DESIGN: COMPONENT DESIGN. THE MODEL COMPONENT

SU:E15:L10

LECTURE PLAN: UPDATE

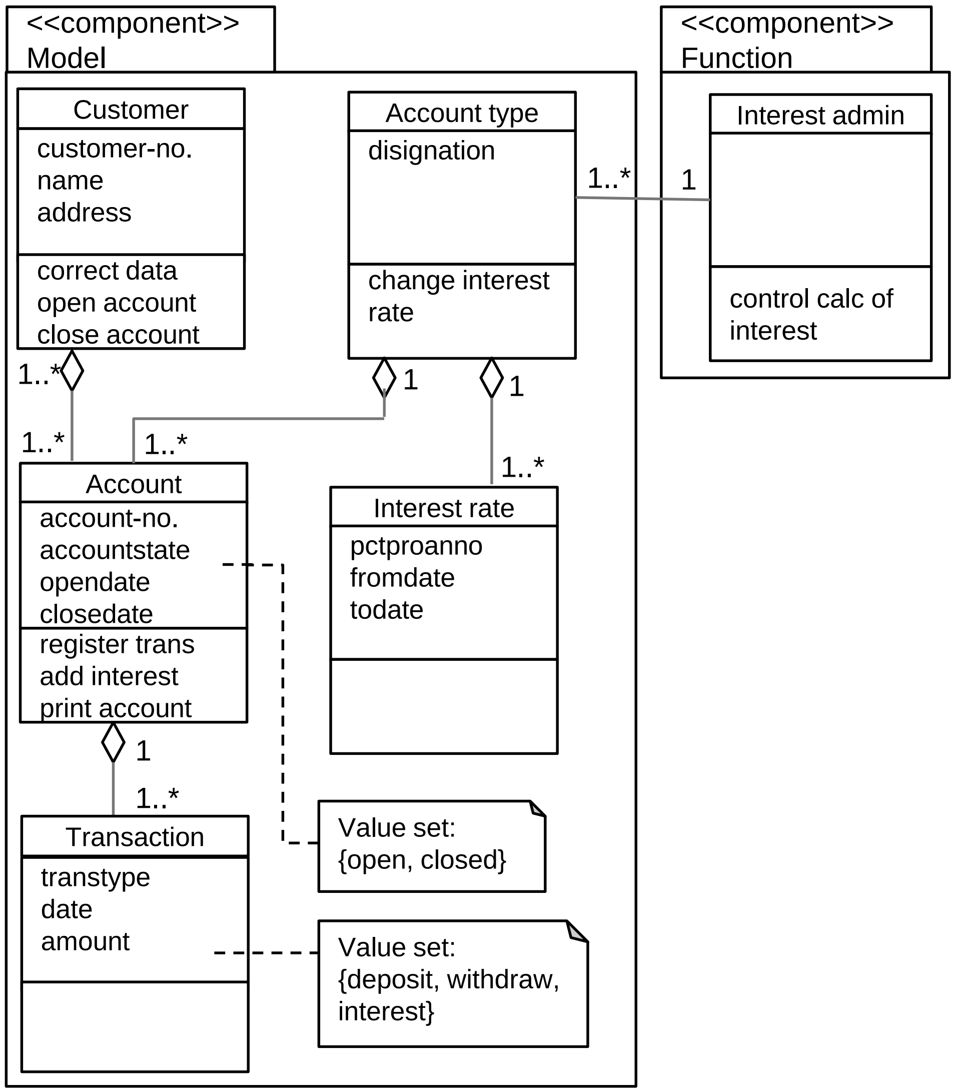
1. Introduction
2. Construction, evolution and prototyping (Exercises first)
3. Collaboration with users and system choice (Exercises first)
4. Modeling – classes
5. Modeling – structure
6. Modeling – behavior (Exercises first)
7. Modeling – use
8. Modeling – functions
9. Design – architecture, criteria, components
10. **Design – model component (today)**
11. **Design – function component, connecting components (2. November)**
12. **Guest Lecture, Per Stilling, Netcompany (4. November)**
    - At 1230, Auditorium Frederik Bajers Vej 7H

2

COMPONENT DESIGN

COMPONENT DESIGN

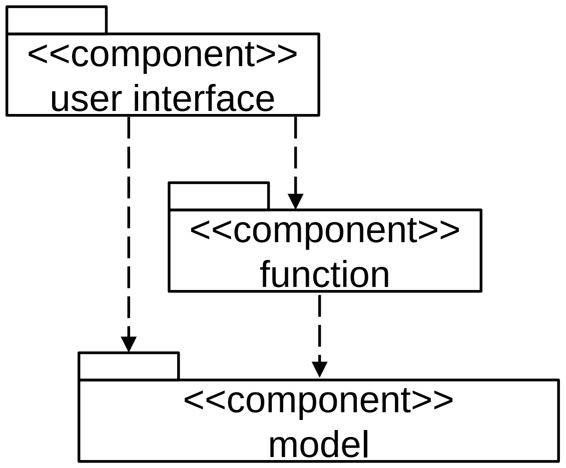
- **Component details**
- **Connections between components**
- **Designing the architecture is an iterative process**
  - Revise the division of components
  - Influences the process architecture

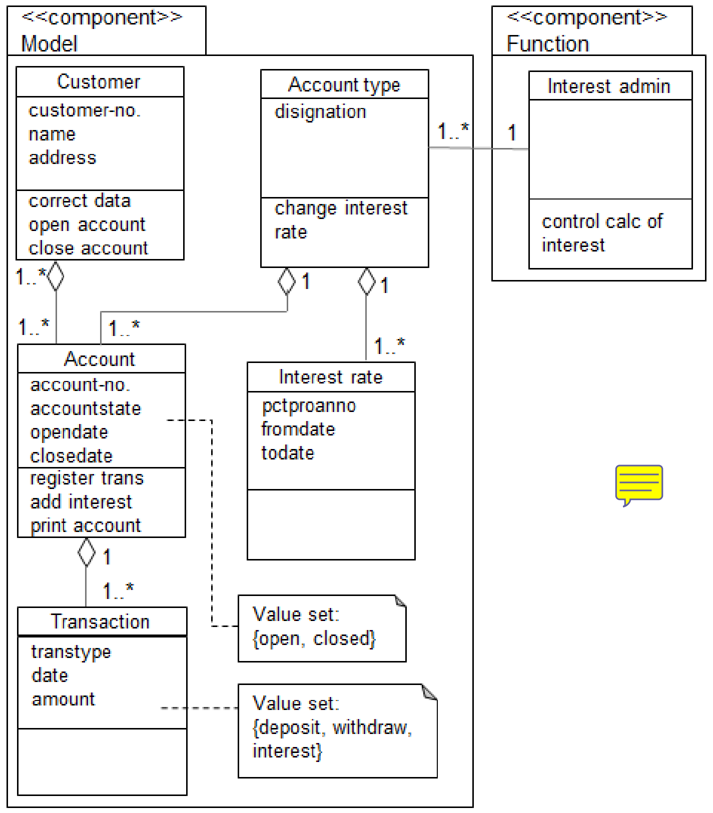


OVERVIEW OF 'COMPONENTS'

Purpose	• To determine an implementation of requirements within an architectural framework.
Concepts	• **Component**: A collection of program parts that constitutes a whole and has well-defined responsibilities. • **Connection**: The implementation of a dependency relation.
Principles	• Respect the component architecture. • Adapt component designs to the technical possibilities.
Results	• A description of the system's components.

FROM ARCHITECTURE TO COMPONENTS

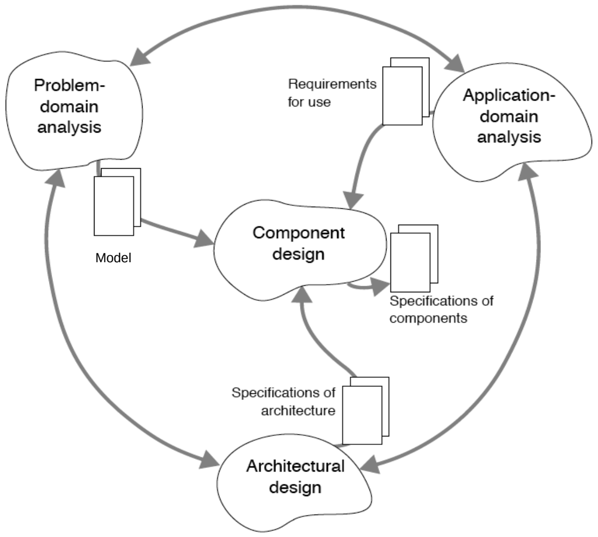




Principles:

- Respect the component architecture
- Adapt component designs to the technical possibilities

ACTIVITIES IN 'COMPONENT DESIGN'


Problem-domain analysis
Requirements for use
Application-domain analysis
Model
Component design
Specifications of components
Specifications of architecture
Architectural design

Model component

- How is the model represented as classes in the system?
- Model component and attribute

Function component

- How are the functions implemented?
- Function component and operation

Connect

- How are components connected?
- Component and connection

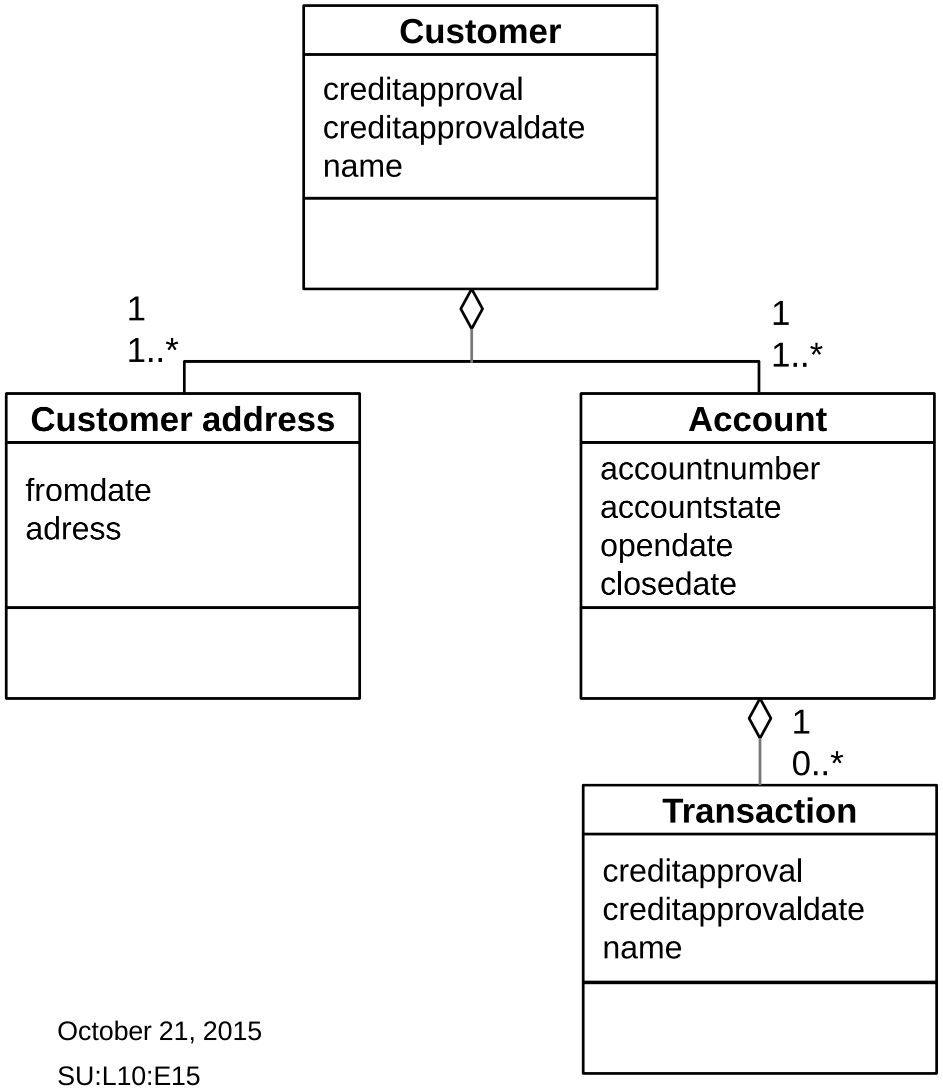
2

MODEL COMPONENT

OVERVIEW OF 'MODEL COMPONENT'

Purpose	• To represent a model of a problem domain
Concepts	• **Model component**: A part of the system that implements the problem-domain model. • **Attribute**: A descriptive property of a class or an event.
Principles	• Represent events and classes, structures and attributes. • Choose the simplest representation of events.
Results	• A class diagram of the model component.

RESULT OF MODEL COMPONENT



- **Starting point: the class diagram from the problem domain analysis**
- **Extended to handle behavior**
  - Adding new classes
  - Adding attributes
  - Adding and revising structures

ACTIVITIES IN ‘MODEL COMPONENT’

Represent private events

- Sequence and selection
- Iteration

Represent common events

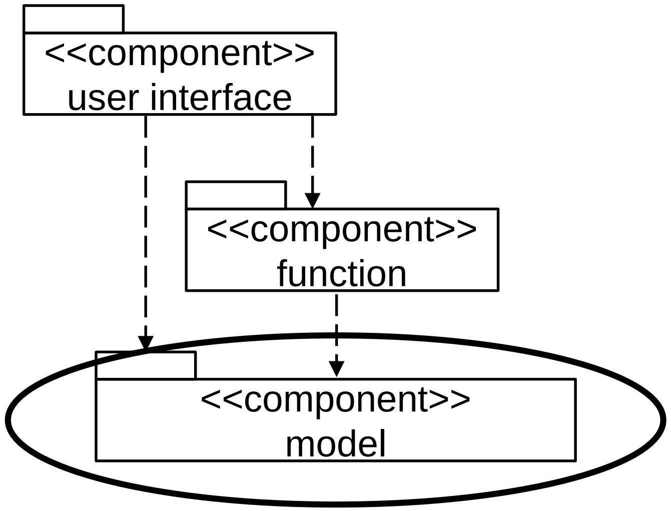
- Choose between alternatives

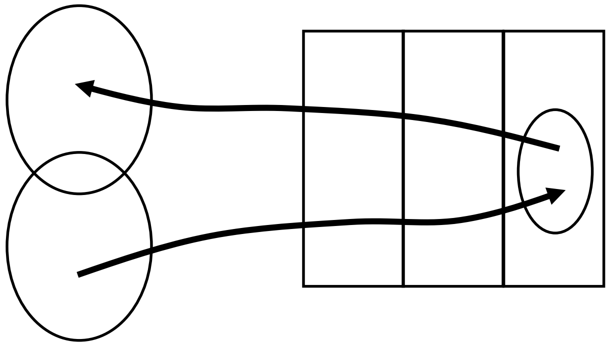
Restructure classes

- Generalization
- Association
- Embedded iterations

FROM OVERVIEW TO SPECIFICATIONS OF DETAILS

- **Component:**
  A collection of program parts that constitutes a whole and has well-defined responsibilities.
- **Responsibility of the model component:**
  maintain an updated representation of the problem domain.

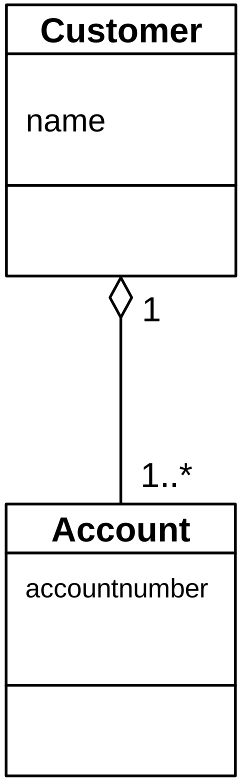


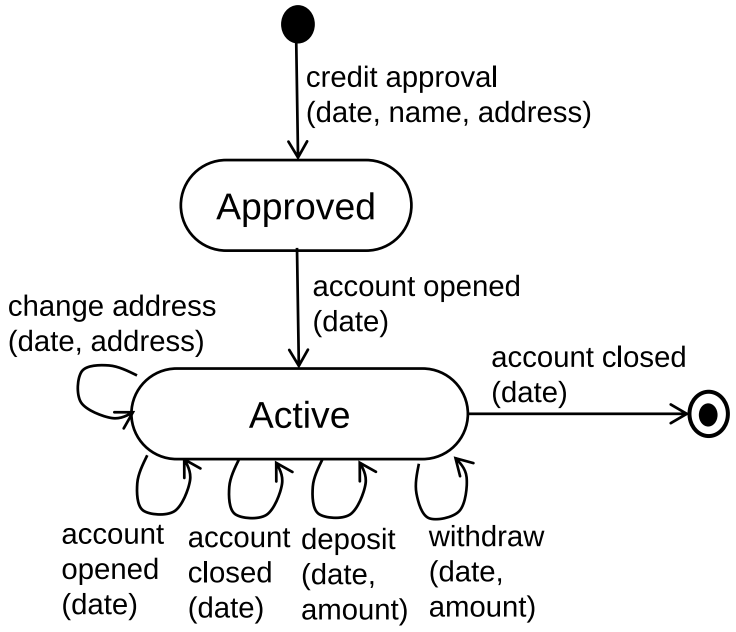


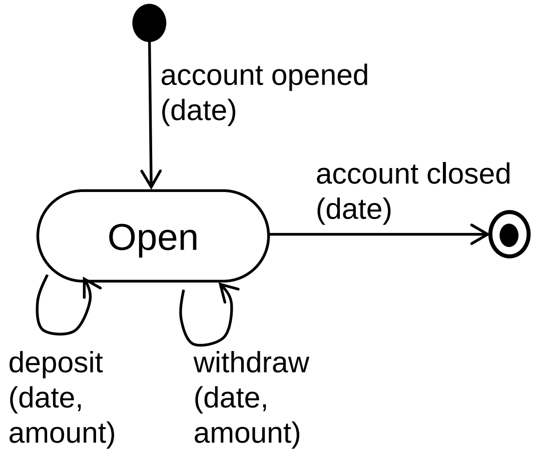
ANALYSIS MODEL FOR BANK SYSTEM

- Class diagram
- Event table

Event	Customer	Account
Credit approval	+	
Change adress	*	
Account opened	*	+
Account closed	*	+
Deposit	*	*
Withdraw	*	*







REPRESENT PRIVATE EVENTS

events that involve only one problem-domain object.

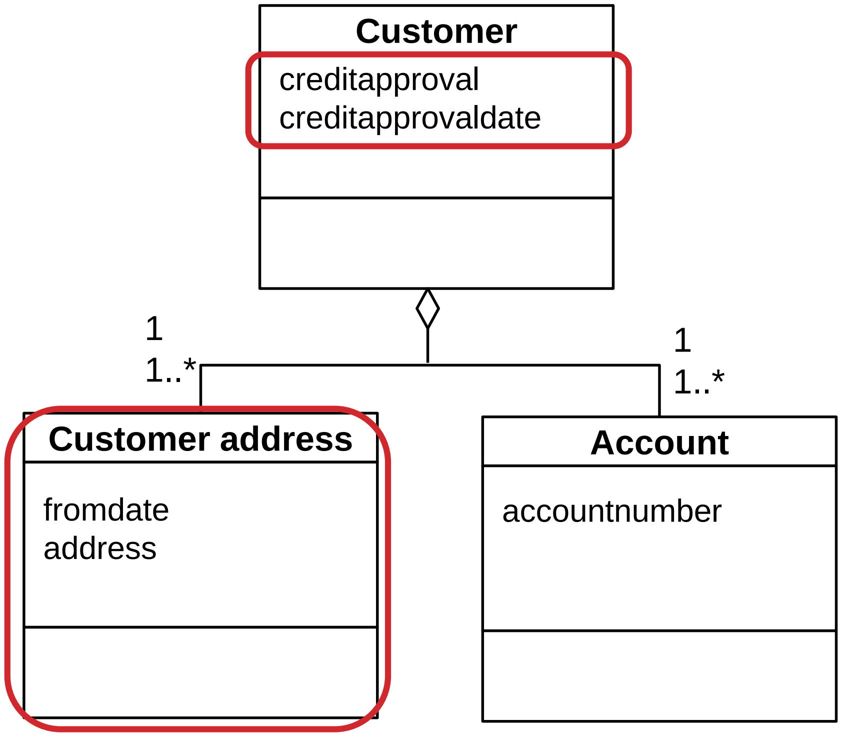
- **Sequence and selection**
  - Represent these events as an attribute in the class described in the state chart diagram.
  - The system assigns a value to the attribute when the event occurs.
  - Integrate the attributes of the event in the class.
- **Iteration**
  - Represent these events as a new class, connect it to the class described in the state chart diagram with an aggregation structure.
  - The system generates a new object of the class each time the event occurs
  - Integrate the attributes of the event in the class.

REPRESENT PRIVATE EVENTS

- The event ‘credit approval’ is private to the class customer and is a sequence in the state chart diagram for the class
  - Represented as an attribute

- The event ‘change adress’ is private to the class Customer and is an iteration in the state chart diagram for the class
  - Represented as a new class

Eventç	Customer	Account
Credit approval	+	
Change adress	*	
Account opened	*	+
Account closed	*	+
Deposit	*	*
Withdraw	*	*



REPRESENT COMMON EVENTS

events that involve two or more problem-domain objects.

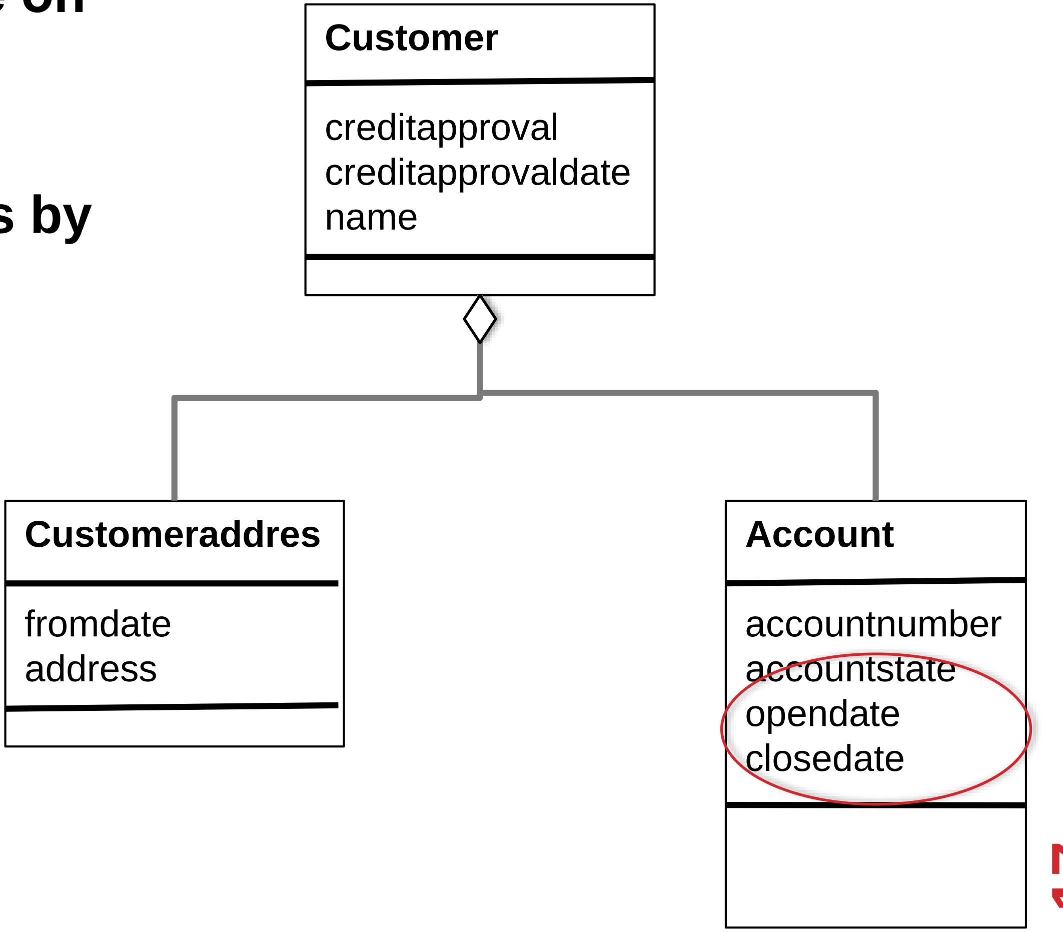
- **Common events**
  - Represent the event in relation to one of the objects
    - Consider adding structural connections to give the other objects access to the relevant attributes.
  - Represent common events in the way that offers the simplest structure.

  - If the event figures differently in the state chart diagrams, it is represented in connection to the class, which gives the simplest representation.
  - If the event figures in the same way in the state chart diagrams, you have to consider the possible representations

REPRESENT COMMON EVENTS: CHOOSING A SIMPLE ALTERNATIVE

Eventç	Customer	Account
Credit approval	+	
Change adress	*	
Account opened	*	+
Account closed	*	+
Deposit	*	*
Withdraw	*	*

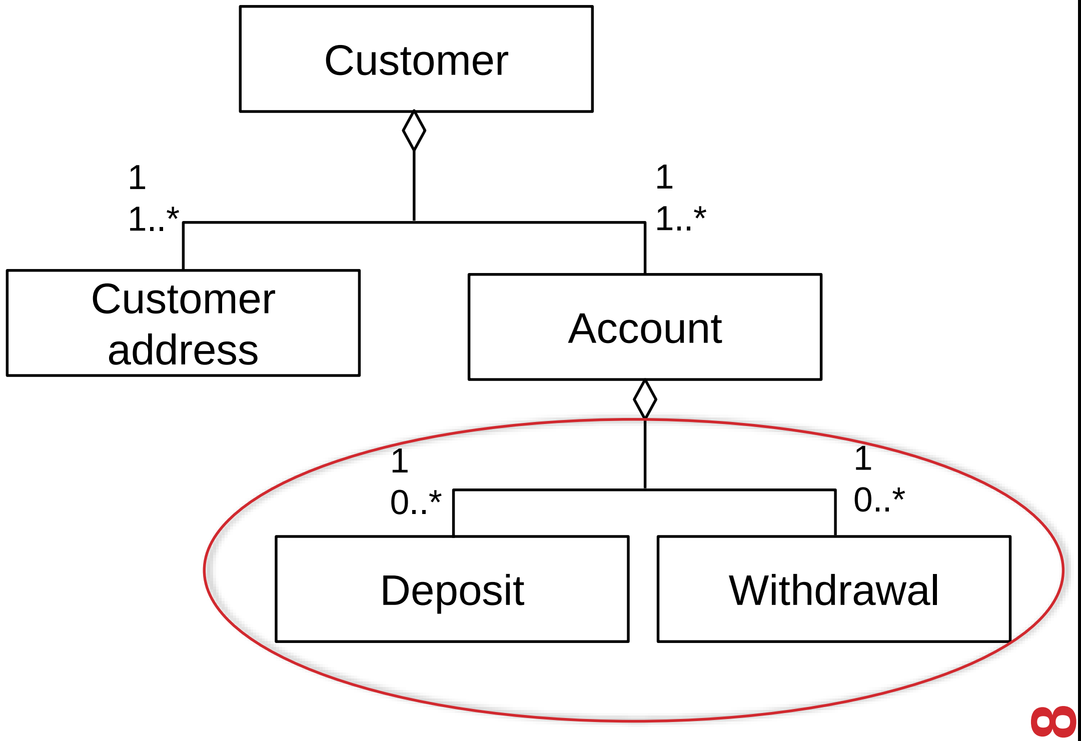
- The events ‘account opened’ and ‘account closed’ are interative on class Customer and in a sequence on class Account
- The simplest representation is by adding attributes to class Account



REPRESENTATION OF COMMON EVENTS: ITERATIONS SOLUTION A

Event	Customer	Account
Credit approval	+	
Change adress	*	
Account opened	*	+
Account closed	*	+
Deposit	*	*
Withdraw	*	*

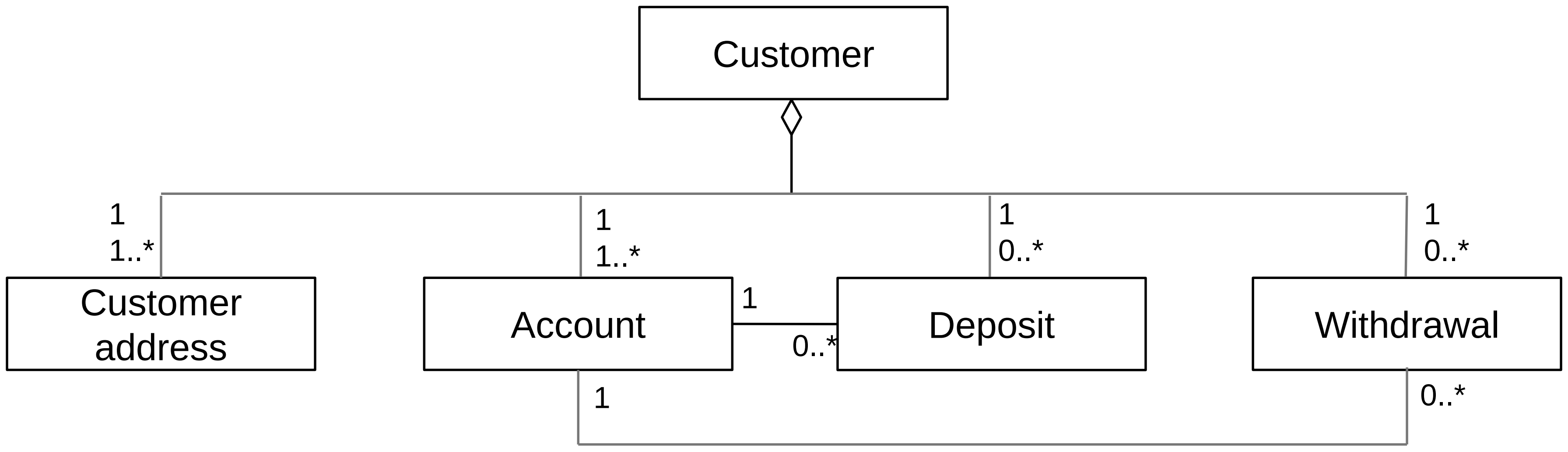
- The events 'withdraw' and 'deposit' are iterations on two classes
- The events can be represented as new classes under Account



REPRESENTATION OF COMMON EVENTS: SOLUTION B

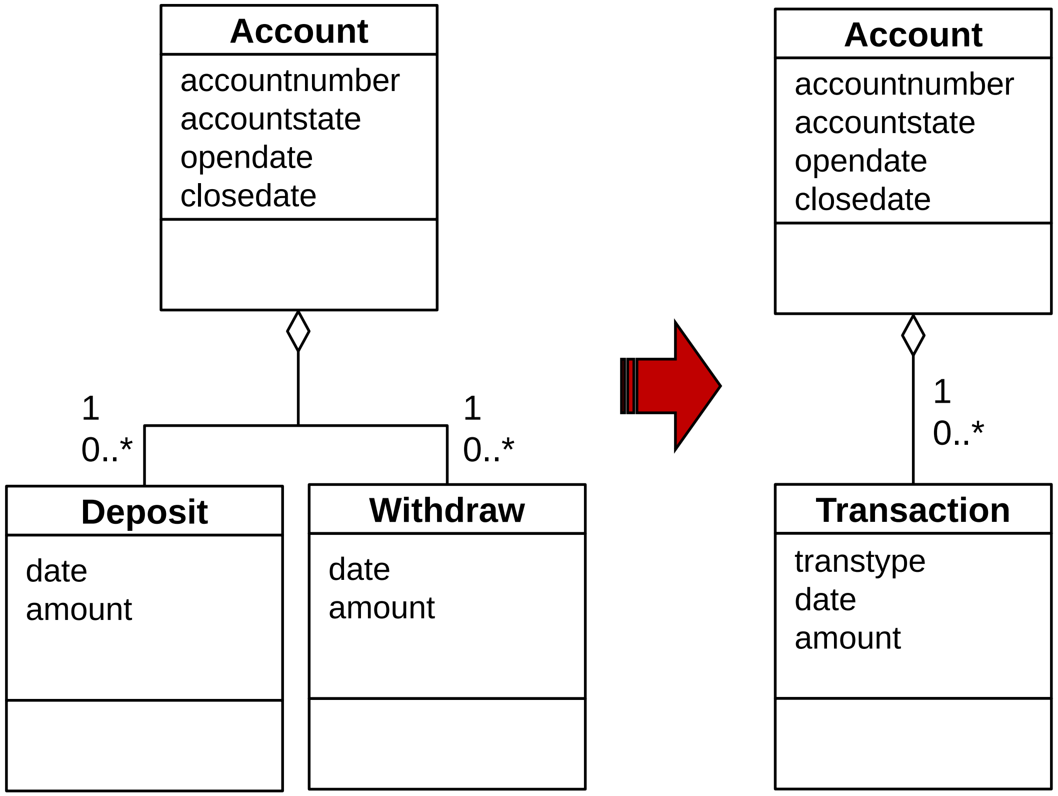
- Alternatively: the events can be represented as new classes under the customer class

- Gives a complex structure (two associations across)
- We would therefore choose solution A

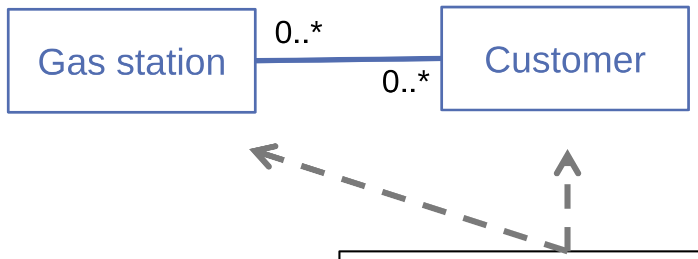


RESTRUCTURE CLASSES (1)

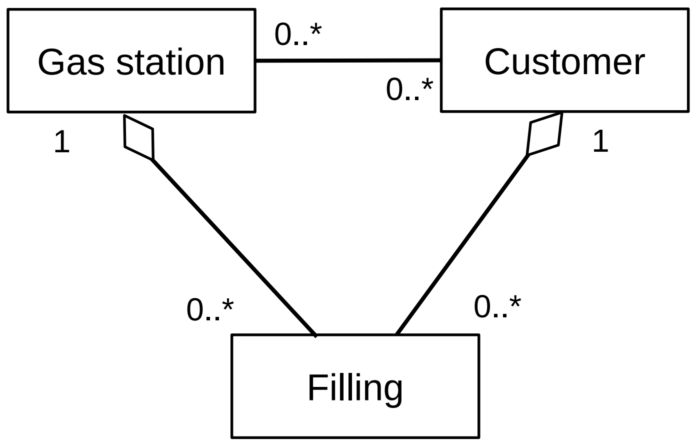
- **The revised class diagram represents the information from the state chart diagrams.**
- **The class diagram can often be restructured and simplified without any loss of information:**
  - Generalization
  - Association
  - Embedded iterations

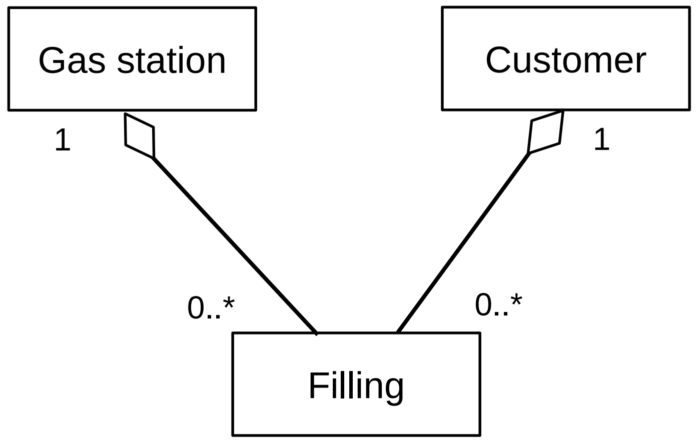


RESTRUCTURE CLASSES (2)



The event ‘Fill’ is iterative on both classes

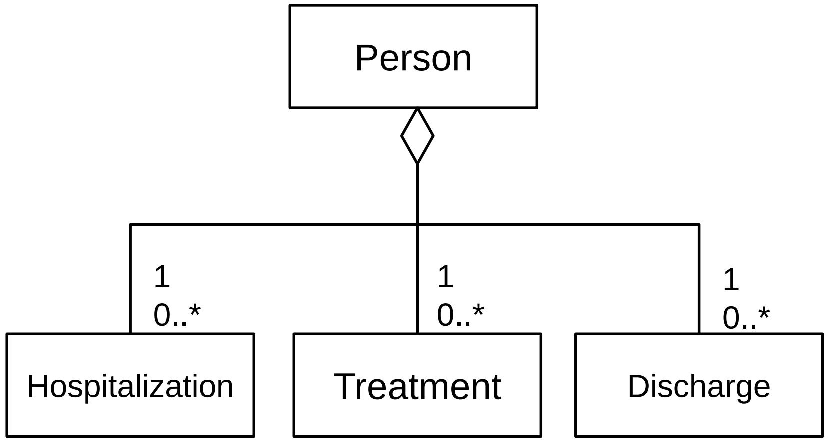




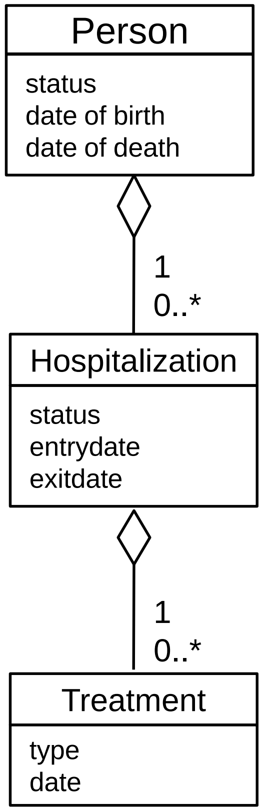
October 21, 2015

SU:L10:E15

RESTRUCTURE CLASSES (3)

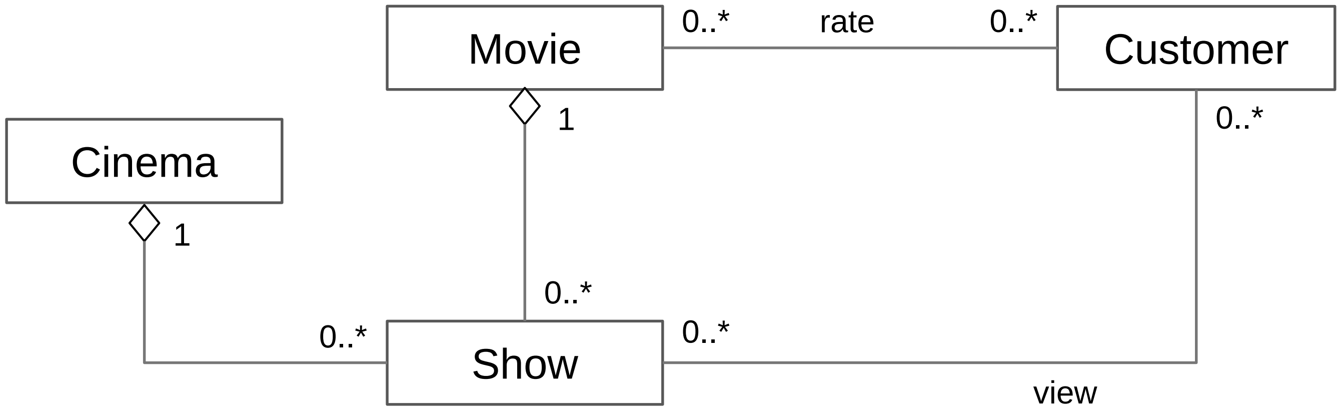


Does not represent the association between "Treatment", "Discharge" and a specific "Hospitalization"



GROUP ASSIGNMENT

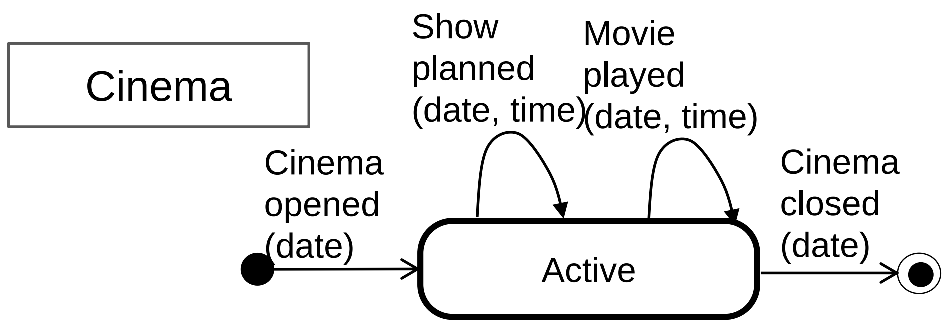
Complete the activity “Model component” for the cinema example:

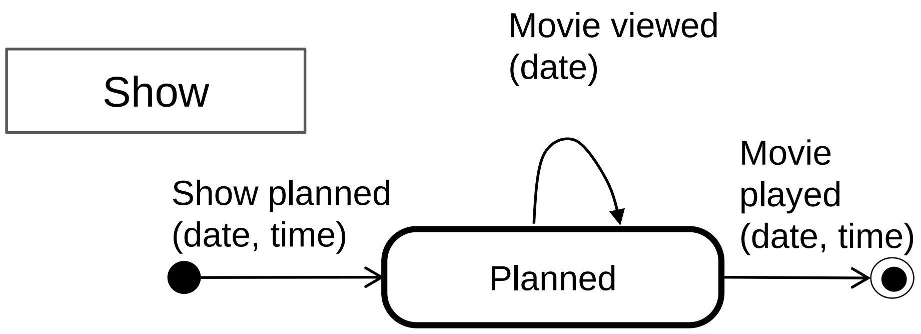


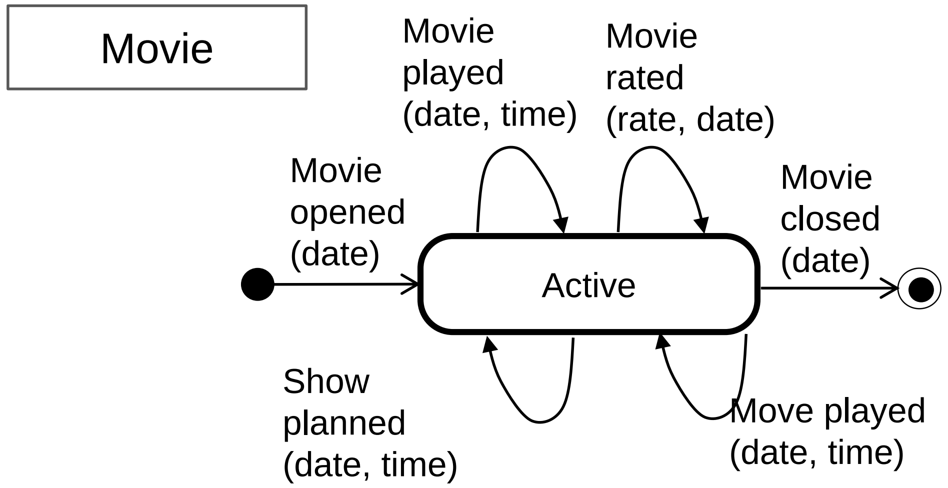
GROUP ASSIGNMENT

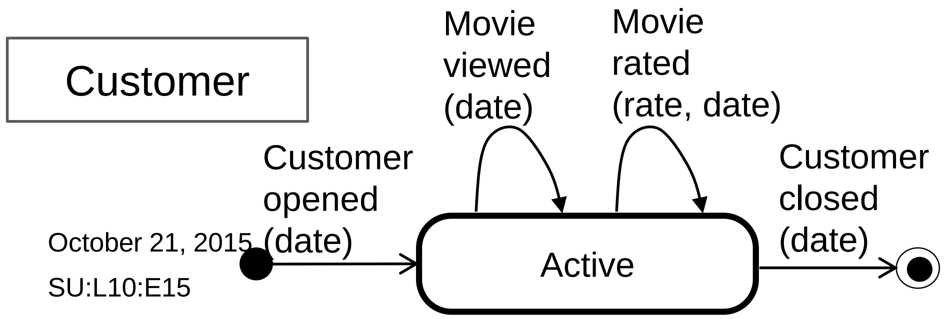
Represent private events
- Selection/sequence: attribute
- Iteration: class

Represent common events
- Choose among alternatives









	Movie	Customer	Show	Cinema
Movie opened	+			
Movie closed	+			
Movie played	*		+	*
Show planned	*		+	*
Customer closed		+		
Customer opened		+		
Movie viewed		*	*	
Movie rated	*	*		
Cinema opened				+
Cinema closed				+

OVERVIEW OF ‘MODEL COMPONENT’

Purpose	• To represent a model of a problem domain
Concepts	• **Model component**: A part of the system that implements the problem-domain model. • **Attribute**: A descriptive property of a class or an event.
Principles	• Represent events and classes, structures and attributes. • Choose the simplest representation of events.
Results	• A class diagram of the model component.