



**PXL – IT**

# **42TIW1030 Software analysis - Model Based Documentation of Requirements**

**Week 10 – semester 01**

**Luc Doumen**

**Nathalie Fuchs**

**DE HOGESCHOOL  
MET HET NETWERK**

Hogeschool PXL – Elfde-Liniestraat 24 – B-3500 Hasselt  
[www.pxl.be](http://www.pxl.be) - [www.pxl.be/facebook](https://www.pxl.be/facebook)



# Content

- Models & using models
- Topic overview
  - Goal models
  - Use models (system use cases)
  - 3 perspectives on requirements
    - Data/structural: ERM, Class models
    - Functional: Data Flow Diagram, Activity diagram
    - Behavioral: State charts
  - Sequence diagram
- Key learning points
- Questions & answers



## Models & using models



# Topic Overview

- Goal models
- System use case models (+ descriptions)
- **Three perspectives on requirements**
  - **Data perspective → to read, cf. 1TIN**
    - ERM, Class models (UML)
  - Functional perspective
    - Data flow diagrams, Activity diagrams (UML)
  - Behavioral perspective
    - State charts
- Sequence diagrams

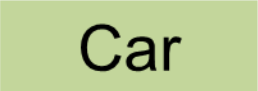
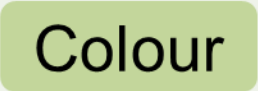


# Data/Structural Requirements Models

- Documenting the **structure of the systems' data** as well as the **usage** and **relationships inside** the system context
- Traditionally: ERM ➔ Entity Relationship Models
- Today also substituted by Class models (UML)

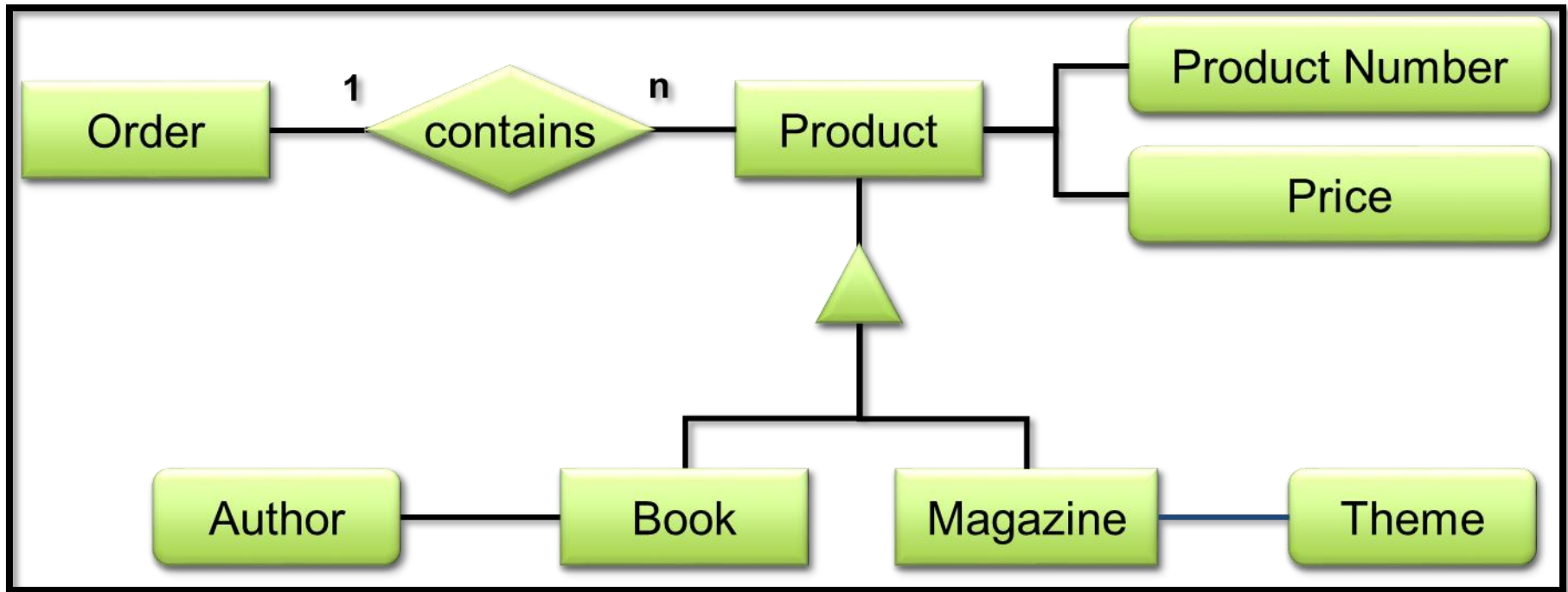
# Entity Relationship Model (ERM)

- Models the structure of data (static)
- Not for functions, behavior, control
- Widely used for (relational) databases
- Clarifies breakdown (structure, hierarchy)
- Understandable (user communication)
- Typical modelling elements:
  - Entity types
  - Relationship types
  - Attributes

# ERM - Elements

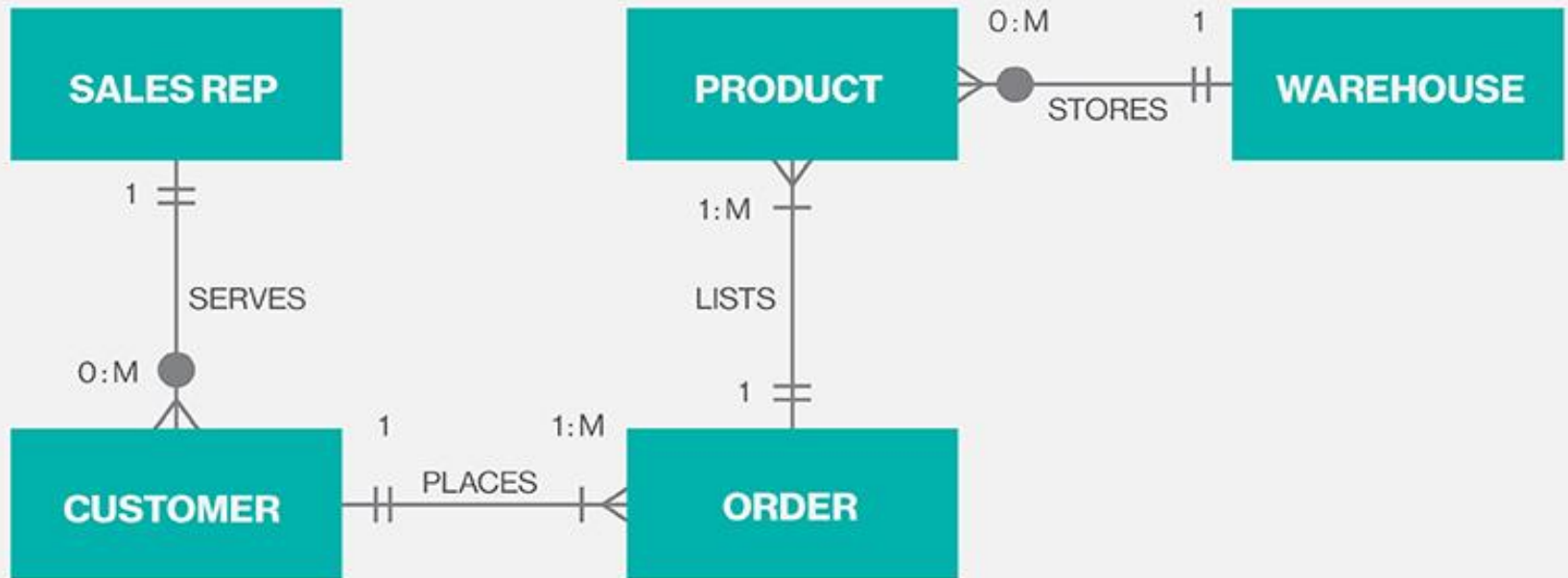
Symbol	Term	Examples
 Car	Entity	Order, Employee, Table
 Colour	Attribute	Price, Address, Size
	Generalization	Vehicle - {car, truck}
 contains	Relation	Order contains Article
(min,max)	Cardinality	(1,n) , 1 , (0,n)

# ERM - Example





# ERM – Another example

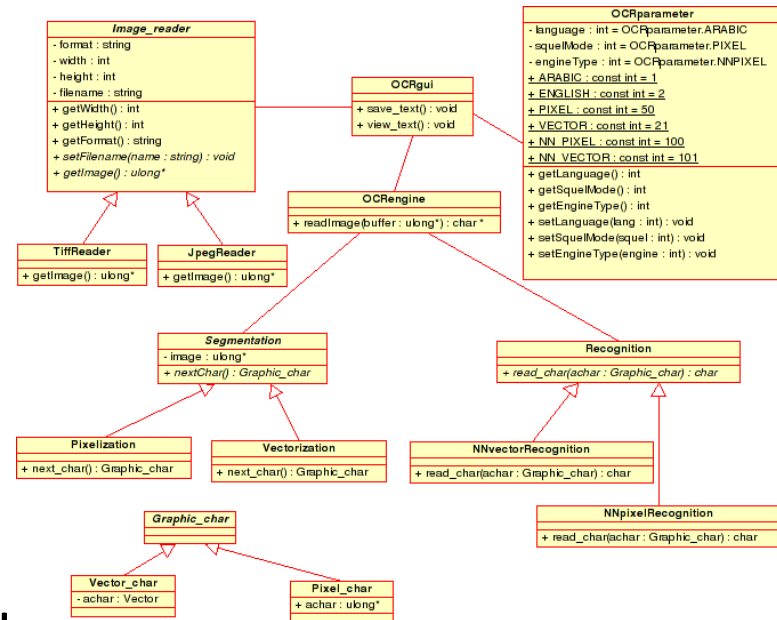


**Figure 1. Entity-Relationship Diagram**

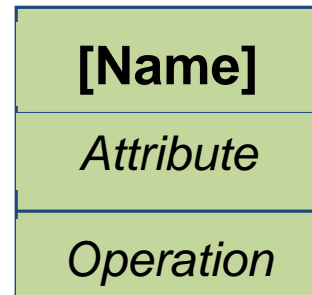
- \*1 INSTANCE OF A SALES REP SERVES 1 TO MANY CUSTOMERS
- \*1 INSTANCE OF A CUSTOMER PLACES 1 TO MANY ORDERS
- \*1 INSTANCE OF AN ORDER LISTS 1 TO MANY PRODUCTS
- \*1 INSTANCE OF A WAREHOUSE STORES 0 TO MANY PRODUCTS

# Class Models (UML)

- To model static structures
  - Data, dependencies, relationships, constraints
- Domain specific knowledge
- Different levels
  - Conceptual, specification, implementation
- Tools: UML modelling tool (e.g. Rose)
- Frequently used modelling elements:
  - Classes
  - Associations (multiplicities and roles)
  - Aggregation, Composition, Generalization



# Class Models: Elements



Classes

0...1 

m 

n...m 

1...\* 

[Name]

Association

Generalization

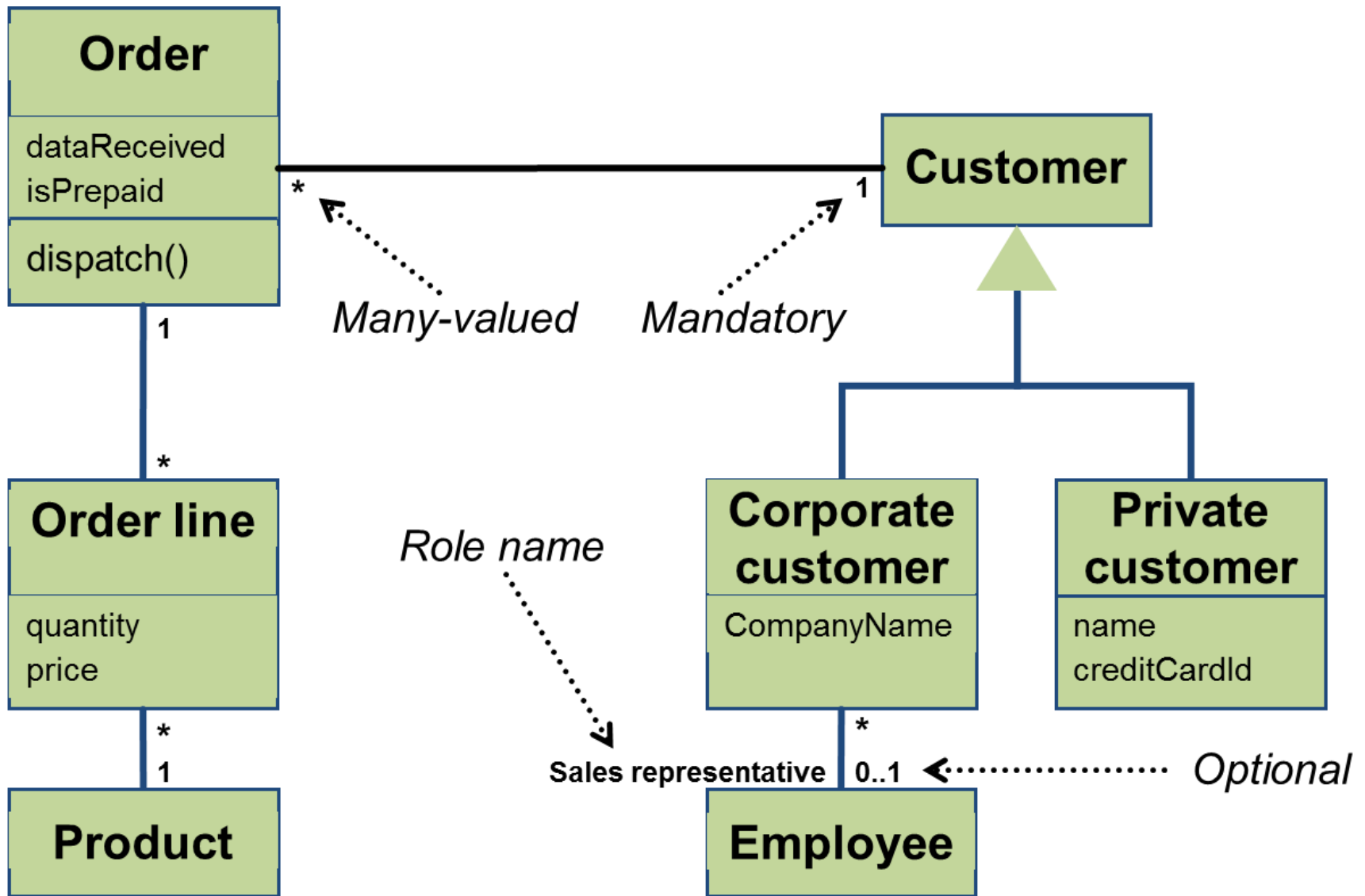


Aggregation



Composition

# Class Models: example

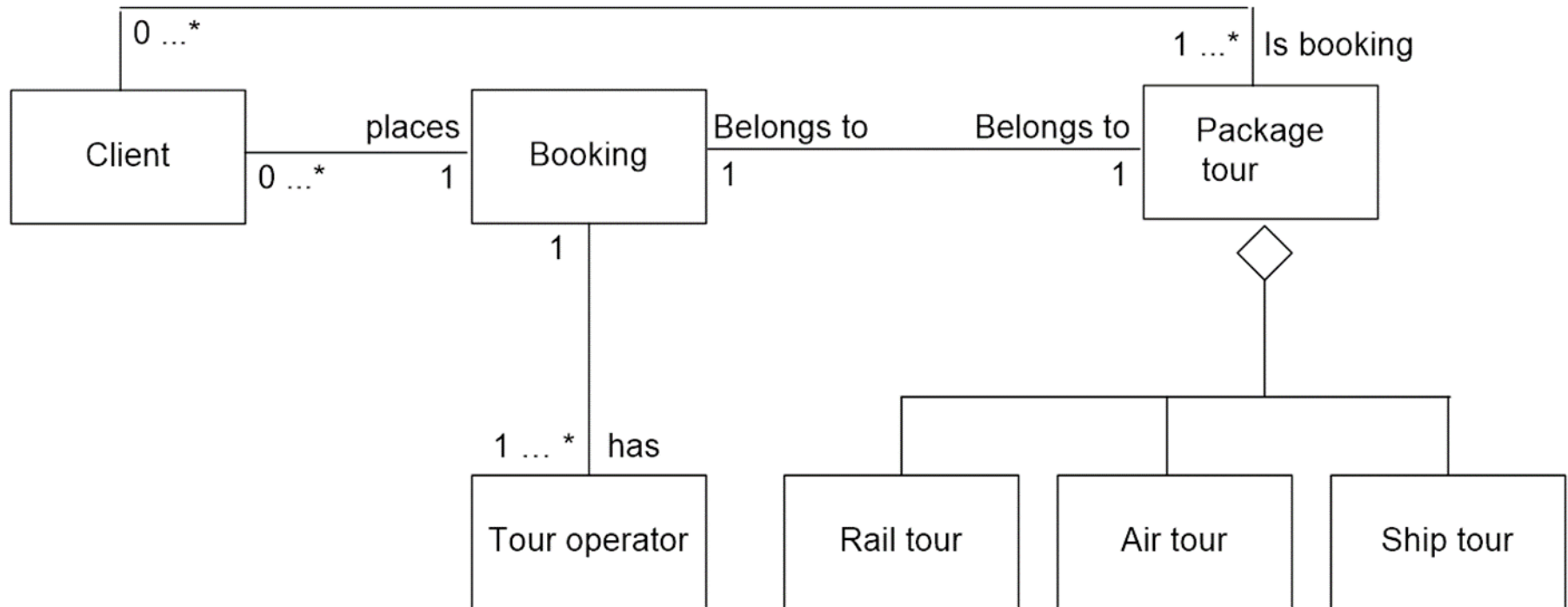


# Analysis: Identifying classes

Type	Candidate	Examples
Noun	Class	Car, Motor, Tyre
Adjective	Attribute	Colour, Size, Address
Verbs	Relation	Submits, Buy, Has, Contains

# Class diagram - Case Booking System (1)

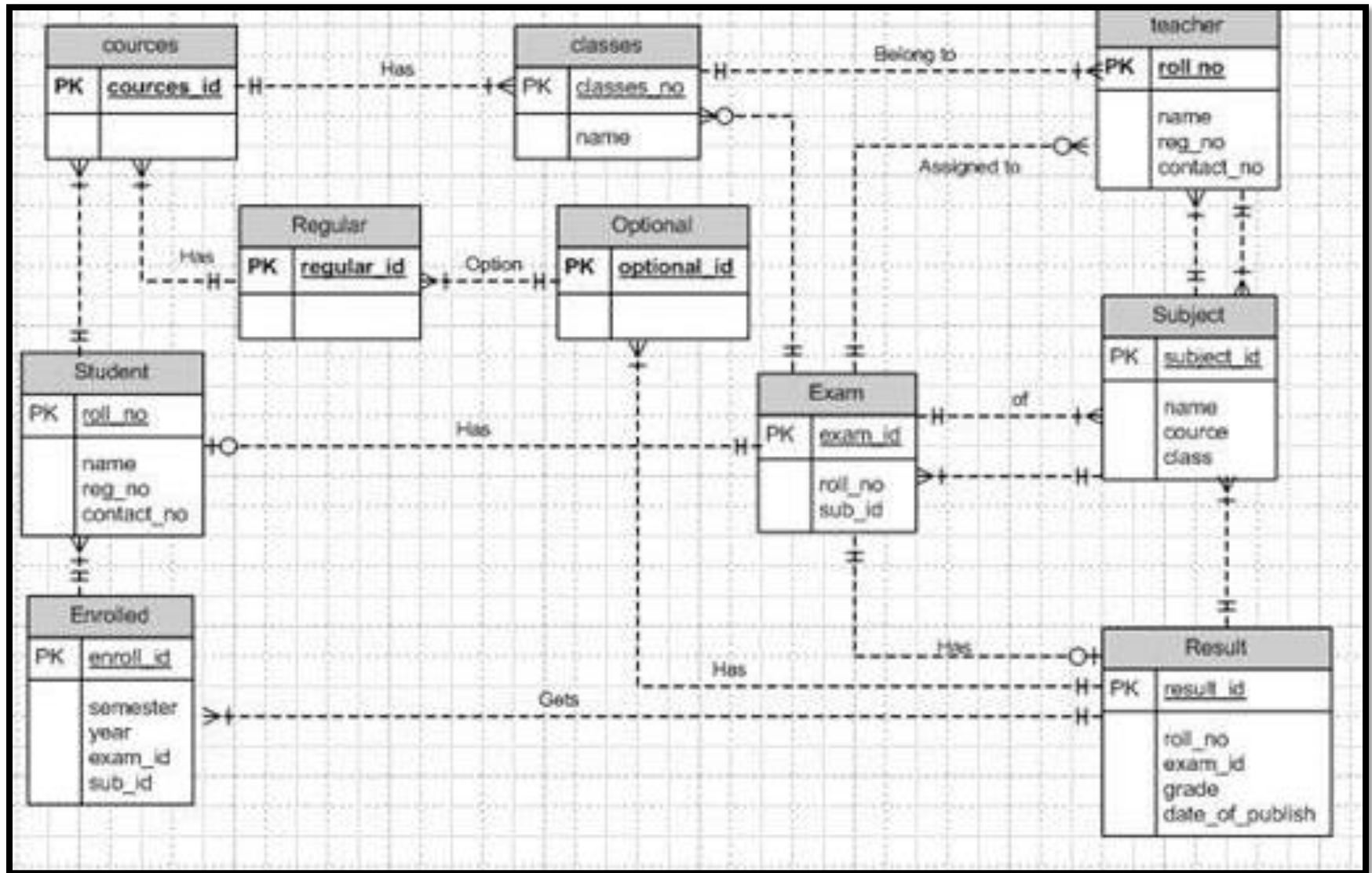
- Review the class diagram and identify the defects, if any.



# Class diagram - Case Booking System (2)

- True or False?
  - The association between client and package tour is redundant and also inconsistent; it should therefore be removed from the model.
  - The association between booking and tour operator makes no sense and should be replaced by a association between package tour and tour operator.
  - The client statement that the package tour sold can be either a rail, air or ship tour is correctly modelled.
  - The multiplicity of the association between client and booking adequately reflect reality.

# Class diagram - Explain





# Topic Overview

- Goal models
- System use case models (+ descriptions)
- **Three perspectives on requirements**
  - Data perspective
    - ERM, Class models (UML)
  - **Functional perspective**
    - Data flow diagrams, Activity diagrams (UML)
  - Behavioral perspective
    - State charts
- Sequence diagrams

# Functional Requirements Models

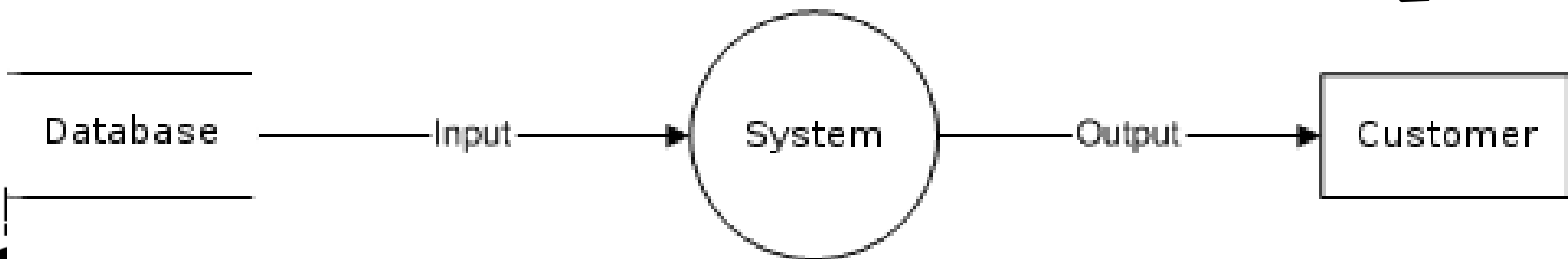
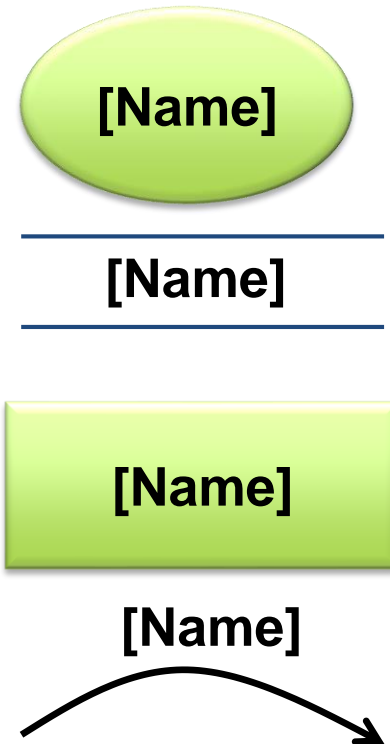
- Focus is on processing input data from the environment of the system to create output data of the system
- Most abstract form: System Use Case Model
- Others:
  - Data flow diagram
  - Activity diagram (UML)

# Data Flow Diagram

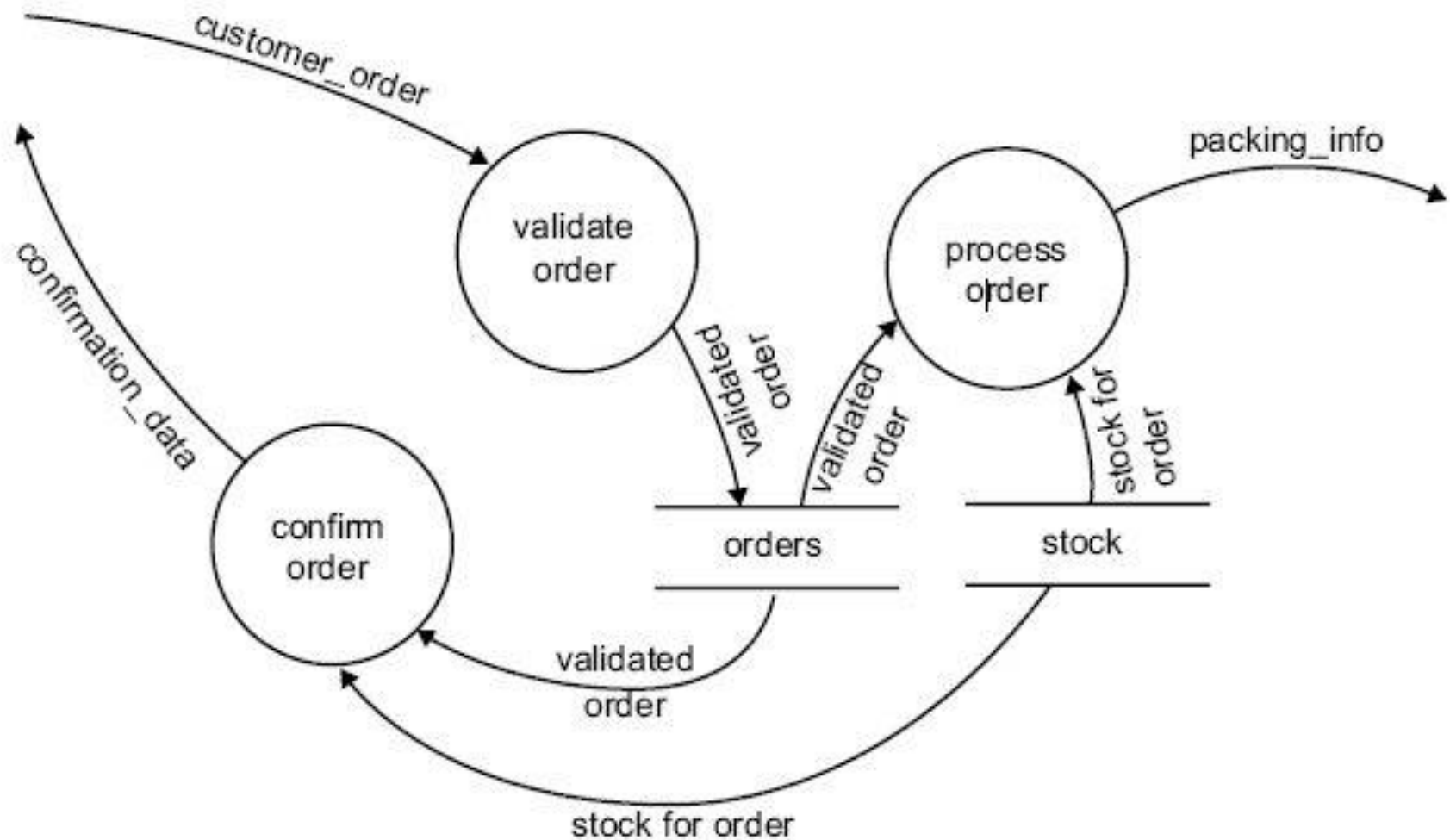
- A data-flow diagram (DFD) is a graphical representation of the **"flow" of data** through an information system
- Provides no information about the timing or ordering, sequences or parallel behavior
- Levels: high level diagram, low level diagram
- Modeling elements:
  - Processes
  - Data flows, Data stores
  - Sources / sinks

# Data Flow Diagram: elements

- Node: Function / Process
- Data store: File / Database
- Terminator: Input / Output
- Data flow



# Data Flow Diagram: example



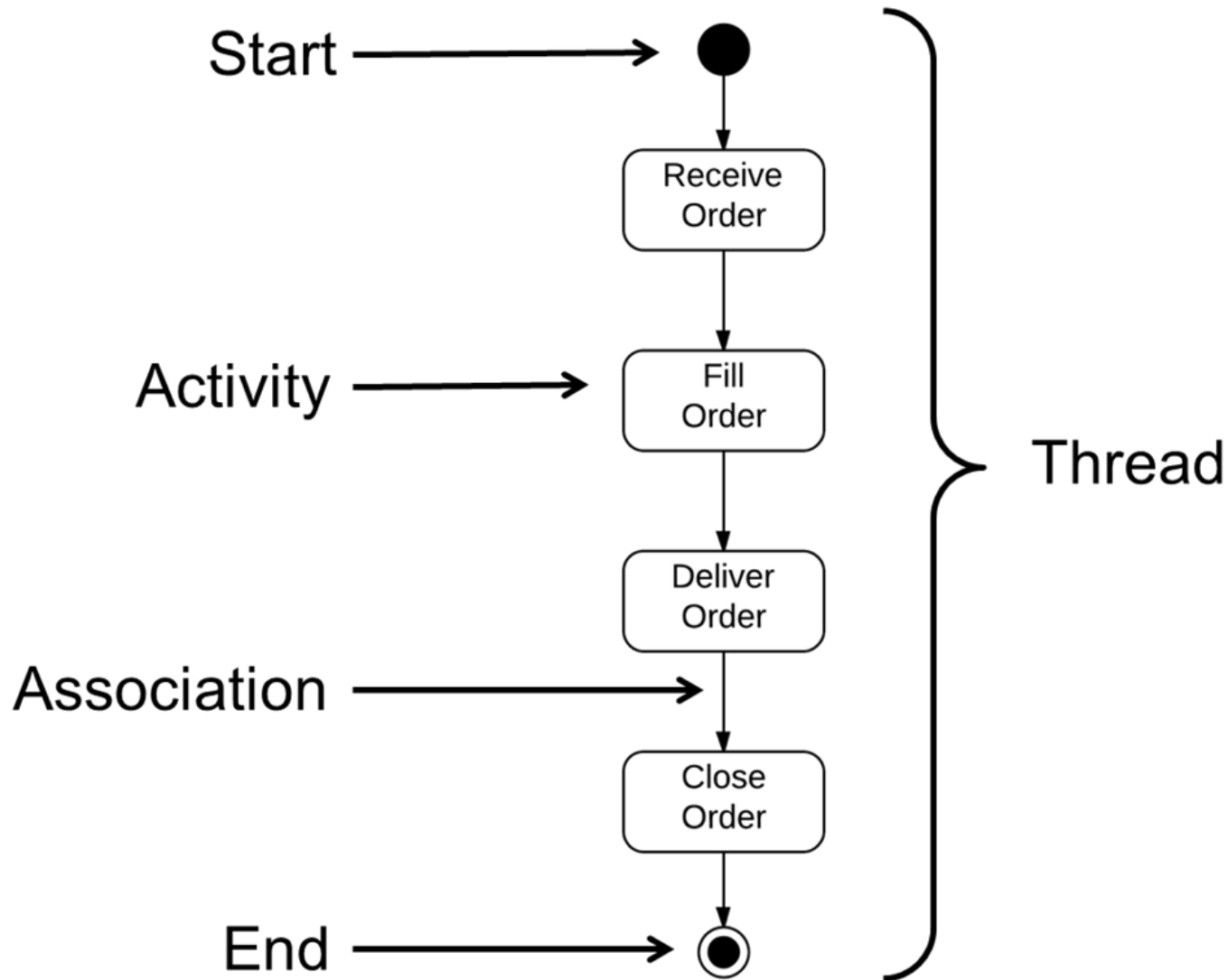
# Activity Diagram



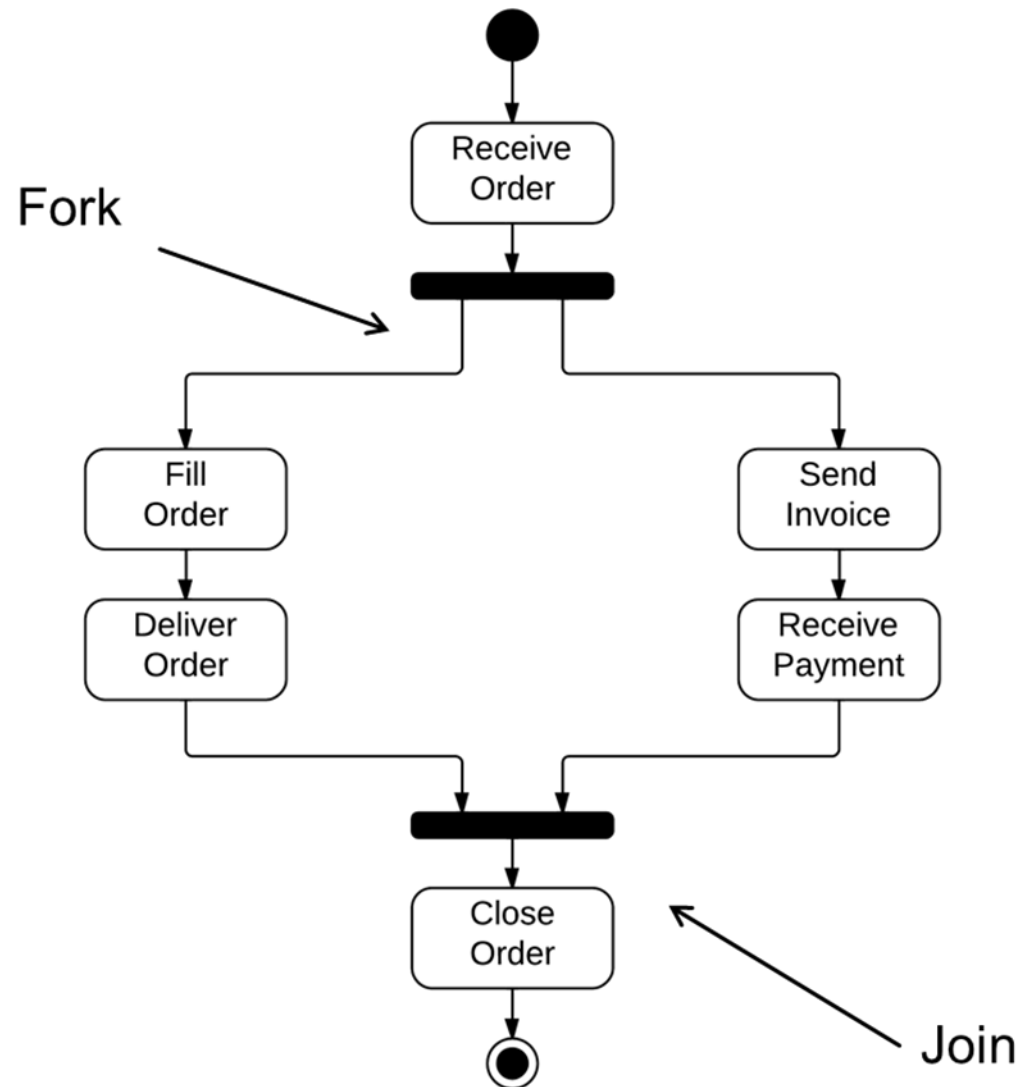
- Focus on activities of a process and the control between them
- Describes the sequencing of activities
- Procedure resulting from individual actions
- Threads may be parallel or conditional
- Often used to clarify/detail the scenarios (normal, alternative and exception flows) of the use cases

**Essential modeling elements: actions, start- and end nodes, control flow, object flow, decision nodes, merge of alternative control flows, fork, join, hierarchic elements**

# Activity Diagram - model elements

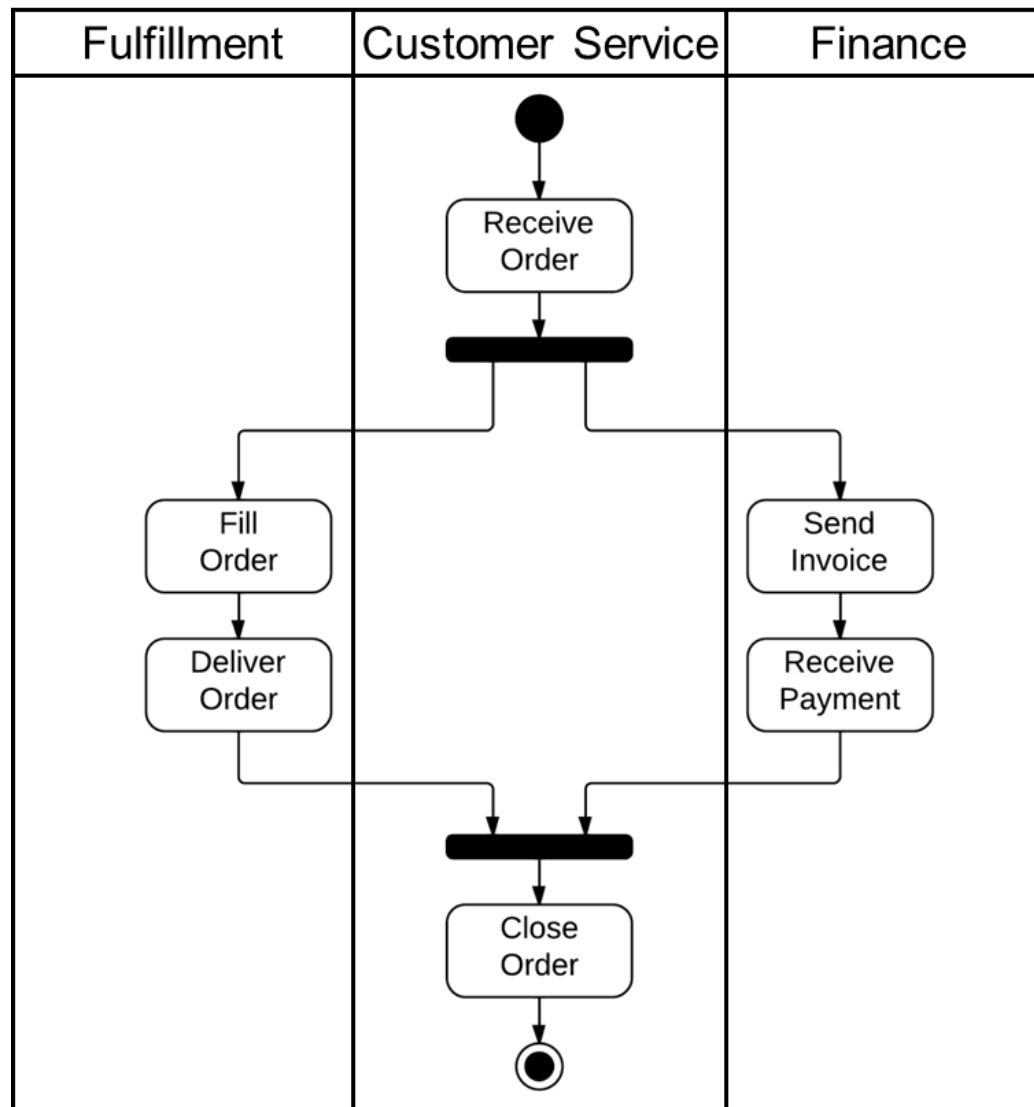


# Activity Diagram - parallel behavior

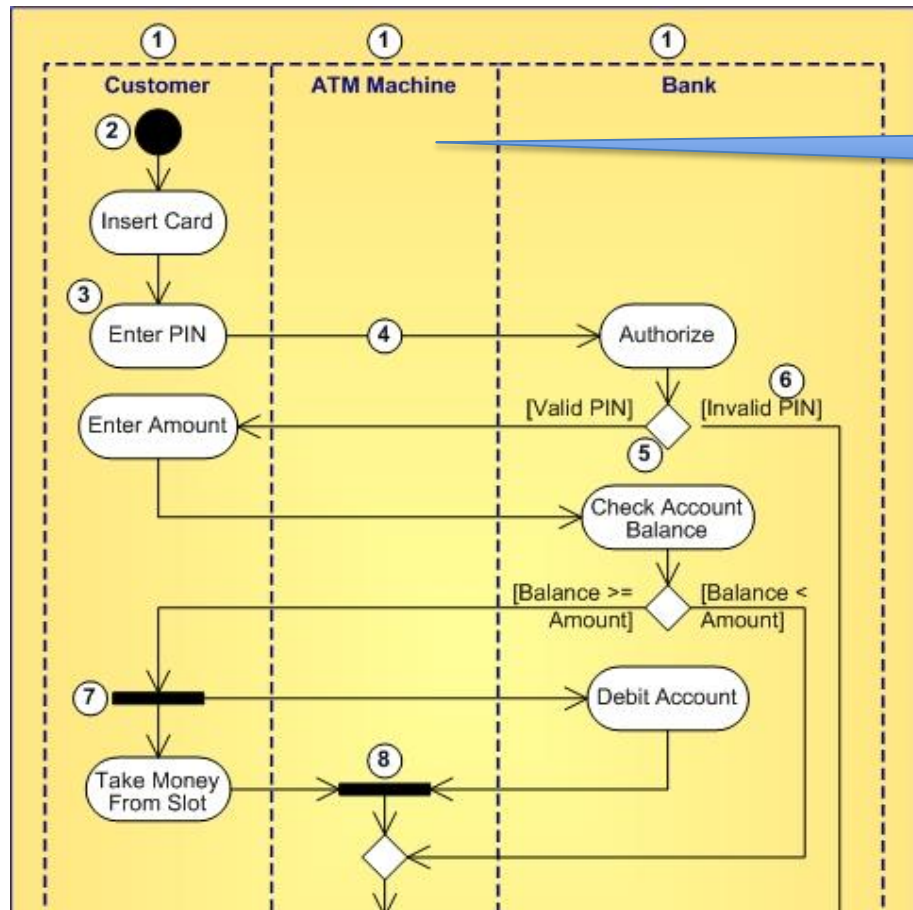




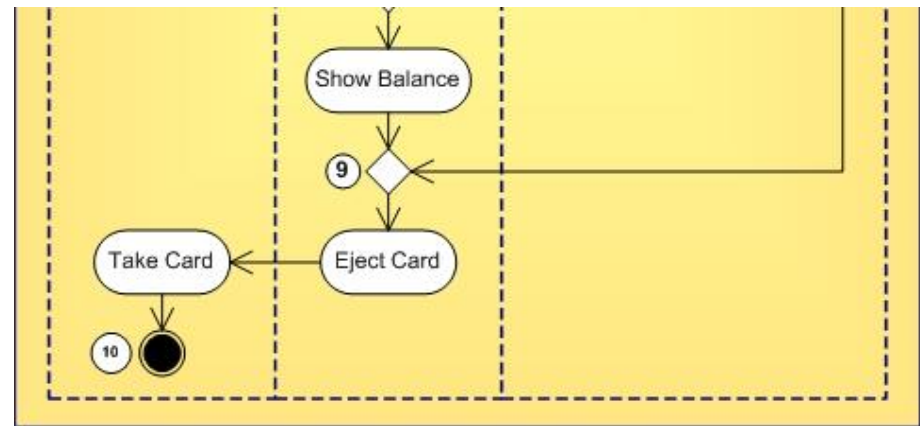
# Activity Diagram - swim lanes (1)



# Activity Diagram - swim lanes (2)

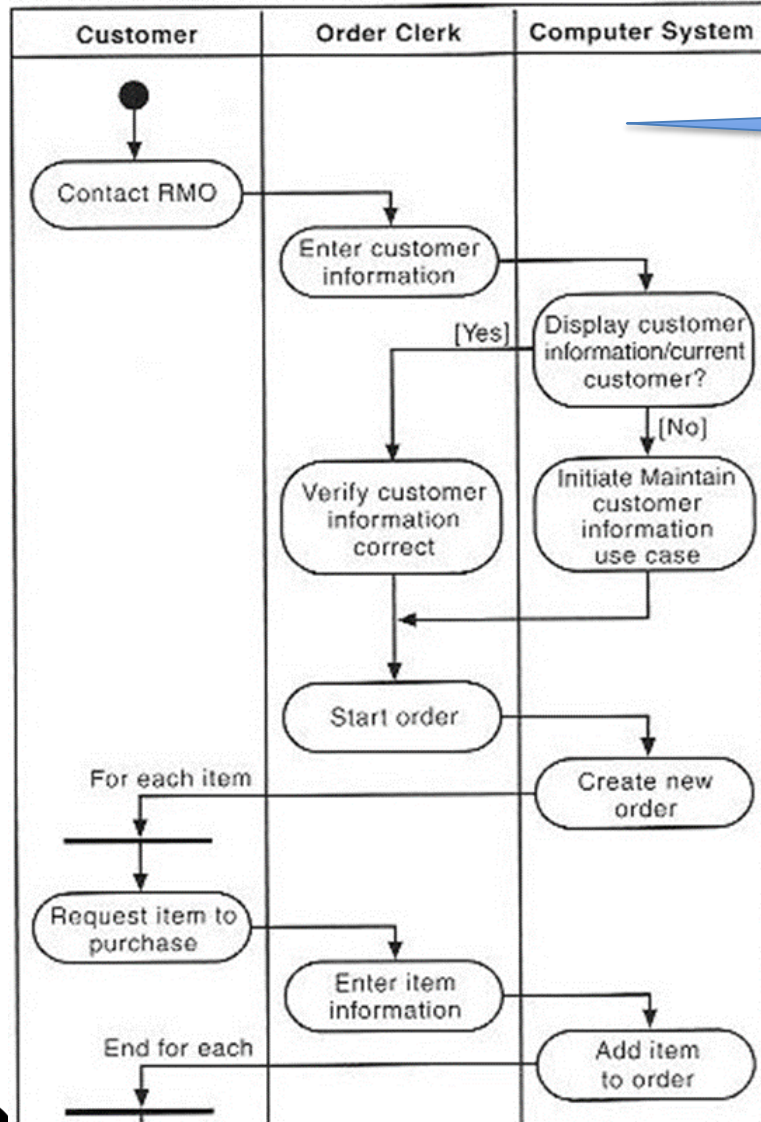


Can be a system

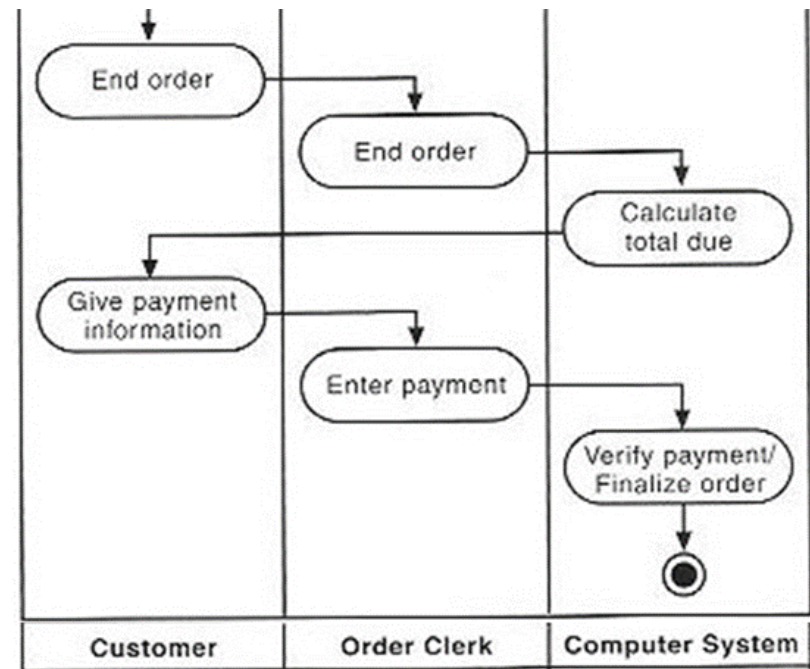


- |            |               |                    |        |         |
|------------|---------------|--------------------|--------|---------|
| ① Swimlane | ③ Activity    | ⑤ Branch           | ⑦ Fork | ⑨ Merge |
| ② Start    | ④ Transisiton | ⑥ Guard Expression | ⑧ Join | ⑩ End   |

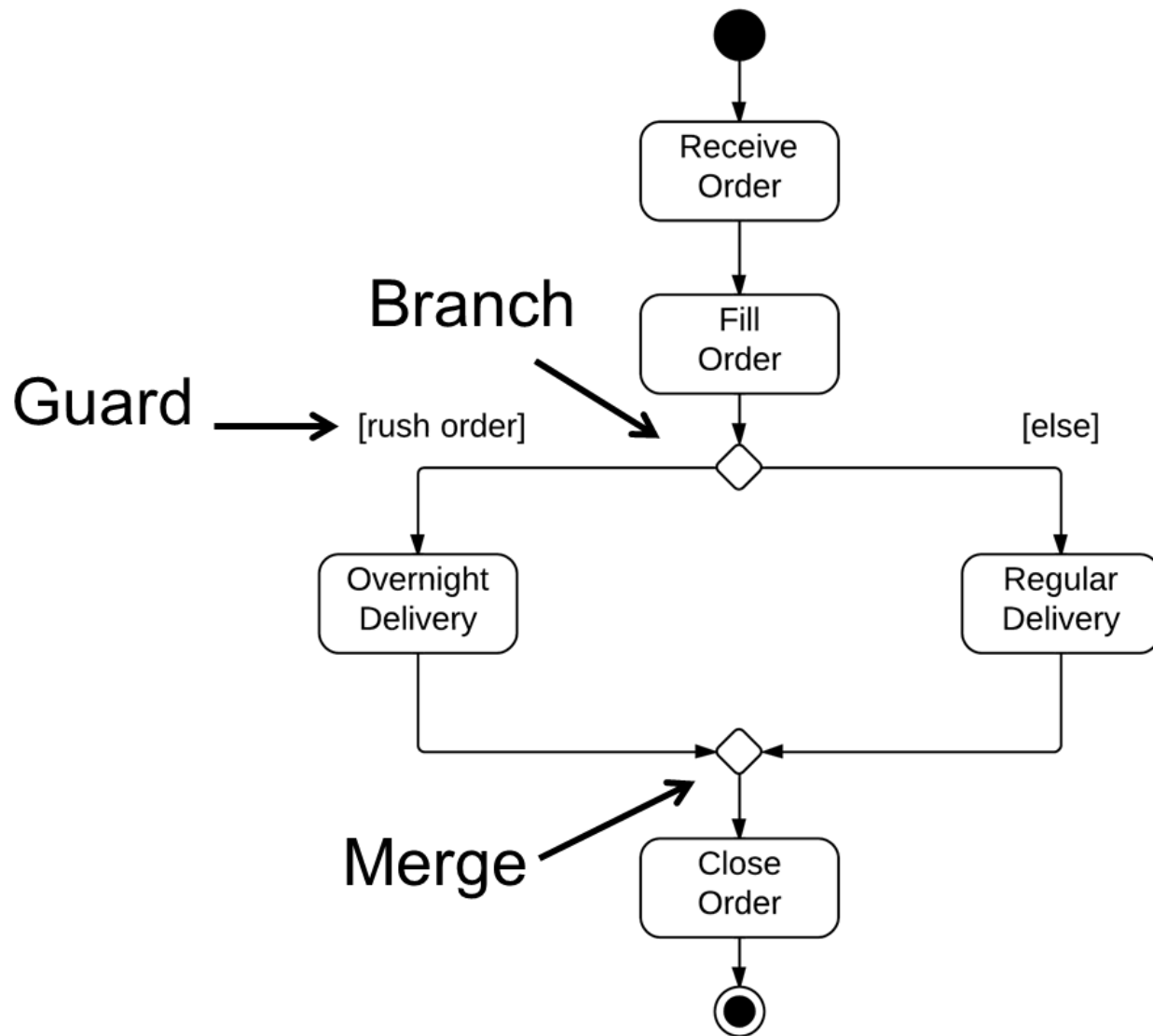
# Activity Diagram - swim lanes (3)



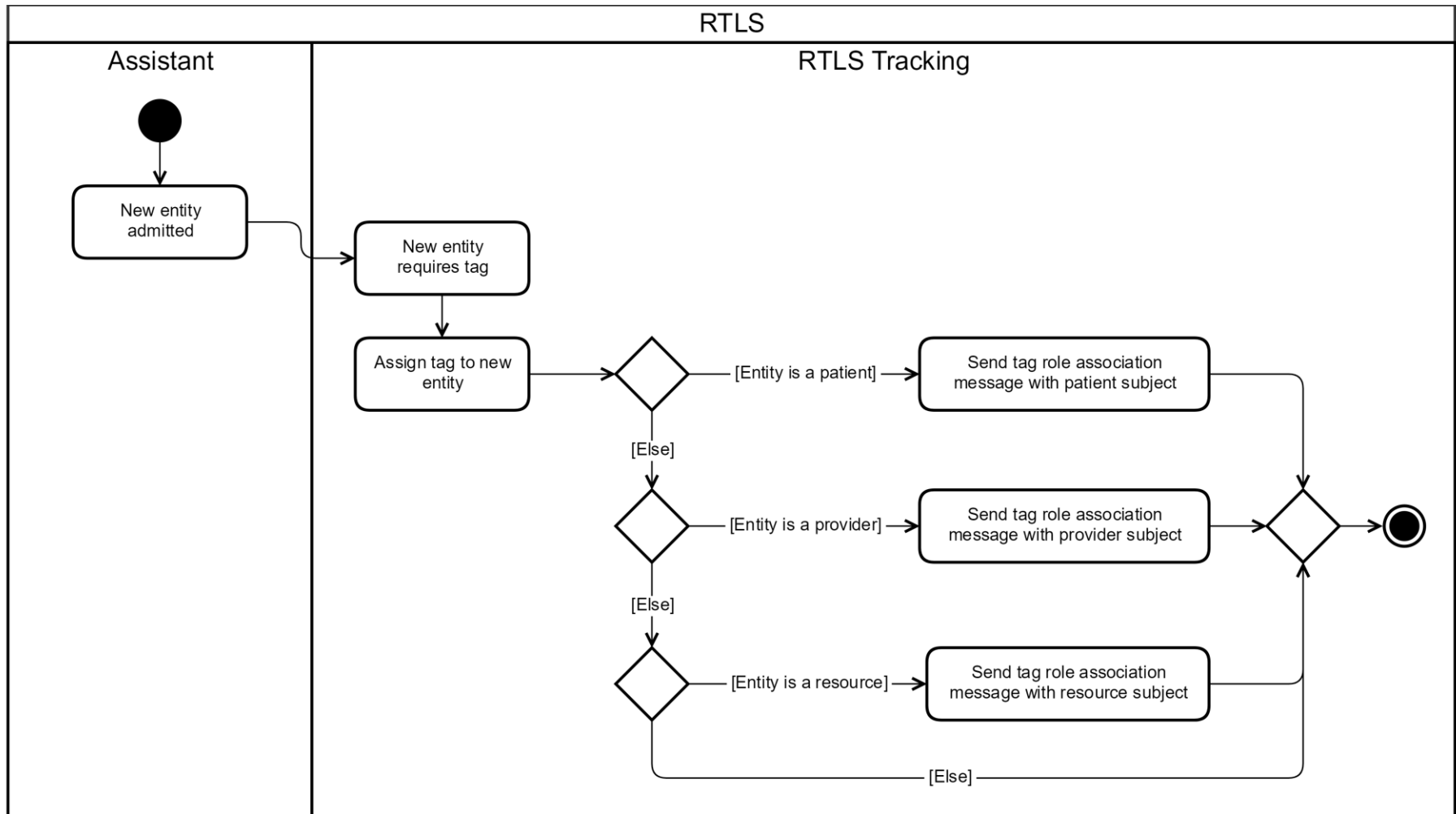
Can be a system



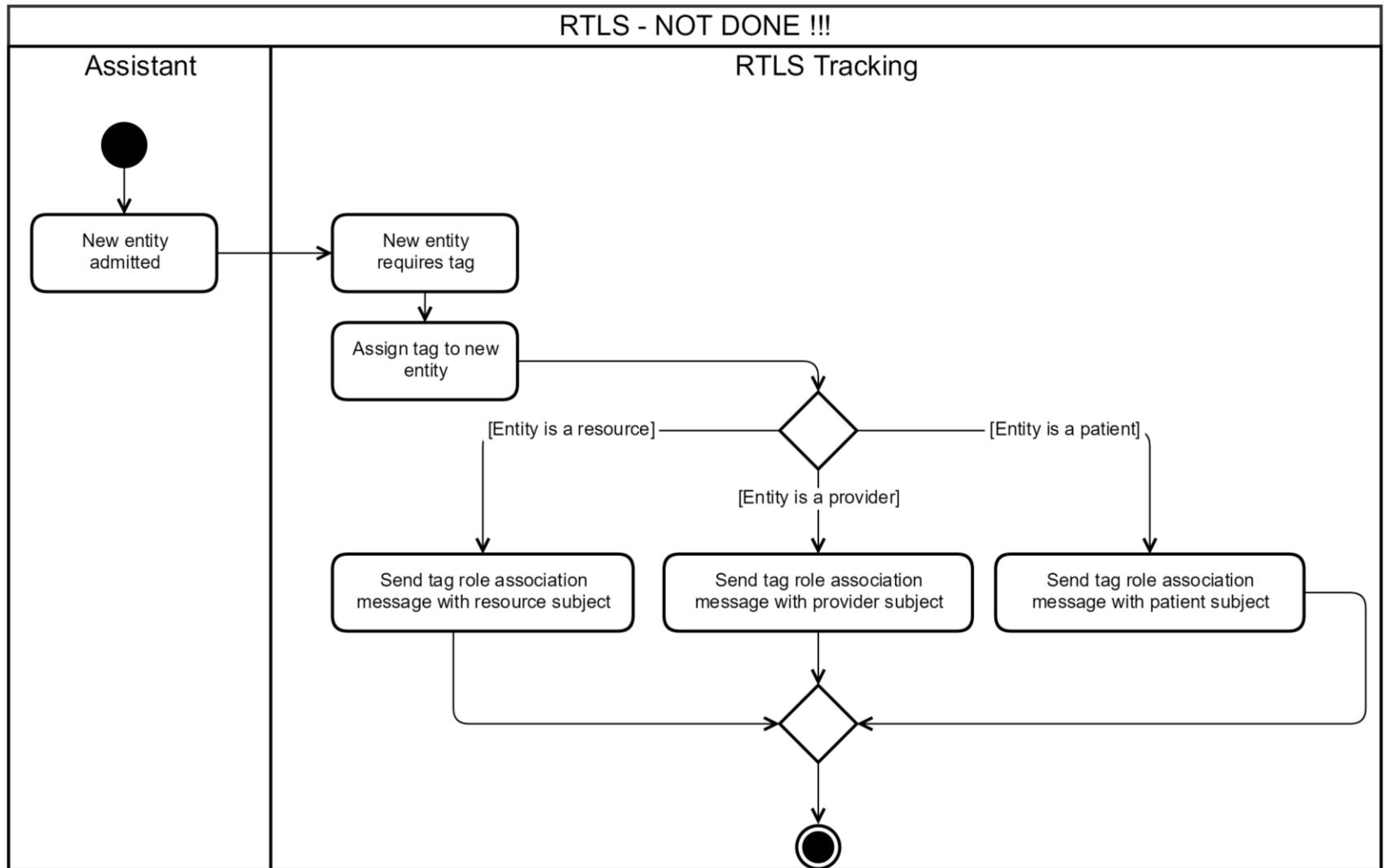
# Activity Diagram - conditional behavior



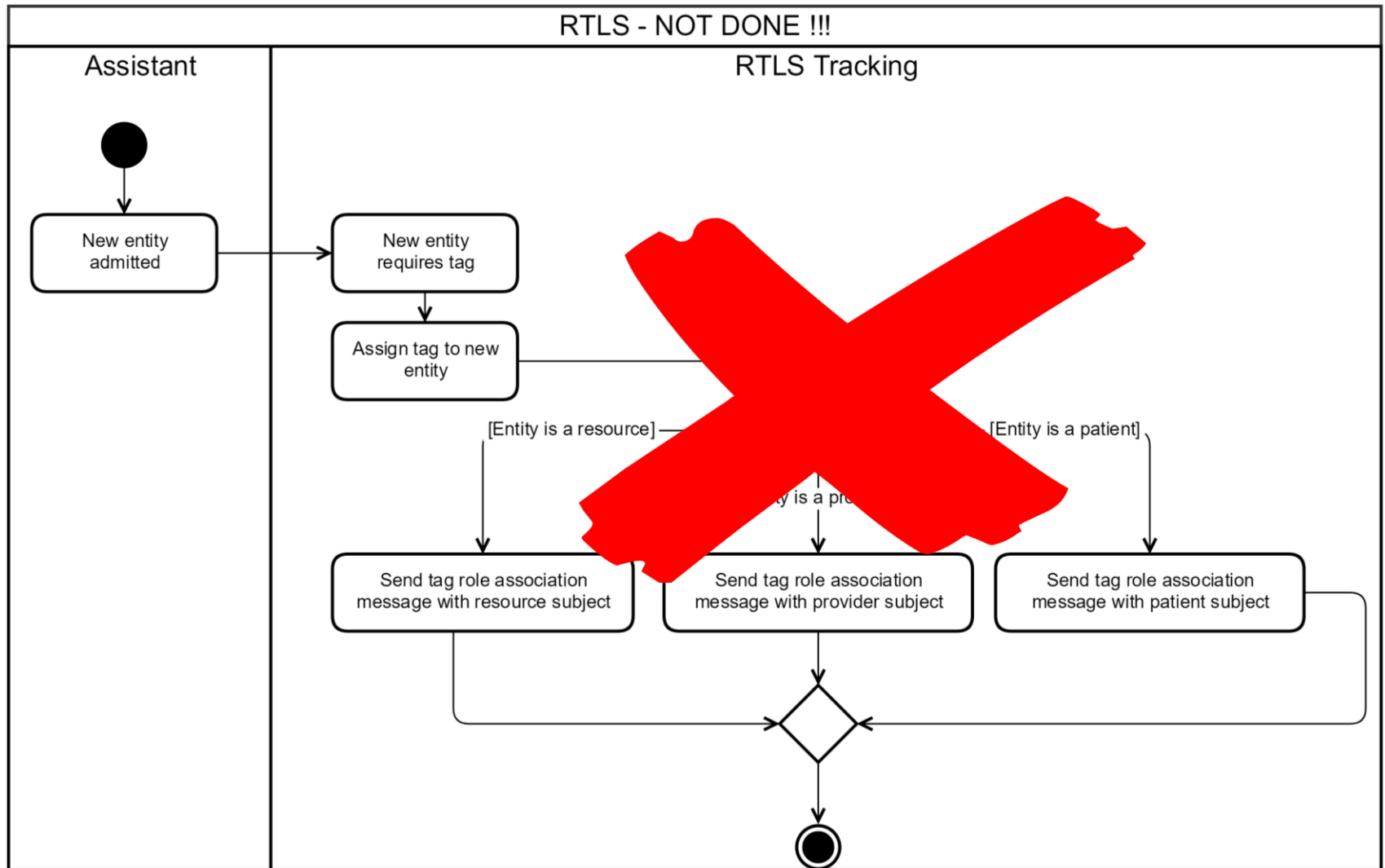
# Activity Diagram – condition ➔ switch case



# Activity Diagram – condition ➔ NOT DONE



# Activity Diagram – condition → NOT DONE



# Activity Diagram – how to draw (1)?

- Flow chart consisting of activities performed by system
- Activity diagram not exactly flow chart
  - Additional capabilities: branching, parallel flow, swim lane, etc.
- Before drawing activity diagram
  - Clear understanding about elements used
  - The main element = activity itself
    - Activity = function performed by system
- After identifying activities
  - Understand how activities associated with constraints and conditions



# Activity Diagram – how to draw (2)?

- So before drawing activity diagram, identify following elements:
  - Activities
  - Association
  - Conditions
  - Constraints
- Then draw diagram

# Activity Diagram – example (1)

- Example of activity diagram for order management system.
- 4 activities identified which are associated with conditions
  - Send order by the customer
  - Receipt of the order
  - Confirm order
  - Dispatch order
- After receiving order request, condition checks are performed to check if “normal” or “special order”
- After that dispatch activity = termination of process.

# Activity Diagram - pro's and cons

- 😊 Easy to learn
- 😊 Widely applicable, e.g. use case details
- 😊 Can show complex, time-related procedures
- 😞 Pitfall: specifying too many details
- 😞 50% of activities would (almost) be enough
- 😞 Little practical information in UML manual

# Activity Diagram – exercises “Aankoop”

- De aankoopdienst verwerkt aankooporders van andere departementen in het bedrijf.
- Werknemers die het aankooporder indienen, worden de klanten van de aankoopdienst genoemd.
- Een assistent van de aankoopdienst ontvangt het order en volgt de verdere afhandeling op, tot het order is besteld en ontvangen.
- Indien de assistent een order ontvangt onder de 1.500 EUR, mag deze onmiddellijk een bestelbon opmaken en naar de vooraf goedgekeurde leverancier sturen.
- Bij orders boven de 1.500 EUR, moet er eerst een offerte worden gestuurd naar de leveranciers die het desbetreffende product leveren.
- Wanneer alle offertes terug aangekomen zijn, selecteert de assistent hieruit één leverancier, maakt de bestelbon op, en verstuurt deze naar de leverancier. Teken Activity Diagram



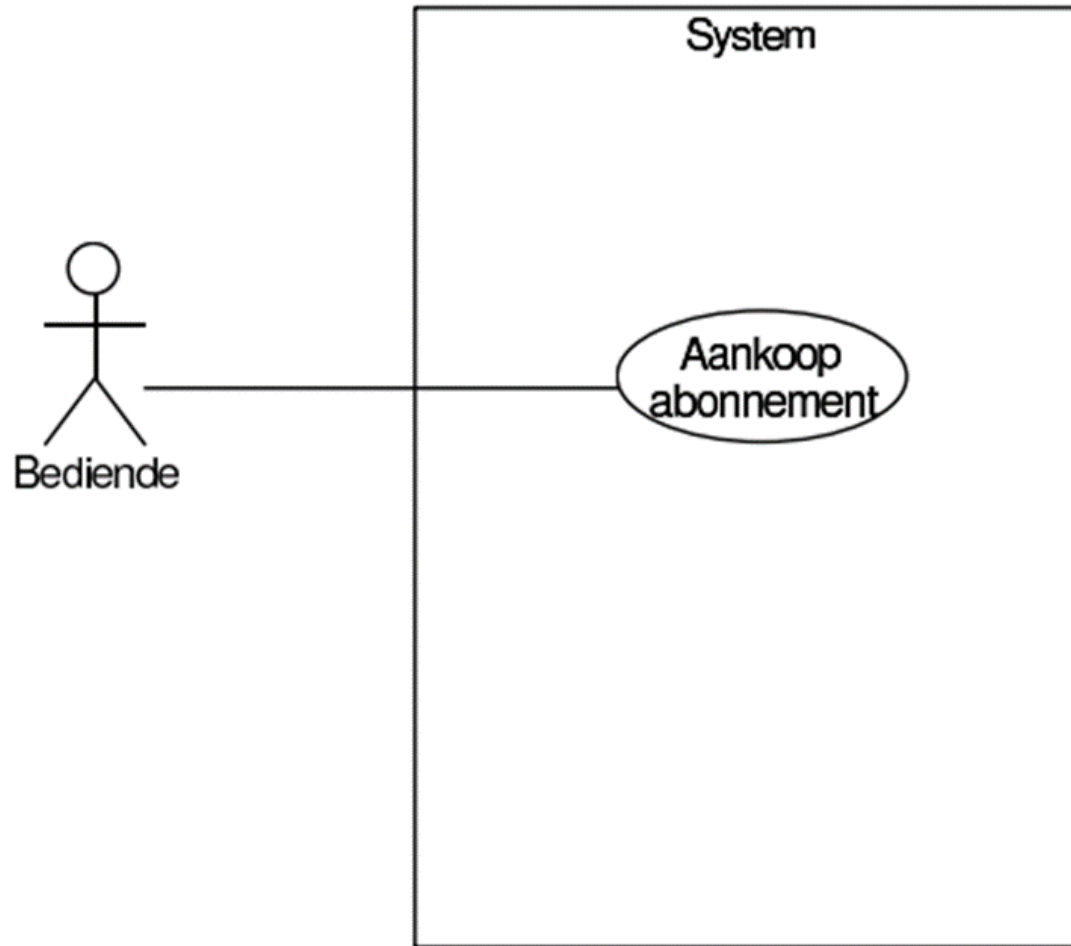
# Activity Diagram – exercises “De Lijn” (1)

- Een informatiesysteem van De Lijn schrijft bus- en tramabonnementen uit
- De klant komt aan het loket en geeft zijn persoonlijke informatie (o.a. naam, adres, telefoonnummer, geboortedatum) door aan de loketbediende die deze informatie ingeeft in het systeem.
- Het systeem zal zelf bepalen — op basis van de geboortedatum — welk type abonnement gepast is (jongeren, normaal of 60+) en wat de bijbehorende prijs is.
- Wanneer de betaling is geregistreerd, wordt het abonnement afgedrukt.
- Op het abonnement wordt de naam van de klant, zijn adres en geboortedatum, evenals het type abonnement afgedrukt
- Ga ervan uit dat het steeds gaat om een nieuwe klant, d.w.z. de persoonlijke informatie moet steeds worden ingegeven door de bediende



# Activity Diagram – exercises “De Lijn” (2)

- Stel het activity diagram op voor volgende use case



# Activity Diagram – exercises “De Lijn” (3)

- Stel het activity diagram op voor volgende use case

<b>Use Case:</b>	Aankoop abonnement	
<b>Triggering event:</b>	Een klant koopt aan het loket een nieuw abonnement.	
<b>Beschrijving:</b>	Een klant arriveert aan een loket van De Lijn en koopt bij een bediende een nieuw abonnement aan. De gegevens van de klant en het abonnement moeten opgeslagen worden in het systeem. Het type abonnement dat wordt aangekocht, is afhankelijk van de leeftijd van de klant.	
<b>Workflow:</b>	<i>Actor</i>	<i>Systeem</i>
	1. Ingeven van de persoonlijke gegevens van de klant.  2. Bezorg abonnement aan klant.	1a. Bewaren van de persoonlijke gegevens in het systeem. 1b. Aanmaak nieuw abonnement (incl. bepalen van type en prijs). 1c. Afdrukken van het abonnement.

# Activity Diagram – exercises “De Bib” (1)

- Een bibliotheek wil een nieuw informatiesysteem introduceren. Binnen het informatiesysteem wordt het begrip “boek” gebruikt voor boeken met eenzelfde ISBN-nummer. Het is echter mogelijk dat de bibliotheek over meerdere fysieke exemplaren van een bepaald boek beschikt.
- Elk exemplaar krijgt een uniek plaatskenmerk toegekend. Per boek wordt de titel en het ISBN-nummer bijgehouden.
- Elk boek in de bibliotheek wordt toegekend aan één (en slechts één) categorie. Voorbeelden van categorieën zijn jeugdboeken, thriller en biografie. Elke categorie heeft een naam en een alfanumeriek ID. Elke categorie kan 0, 1 of meerdere subcategorieën bevatten zodat een boomstructuur ontstaat

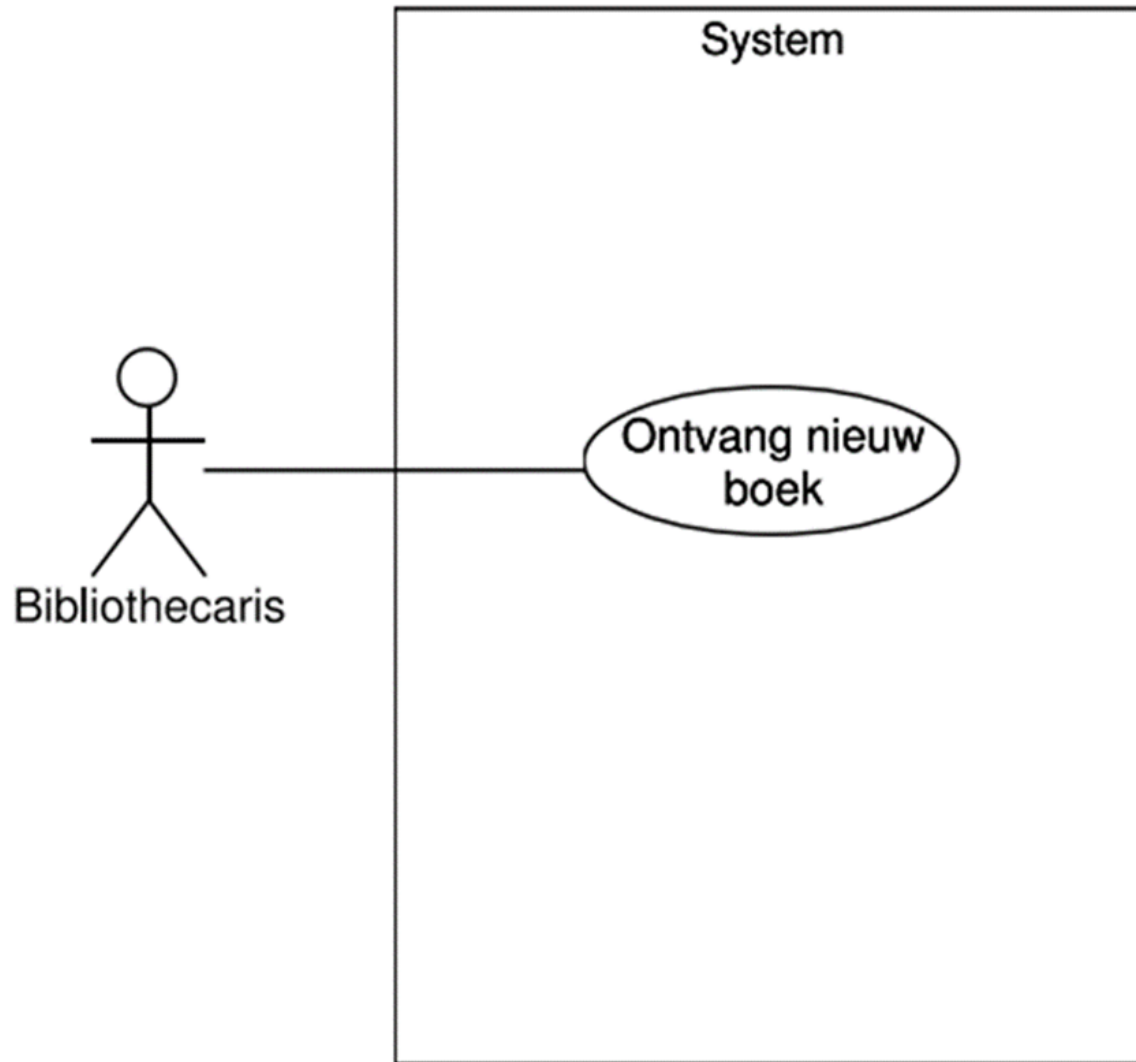
- Fictie
  - Jeugdboeken
  - Nederlandstalig
    - \* Thriller
    - \* Horror
  - Engelstalig
    - \* Thriller
    - \* Horror
- Biografie



## Activity Diagram – exercises “De Bib” (2)

- De bibliothecaris kan bestellingen plaatsen om nieuwe boeken aan te kopen. Voor elke bestelling wordt het ordernummer, de datum van bestelling en de datum van levering bijgehouden. Er wordt momenteel niet bijgehouden op welk boek een bepaalde bestelling betrekking heeft.
- Op het moment dat er een bestelling wordt geplaatst, zal de bibliothecaris de gegevens van het boek ingeven in het systeem. Dit betekent dat er bij het ontvangen van een bestelling enkel moet geregistreerd worden dat er een nieuw exemplaar van het boek werd ontvangen. Het kan dus voorkomen dat er gegevens van boeken in het systeem voorkomen, waarvan de bibliotheek (nog) geen exemplaren bezit.
- Stel het activity diagram op voor aangegeven use case (volgende slide)

# Activity Diagram – exercises “De Bib” (3)



# Activity Diagram – exercises “De Bib” (4)

<b>Use Case:</b>	Ontvang nieuw boek	
<b>Triggering event:</b>	Een nieuw aangekocht boek arriveert.	
<b>Beschrijving:</b>	De bibliothecaris moet registreren dat een bepaalde bestelling werd ontvangen, en dat er een nieuw exemplaar van een bepaald boek is toegevoegd aan het aanbod van de bibliotheek. De gegevens van boeken die werden besteld, zijn reeds aanwezig in het systeem.	
<b>Workflow:</b>	<i>Actor</i>	<i>Systeem</i>
	<ol style="list-style-type: none"> <li>1. Ontvangst van boek, inclusief ontvangstbewijs met vermelding van bestelcode.</li> <li>2. Aanpassen van datum van levering van bestelling.</li> <li>3. Registratie van een nieuw exemplaar van een bestaand boek a.d.h.v. het ISBN-nummer van het boek. Het nieuwe gekozen exemplaarnummer wordt eveneens ingegeven.</li> <li>4. Fysiek boekexemplaar voorzien van sticker met plaatskenmerk.</li> <li>5. Plaatsen van exemplaar in rekken.</li> </ol>	<ol style="list-style-type: none"> <li>2. Opzoeken van bestelling in database en updaten van leveringsdatum.</li> <li>3. Opzoeken van boekgegevens (ISBN-nummer) in het systeem, en toevoegen van nieuw exemplaar.</li> </ol>

# Questions & answers

