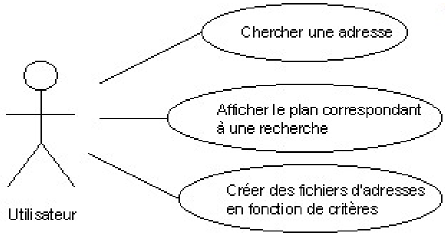
- International standard
  - Very large (many notions)
- More and more used in the industry
- Associated to several
  - methods
  - Tools
- Can be used with different level of details

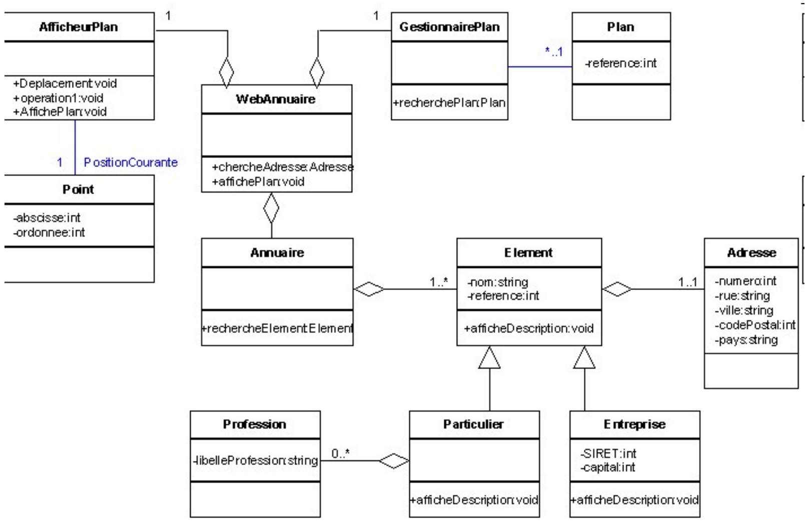
UML: a language, several views

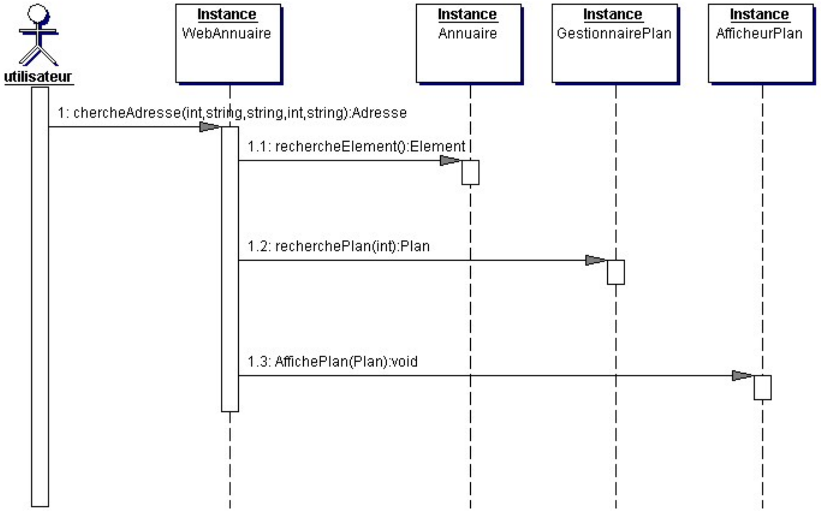
- Different needs
  - To model static or dynamical point of views
  - At different stage analysis, specification, design, …
- Using views
  - Separation of concerns


So I can add all the new content to the site without you?
Uh, yep!... Separation of concerns ;)

UML: a language, several views


Chercher une adresse
Afficher le plan correspondant à une recherche
Créer des fichiers d'adresses en fonction de critères
Utilisateur




utilisateur
Instance
WebAnnuaire
Instance
Annuaire
Instance
GestionnairePlan
Instance
AfficheurPlan
1: chercheAdresse(int,string,string,int,string):Adresse
1.1: rechercheElement():Element
1.2: recherchePlan(int):Plan
1.3: AffichePlan(Plan):void

14 diagrams in UML 2.2

Structure diagrams

- **Class diagram**
- **Object diagram**
- Component diagram
- Composite structure diagram
- **Deployment diagram**
- Package diagram
- Profile diagram

Behaviour diagrams

- **Use case diagram**
- **State Machine diagram**
- Activity diagram

Interaction diagrams

- **Sequence diagram**
- Communication diagram
- Interaction overview diagram
- Timing diagram

UML 2.2

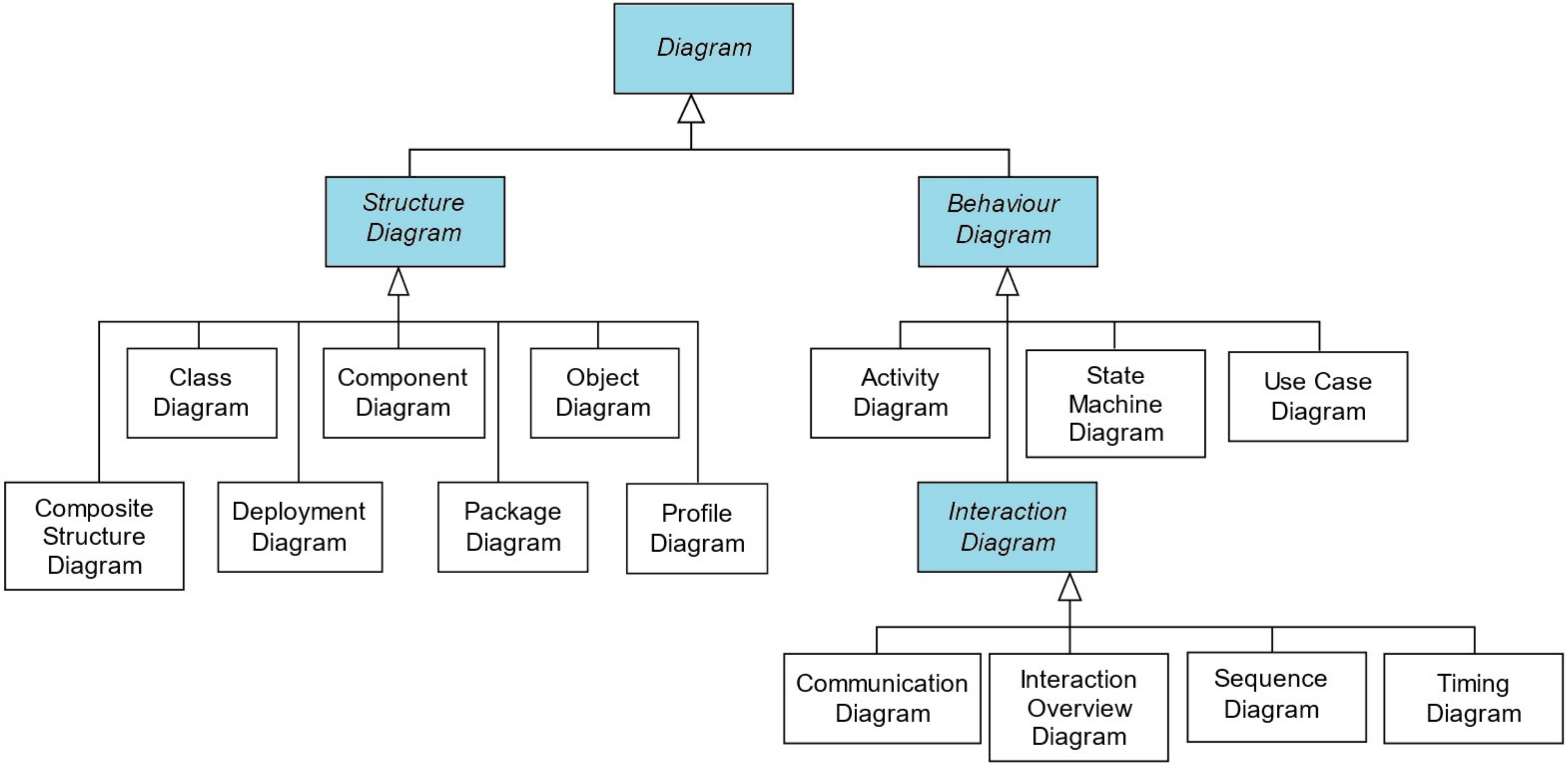
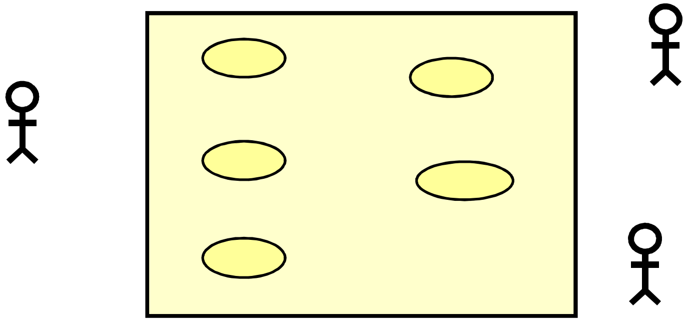


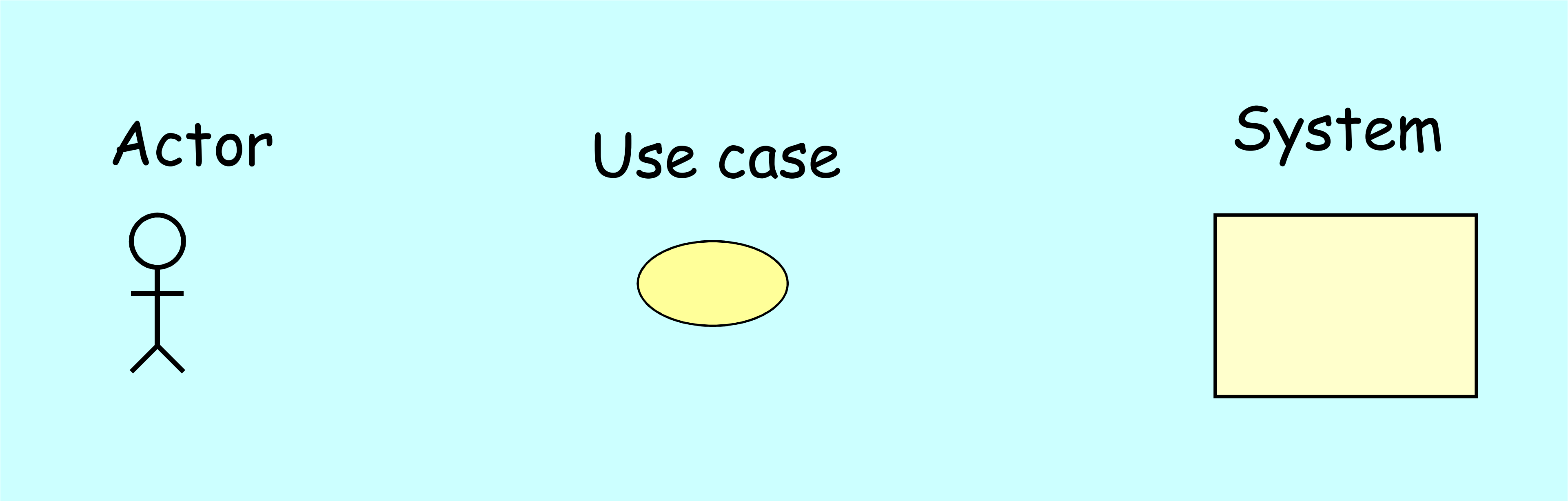
Diagram
Structure Diagram
Behaviour Diagram
Class Diagram
Component Diagram
Object Diagram
Composite Structure Diagram
Deployment Diagram
Package Diagram
Profile Diagram
Activity Diagram
State Machine Diagram
Use Case Diagram
Interaction Diagram
Communication Diagram
Interaction Overview Diagram
Sequence Diagram
Timing Diagram

Use-case diagram

- Documents the system's intended behavior
- Representation of the **relationships** between **actors** and **use-cases**
- Arrows and lines are drawn
  - between actors and use cases (by default «communicates»)
  - between use cases to show their relationships
  - Between actors to show their relationships

Use-case diagram





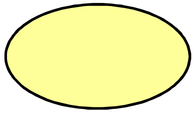
Actor

Use case

System

Use cases

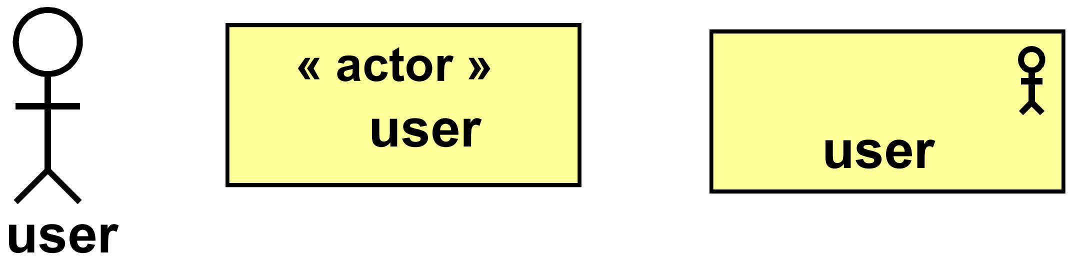
- They are drawn as an **ellipse**



- Names are normally made up of
    - an active verb and a noun
    - noun phrase
- Names should be **representative** of the behavior

Actors

- People or systems that interact with use cases
- They represents **roles**
  (not John or Mary, but assistant or supervisor)



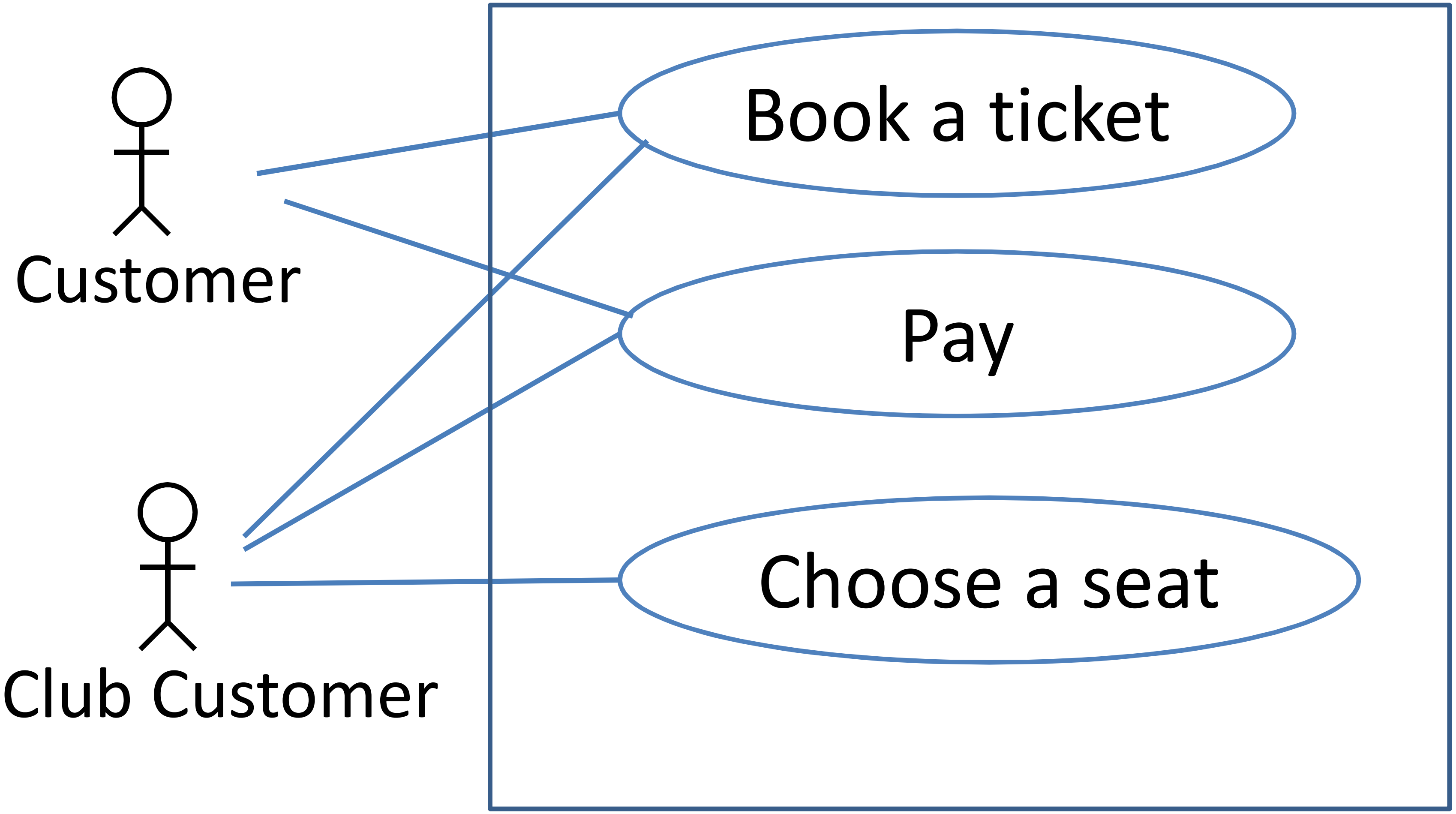
« actor » :
stereotype

- They are **connected** with use cases with which they interact
- The **name** of the actors should be chosen carefully
- **Generalization** can be used between 2 actors

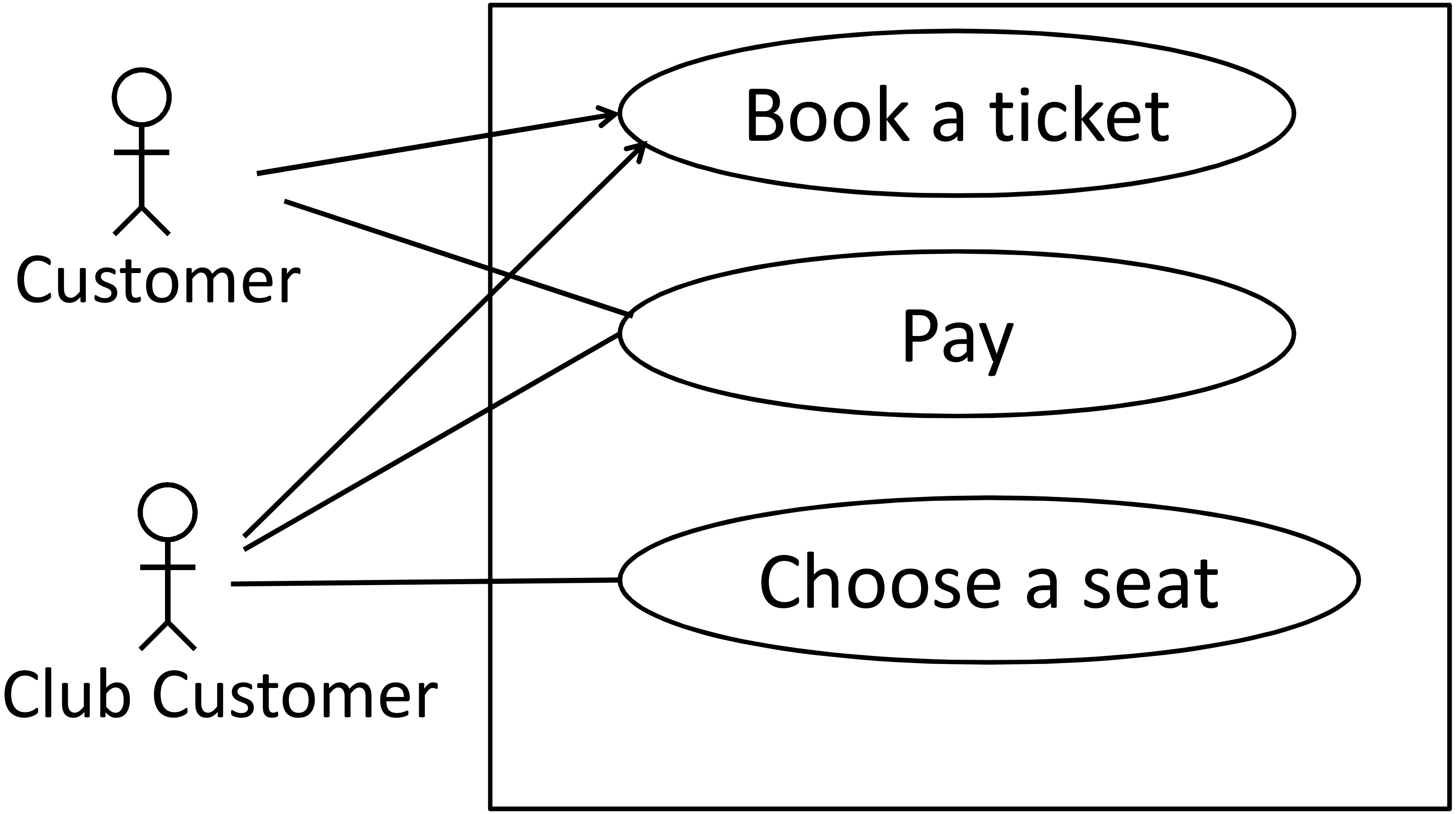
Relationships

- Association between actor and use-case
- Generalization between two actors
- Association between two use-cases
  - Include
  - Extend
- Generalization between two use-cases

Association between actor and use-case
Example: Air flight company system

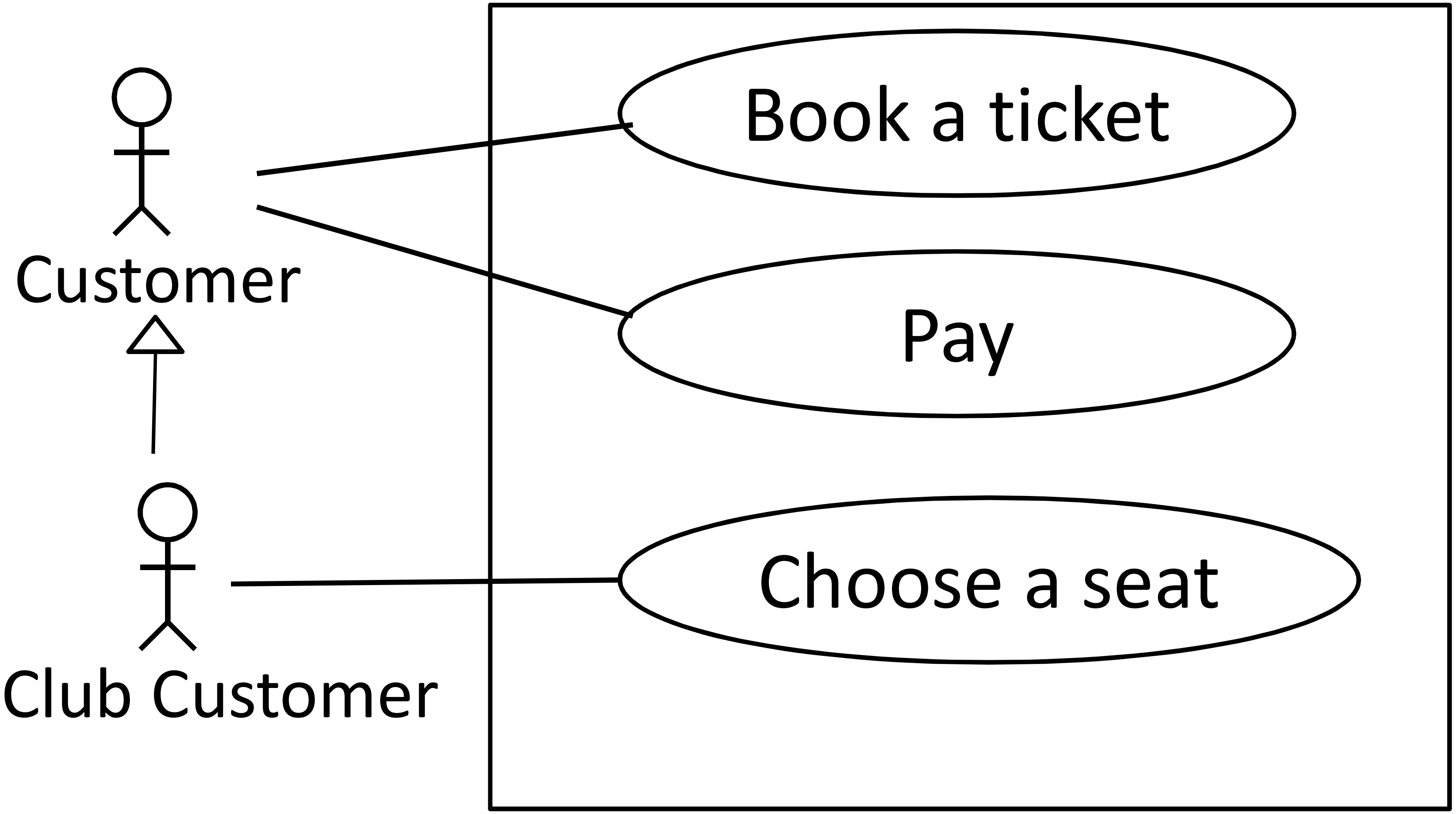


Association between actor and use-case
Example: Air flight company system



An arrow can be used to specify the direction of the initial invokation

Air flight company system



A club customer is a special customer: generalization relation.

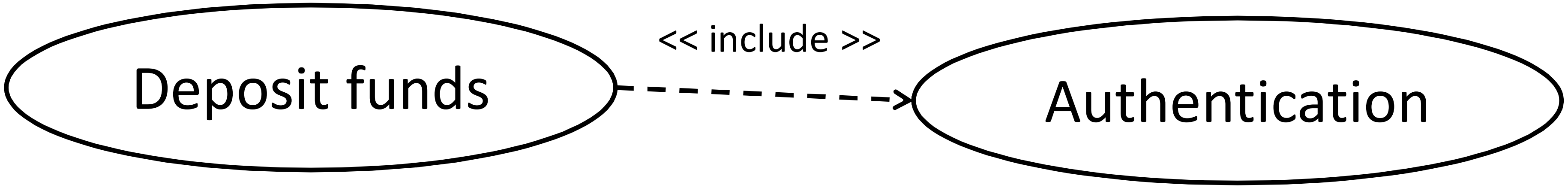
Behavior specification

- Use case **diagram** vs use case **model**
- A use case represents a **sequence of activities** that results in some observable outcome
- The activities have to be **documented**
  - Simple paragraph
  - Two-column presentation (actor and system)
  - UML sequence diagram, communication diagram, …

Include and extend relationships

<< include >>

- encapsulates a functionality used at several point in the systems
- to avoid repetition

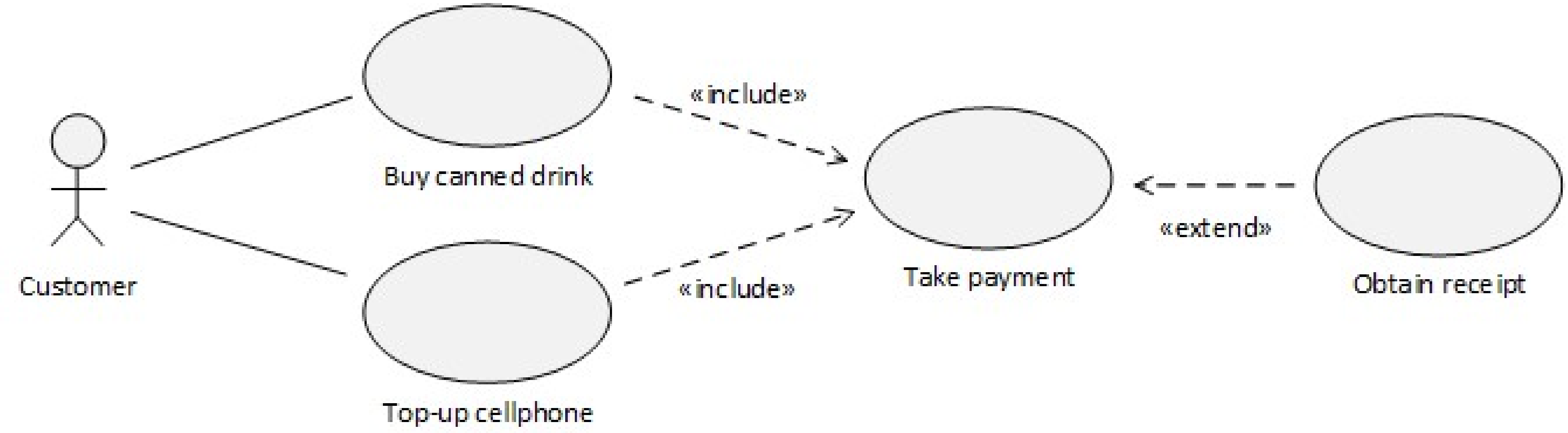


To deposit funds, it is necessary to achieve authentication

Include and **extend** relationships

<< extend >>

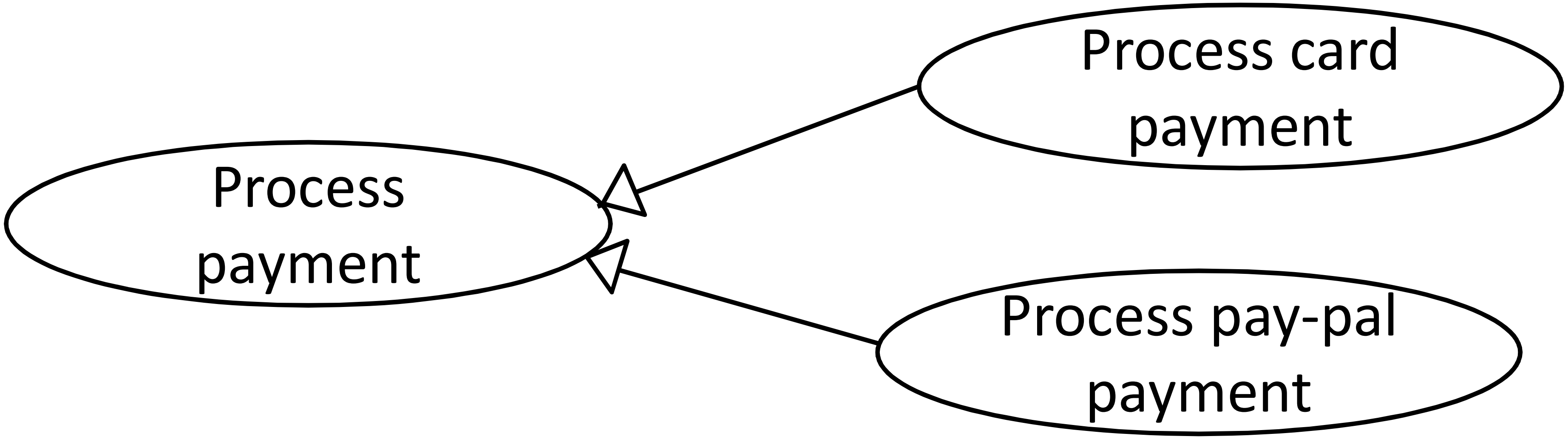
- denotes the fact that the use case may be optionally extended



http://www.ropley.com/use-case-extend-relationships.aspx

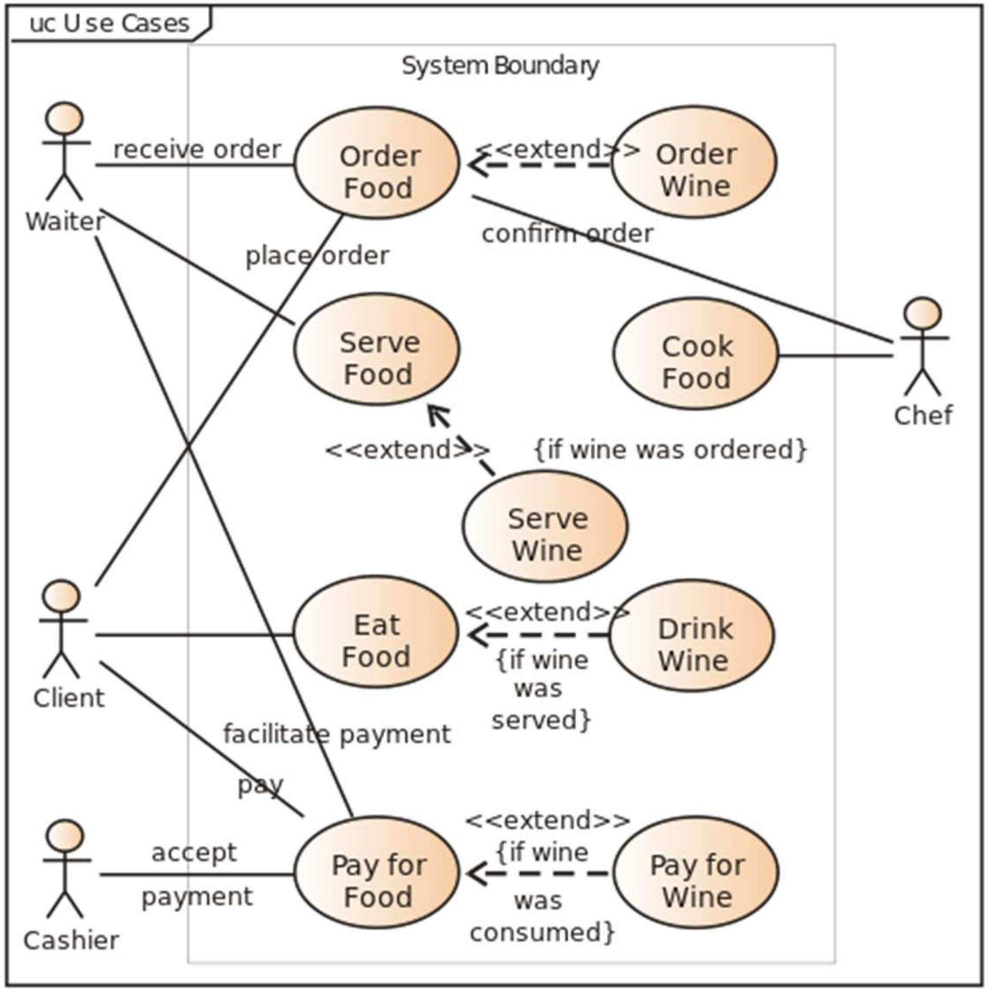
Generalization between two use cases

- A parent use case may be specialized into one or more child use cases that represent more specific forms of the parent.



Express in UML the list of Cyber-video functions as a use-case diagram

Use-case diagram example



use case diagram for the interaction of a client (actor) within a restaurant (system)
https://en.wikipedia.org/wiki/Use_Case_Diagram

Unified Process

(1) Define the use case model

   (1) Introduce the system (small description)

   (2) Find the actors

   (3) Give a small description of each actor

   (4) Find use-cases, express the relations

   (5) Describe the system as a whole

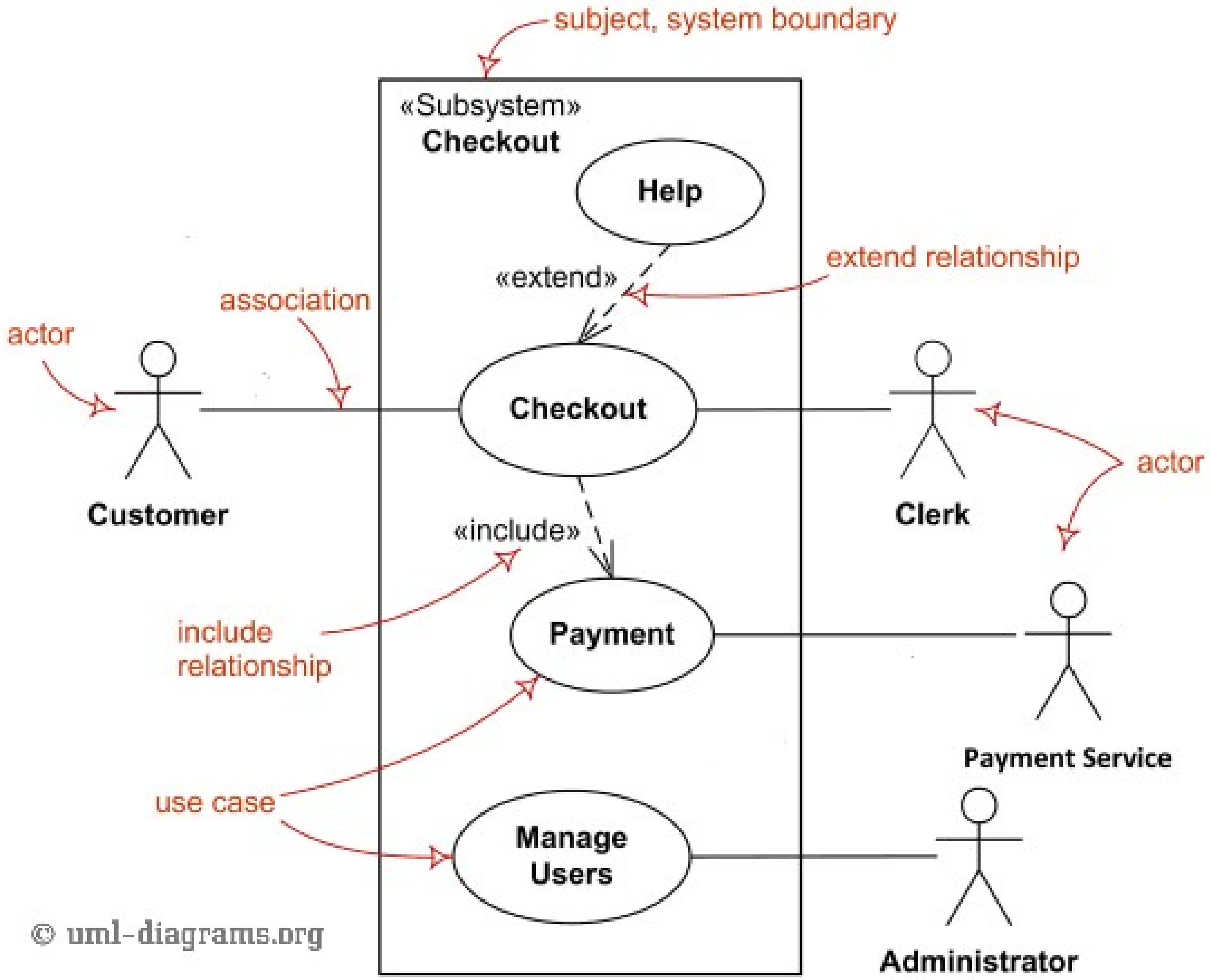
(2) Define the priority among the use cases

(3) Detail the use cases (w.r.t. the priorities)

Advices

- Use **strong** verb and **domain** vocabulary for use cases:
  - – *withdraw is stronger than perform withdrow*
  - – *Convey pakage vs deliver shipment*
- Name actor with domain-relevant name and place primary actor on the top left
- Use an actor « time » to initiate scheduled event (can be represented as ⌛)

Use Case


subject, system boundary
«Subsystem»
Checkout
Help
extend relationship
«extend»
association
actor
Checkout
Customer
Clerk
actor
«include»
include relationship
Payment
Payment Service
use case
Manage Users
Administrator
© uml-diagrams.org

Exercises

3.1

3.2

14 diagrams in UML 2.2

Structure diagrams

- **Class diagram**
- **Object diagram**
- Component diagram
- Composite structure diagram
- **Deployment diagram**
- Package diagram
- Profile diagram

Behaviour diagrams

- **Use case diagram**
- **State Machine diagram**
- Activity diagram

Interaction diagrams

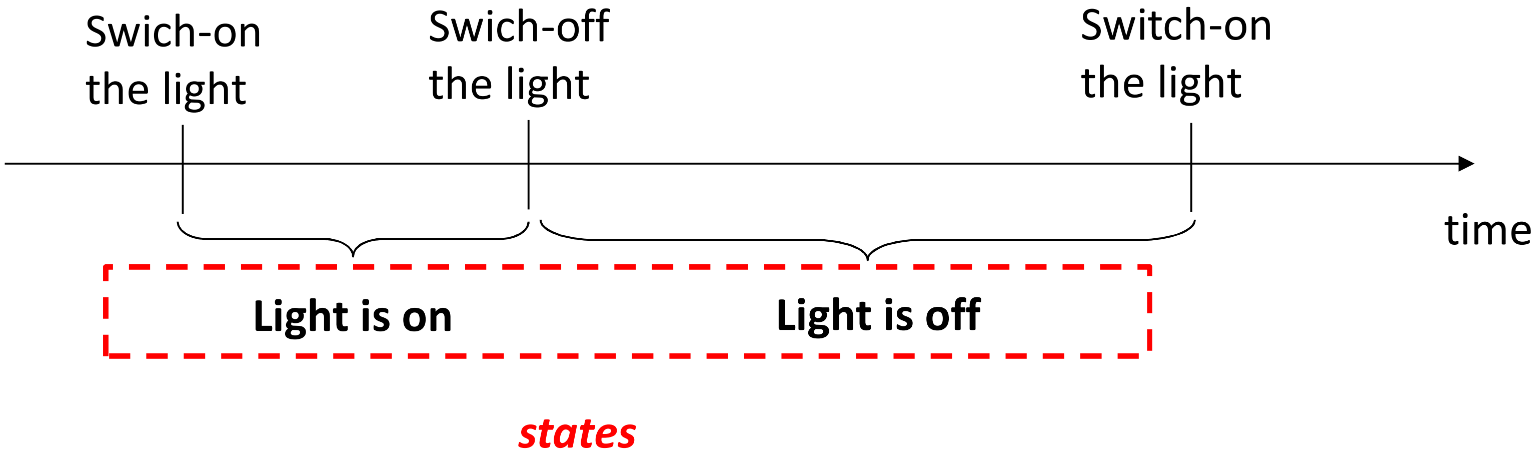
- **Sequence diagram**
- Communication diagram
- Interaction overview diagram
- Timing diagram

UML State Machine diagram (statechart)

- Focused on a specific object
- Describes the flow between states
- Mostly used to describe the states of a class
- Can be used during the analysis, design, …
- Main elements of the diagrams
  - States
  - Events : they trigger the state change
  - Transition : links two states, represented as arrow labeled with event

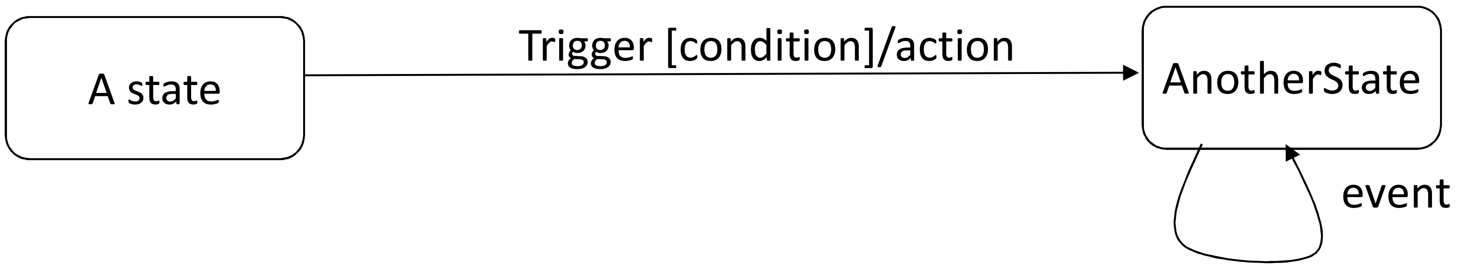
States in UML State machine diagram (statechart)

- It is a moment in the life cycle
- It corresponds to the presence or an absence of activity



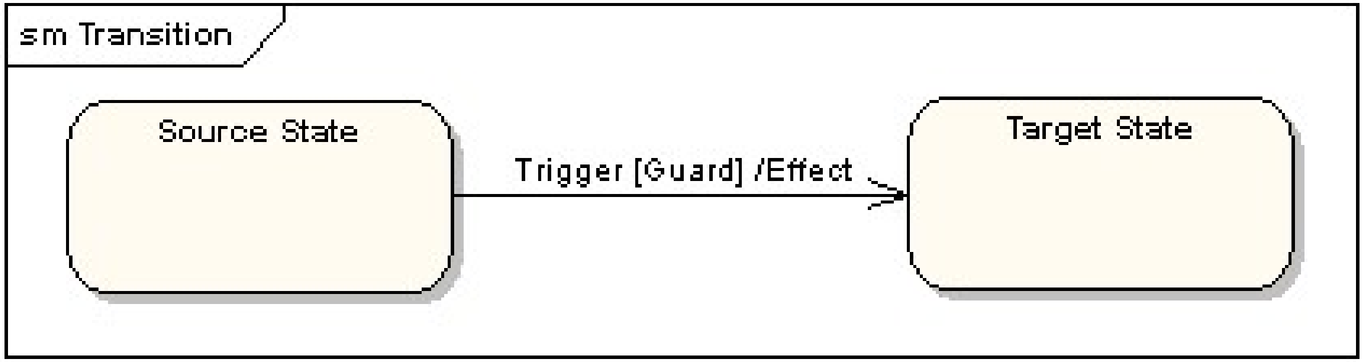
Transition

- Movement from one state to another one
- Drawn a **arrowed** line
- A transition is always in response to an event (trigger)
- Some condition and action can be handle by transition



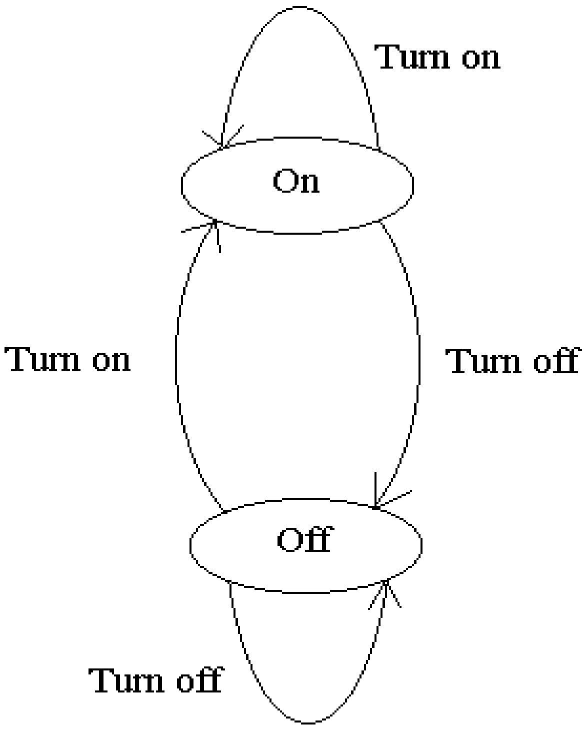
Transitions : Triggers [Guard] / Effect

- **Trigger** = cause of the transition
- **Guard** = condition which must be true in order for the trigger to cause the transition
- **Effect** = action which will be invoked directly on the object that owns the state machine as a result of the transition

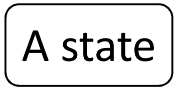


Example 1: a light

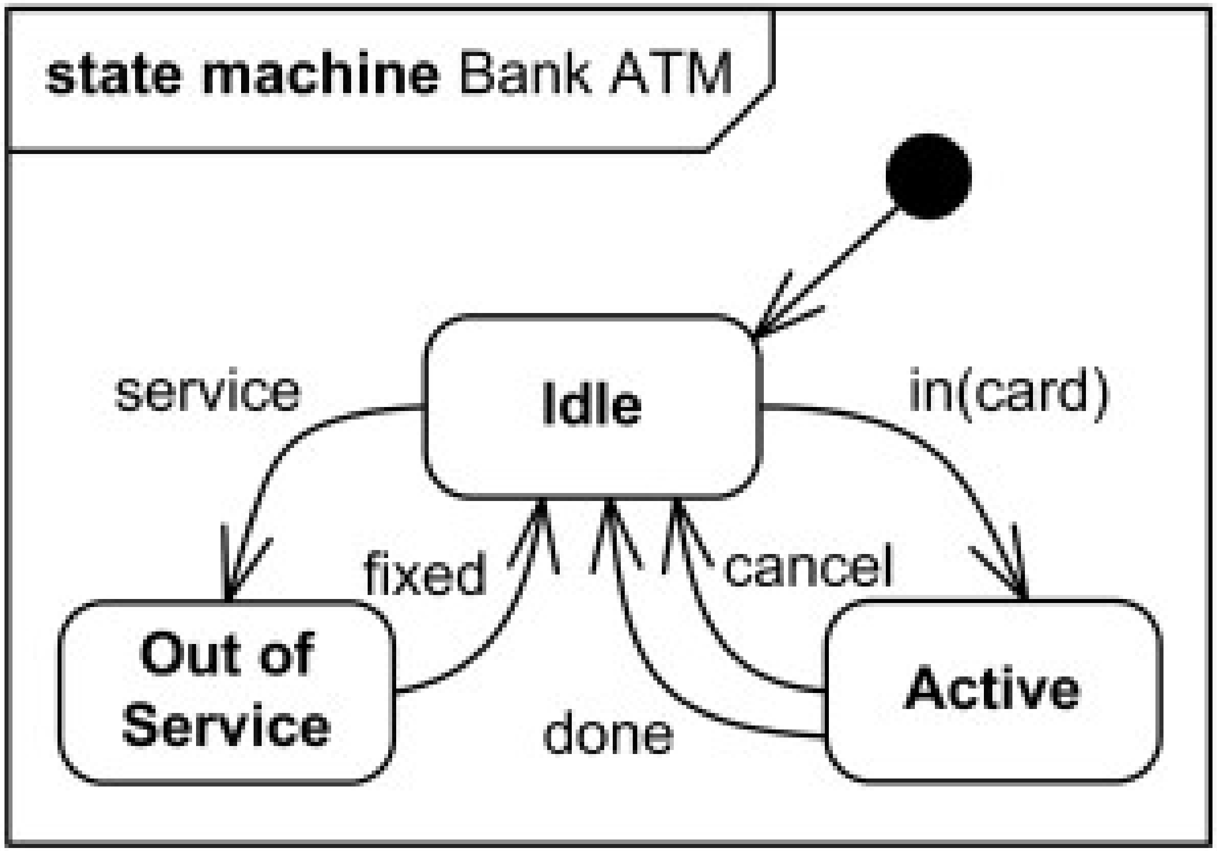
A light is « on » or off (states).
The light can be « turned on » or « turned off » (triggers).
When a light is on, nothing happens if it is turned on
(the state do not change).



States in UML State diagram (statechart)

- Initial ●
- End ◉
- Simple 
A state

Example 2



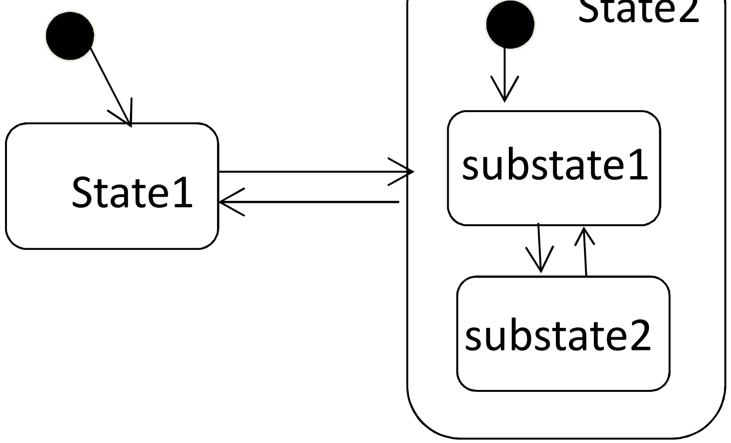
High level behavioral state machine for bank ATM

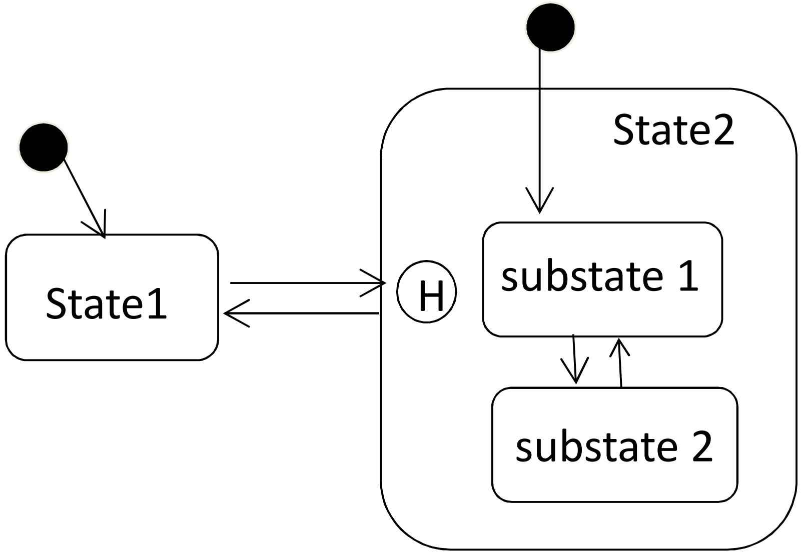
States in UML State diagram (statechart)

- Initial ●
- End ◉
- Simple A state
- Composite
  - hierarchically nested states and
  - orthogonal regions

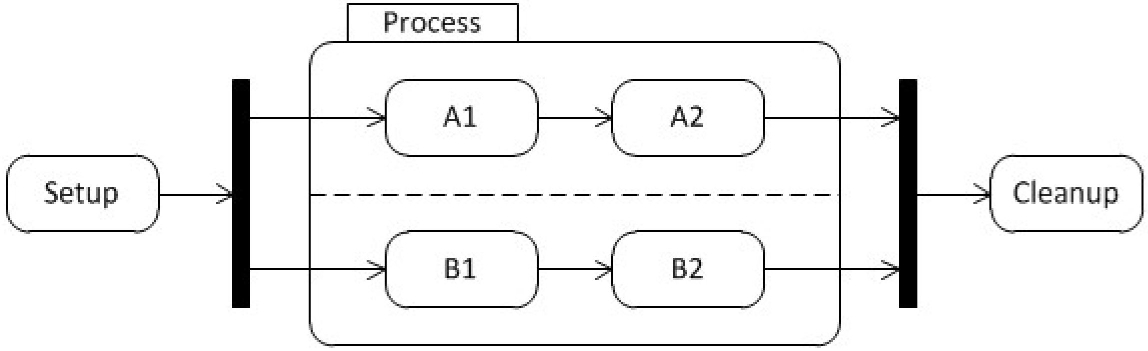
Composite states

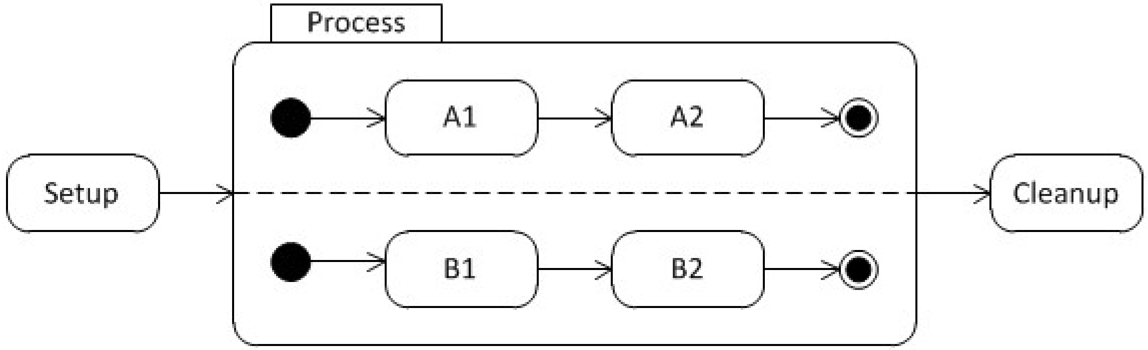
- A composite state
  - Can have one, several or none states
- When leaving a state,
  - It is possible or not to keep memory of the left state





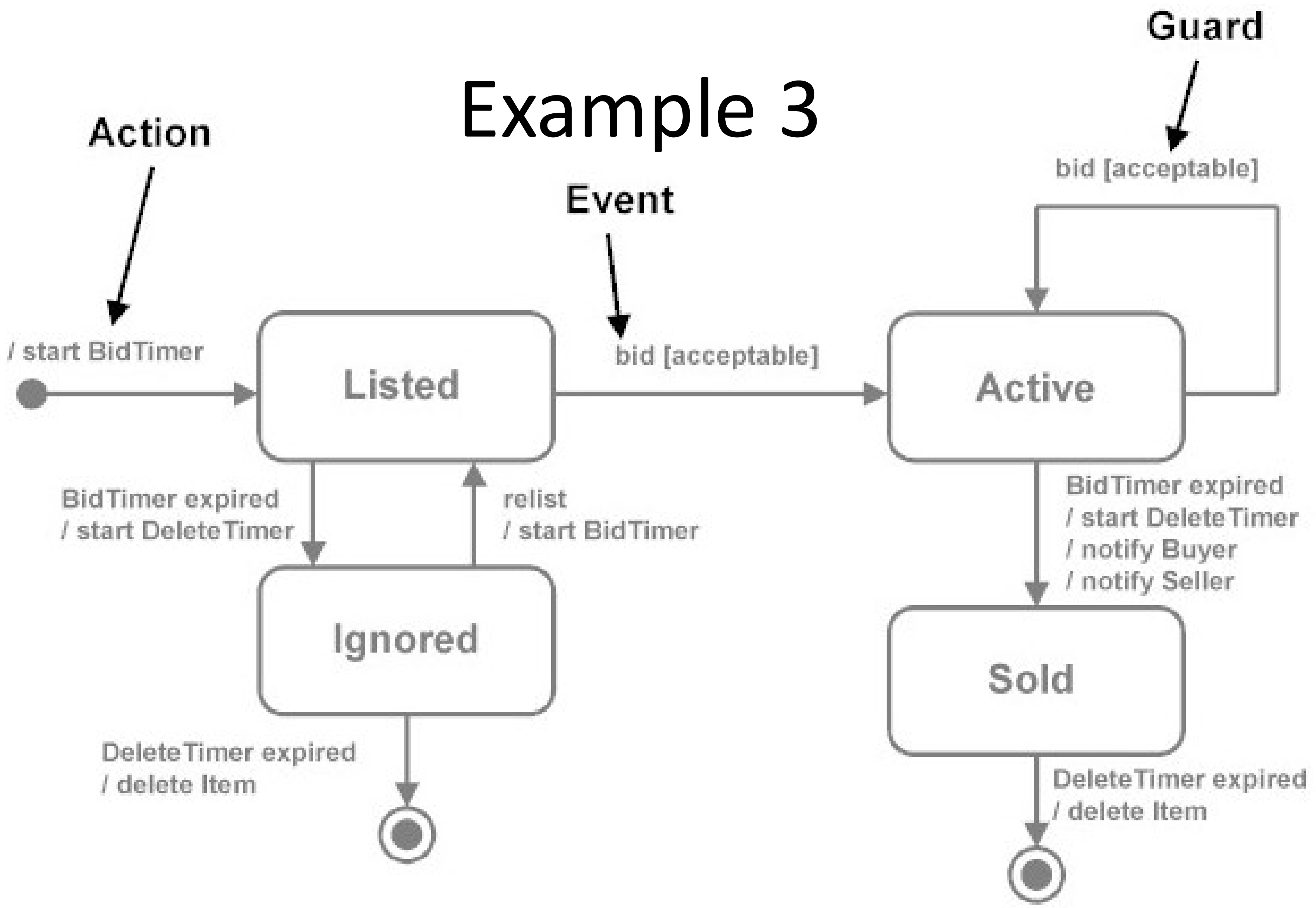
Composite State with parallel activities





« UML, second edition », Schaum's Outline. S. Bennett, J. Skelton, K. Lunn

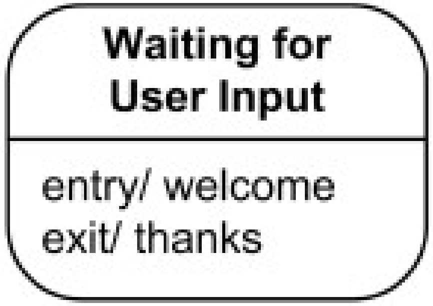
Example 3



https://www.stickyminds.com/article/state-transition-diagrams

States and labels

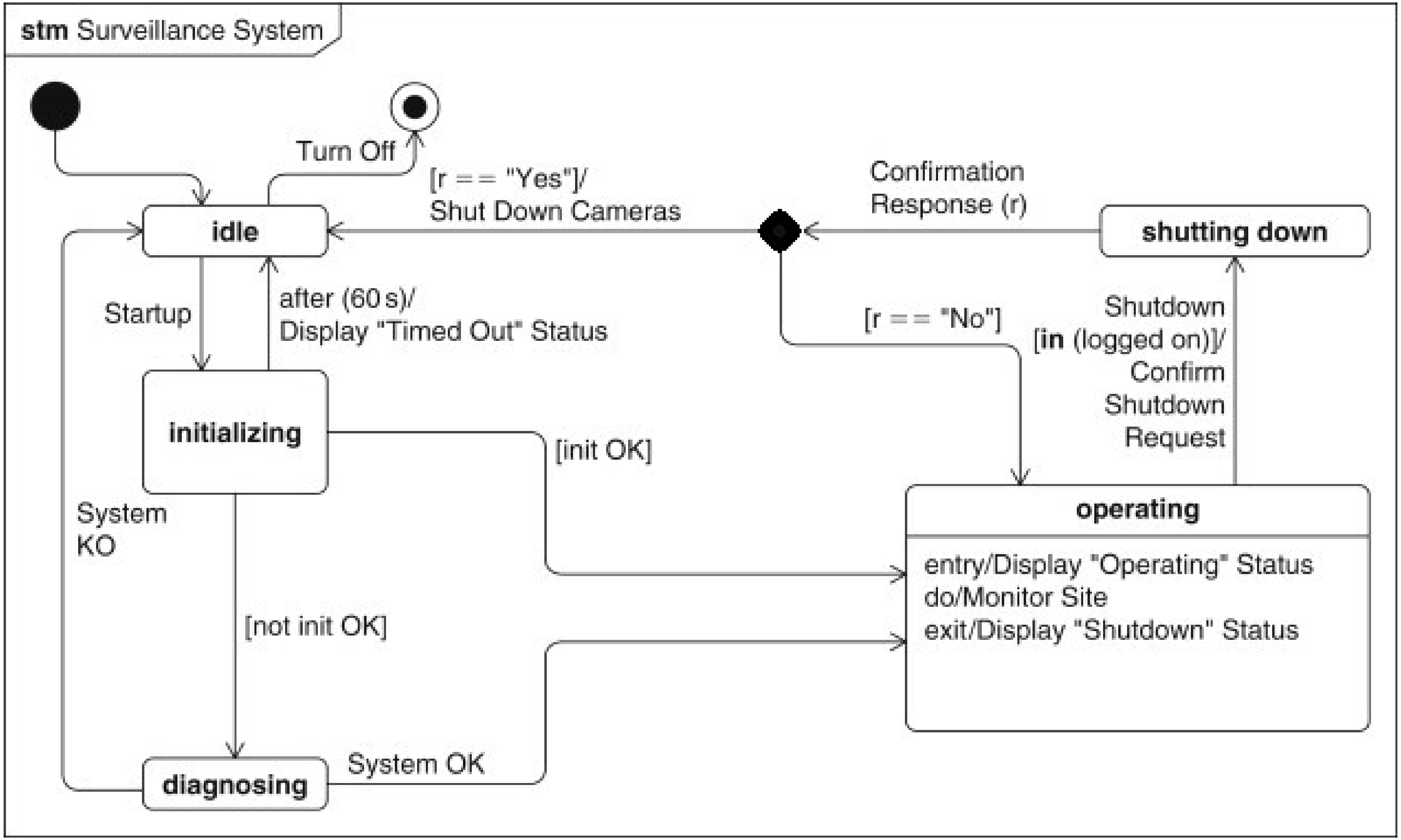
- **Entry**: behavior performed upon entry to the state
- **Do**: ongoing behavior, performed as long as the element is in the state
- **Exit**: behavior performed upon exit from the state



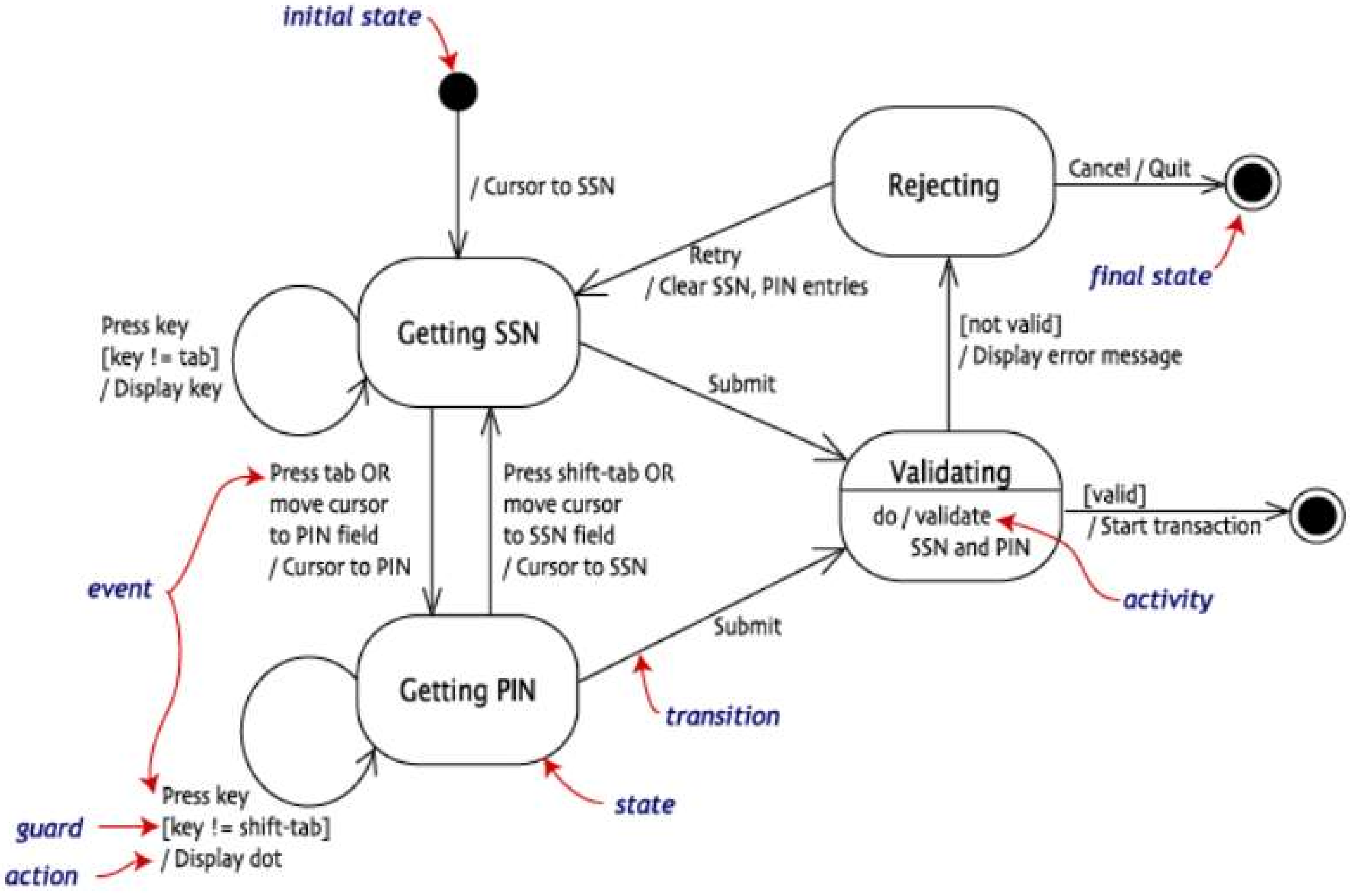
Waiting for User Input

entry/ welcome
exit/ thanks

Example 4


stm Surveillance System
Turn Off
[r == "Yes"]/
Shut Down Cameras
Confirmation
Response (r)
idle
shutting down
Startup
after (60 s)/
Display "Timed Out" Status
[r == "No"]
Shutdown
[in (logged on)]/
Confirm
Shutdown
Request
initializing
[init OK]
System
KO
operating
entry/Display "Operating" Status
do/Monitor Site
exit/Display "Shutdown" Status
[not init OK]
diagnosing
System OK

UML State Machine diagram



https://people.eecs.ku.edu/~hossein/810/Readings/UML-diagrams.pdf

Exercises

4.1

4.2

4.4

For the evaluations,
you should be able to

- @Midterm
  - Understand a diagram and reformulate it in a textual description
  - Produce a diagram from a simple description
- @Final
  - Reformulate a medium size textual description with the appropriate UML diagram