Systems Development

Lecture 10: Function Component

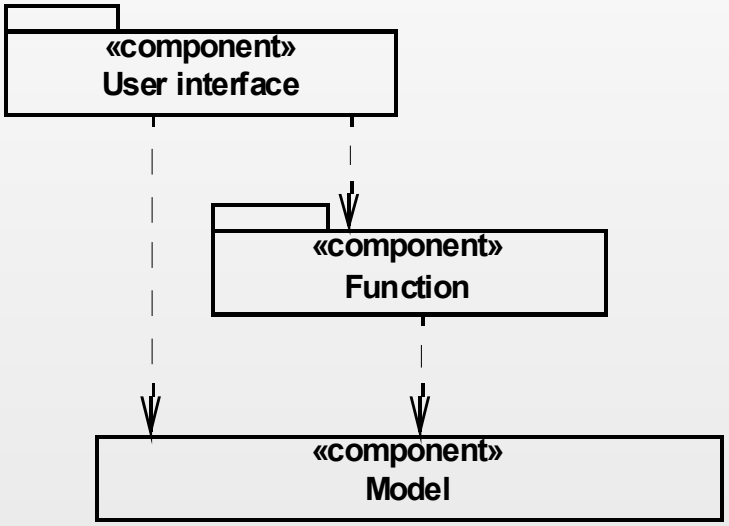
Contents

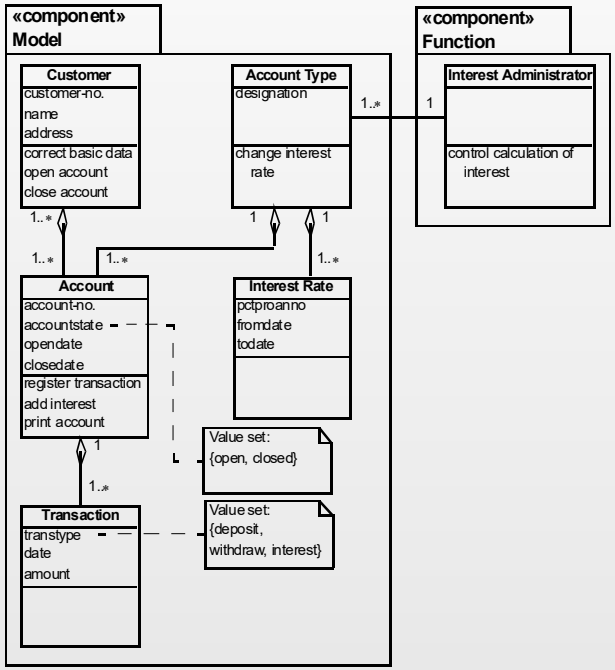
- Summary of last lecture
- The Function Component activity
- Solution to Written exam 2018-06

Contents

- Summary of last lecture
  - Component design
  - The Model Component activity
- The Function Component activity
- Solution to Written exam 2018-06

Key Concepts: From Architecture to Components



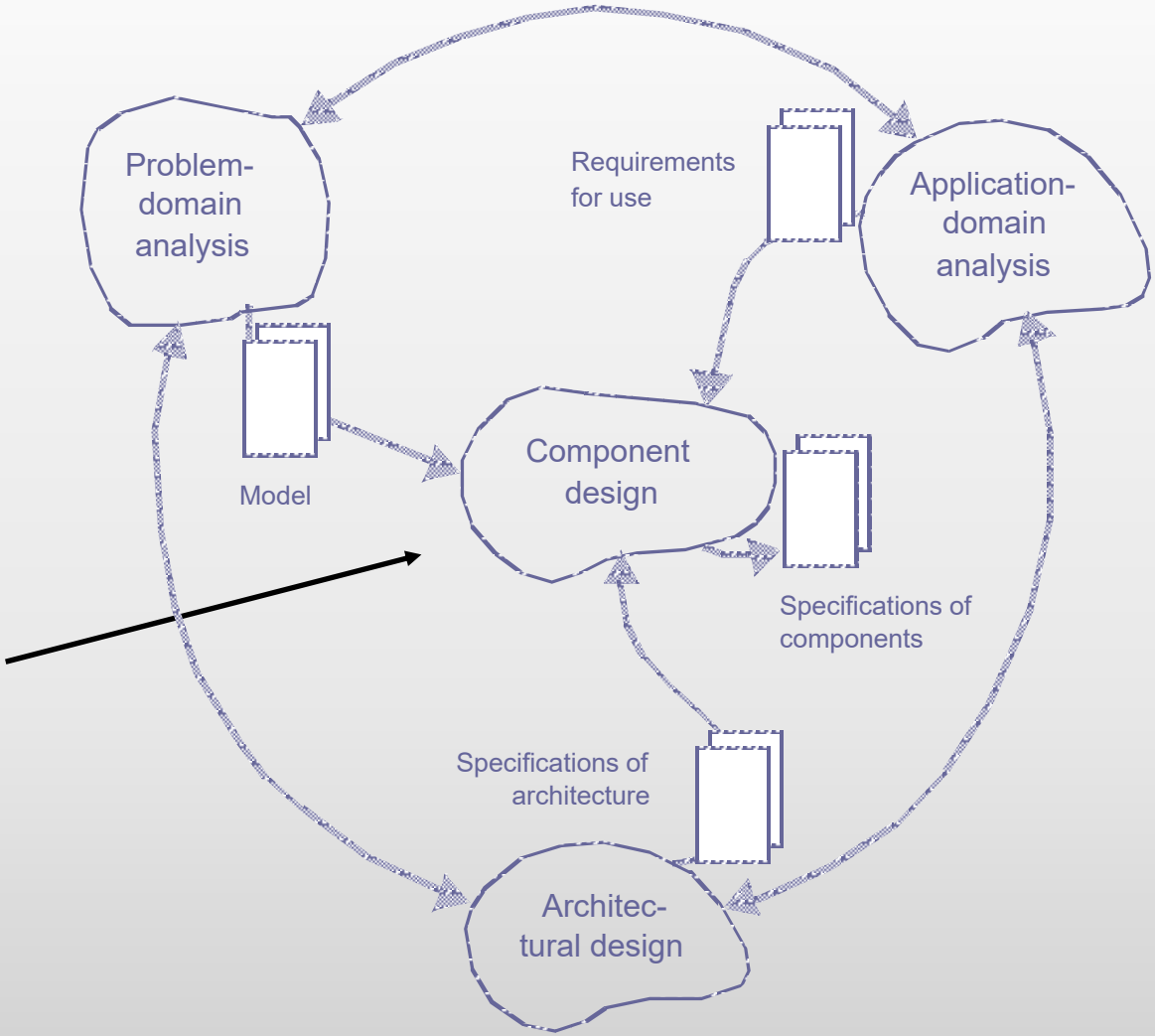


Principles:

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

Component Design: Activities

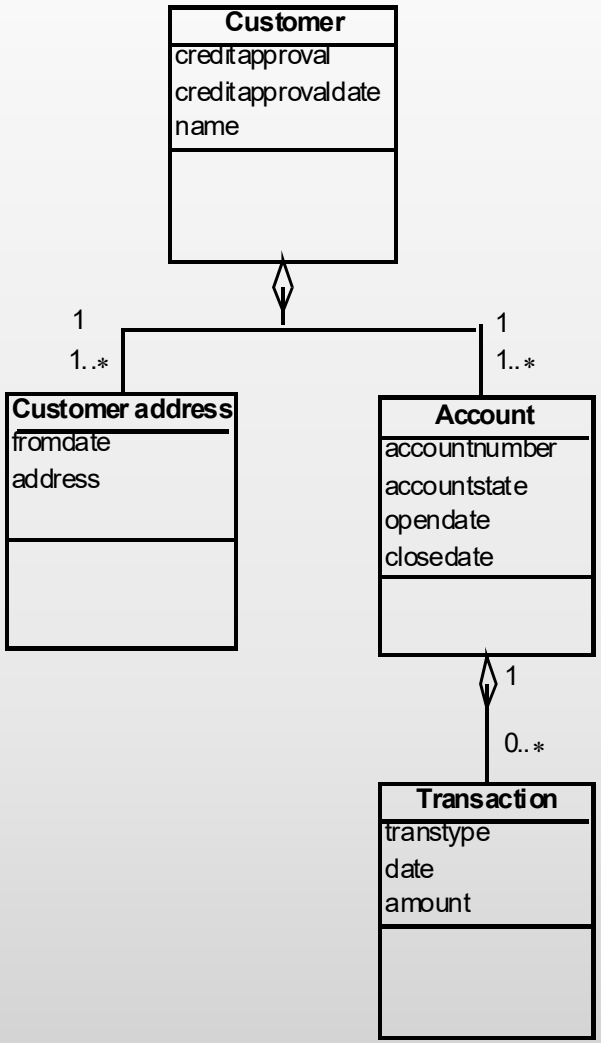
Model component
Function component
Connect components
... more components



Component Design: Summary

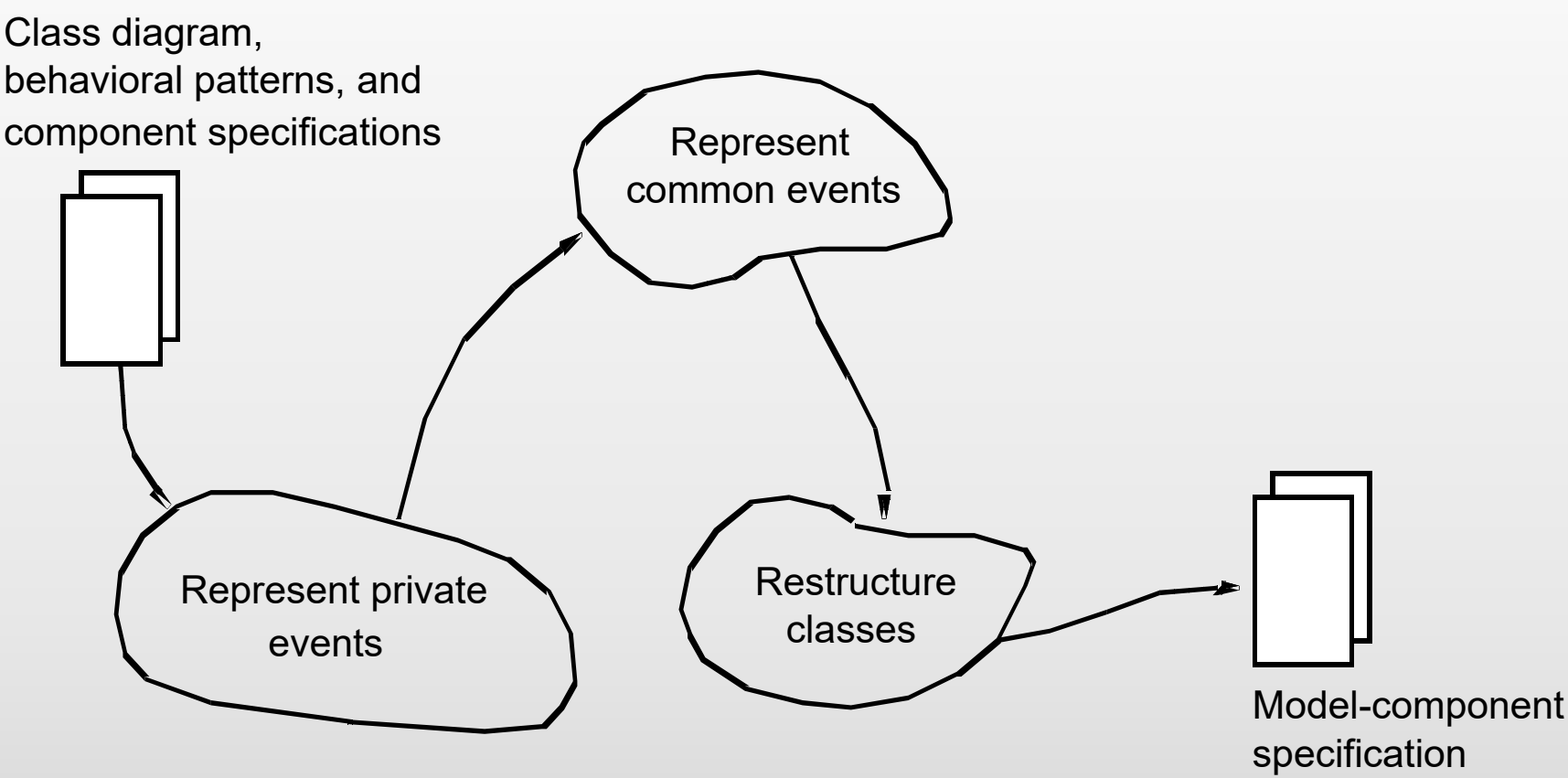
Purpose	• To determine an implementation of requirements within an architectural framework.
Concepts	• Component: a collection of program parts that constitutes a whole and has well-defined responsibilities. • Connection: the implementation of a dependency relation.
Principles	• Respect the component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

Model Component: Results



- Point of departure in the class diagram from the problem domain analysis
- Extended with representation of behavior described in the statechart diagrams

Model Component: Activities

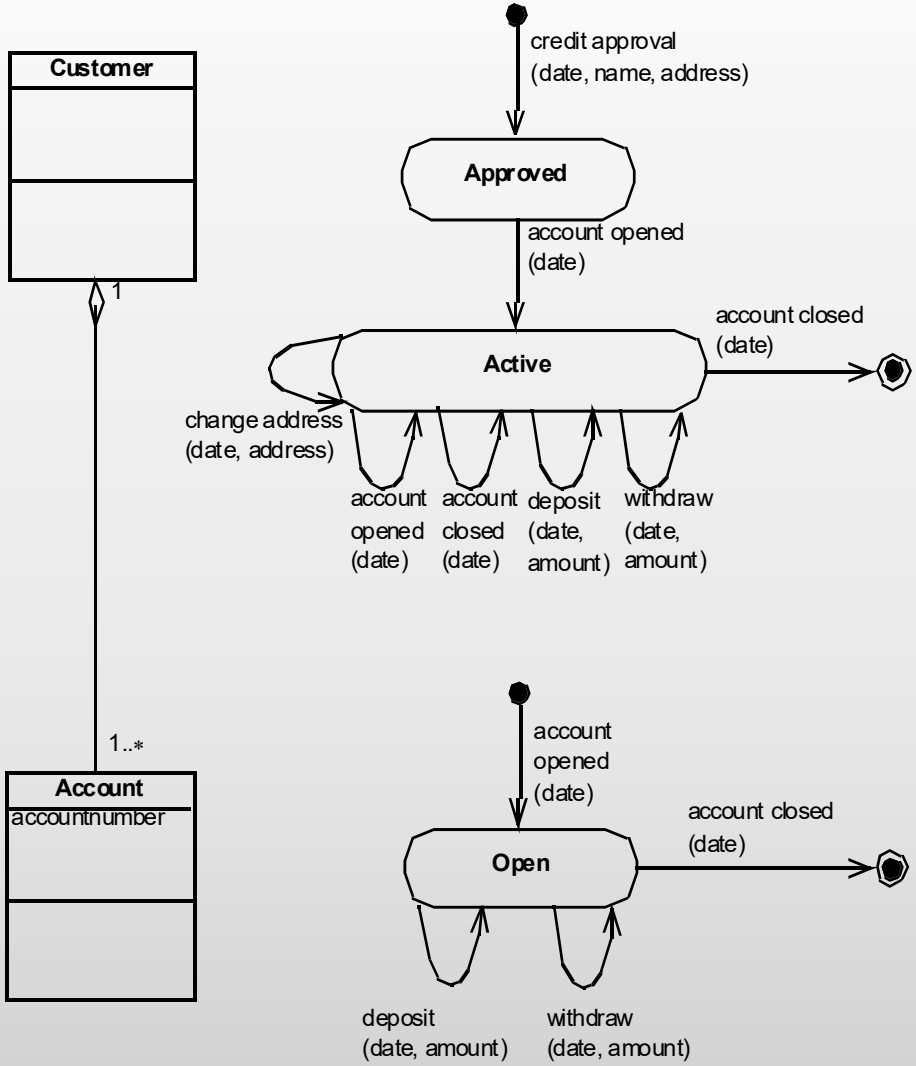


Example: Bank System

Analysis model:

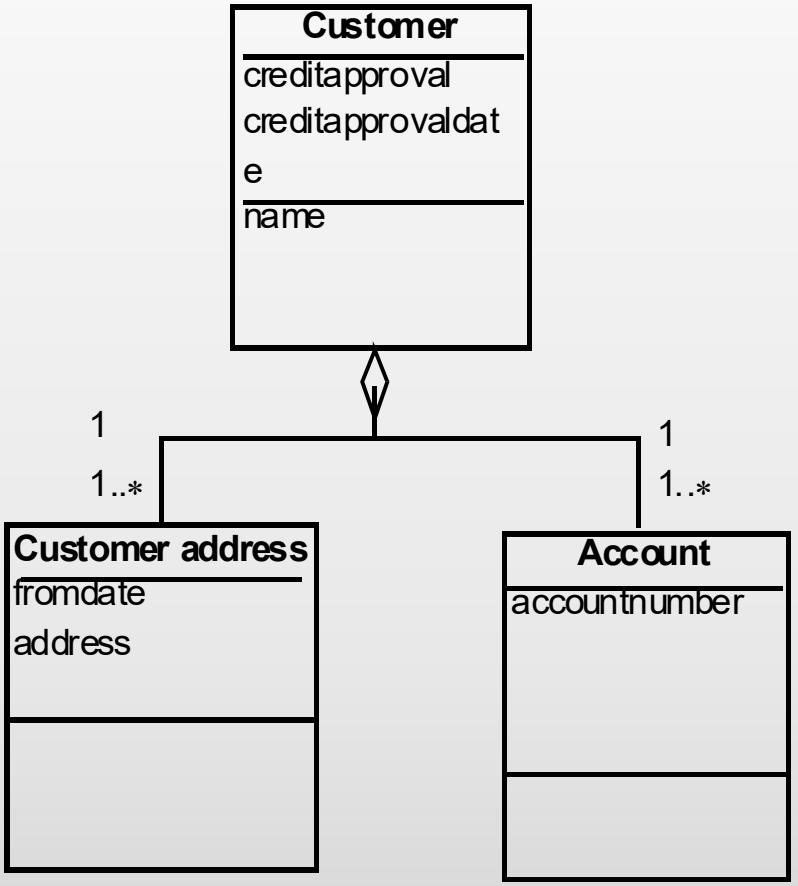
- Class diagram
- Statechart diagrams
- Event table

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



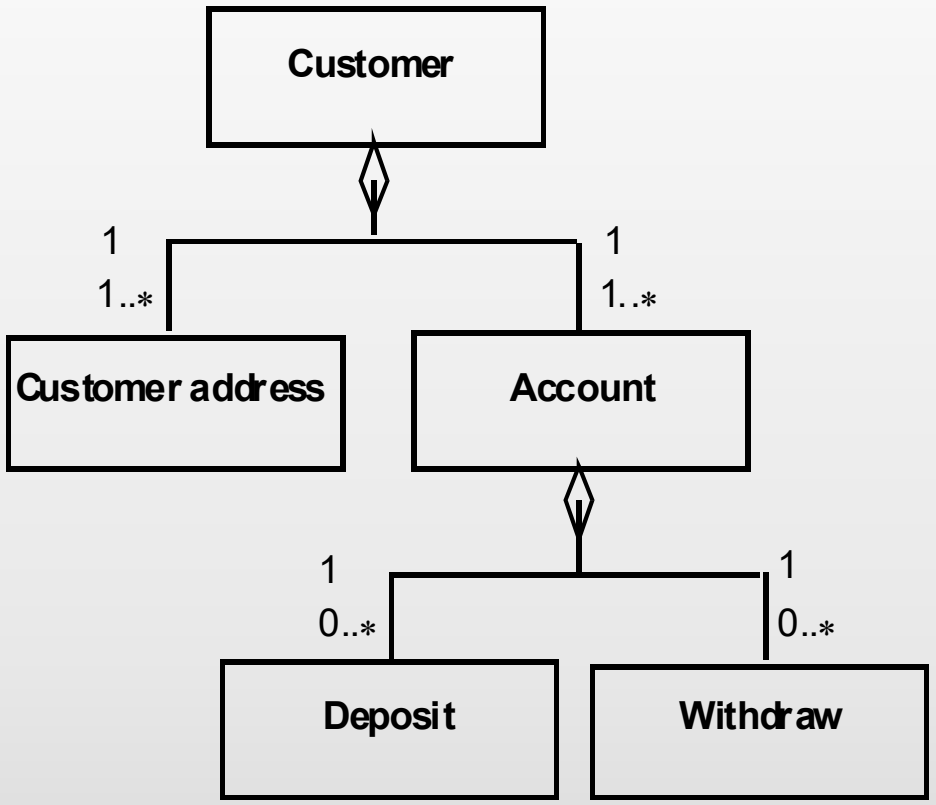
Represent Private Events

- The event ‘change address’ is private to the class Customer. It is an iteration in the statechart diagram of the class
- Represent this event as a new class
- The event ‘credit approval’ is private to the class Customer. It is part of a sequence in the statechart diagram
- Represent this event as an attribute



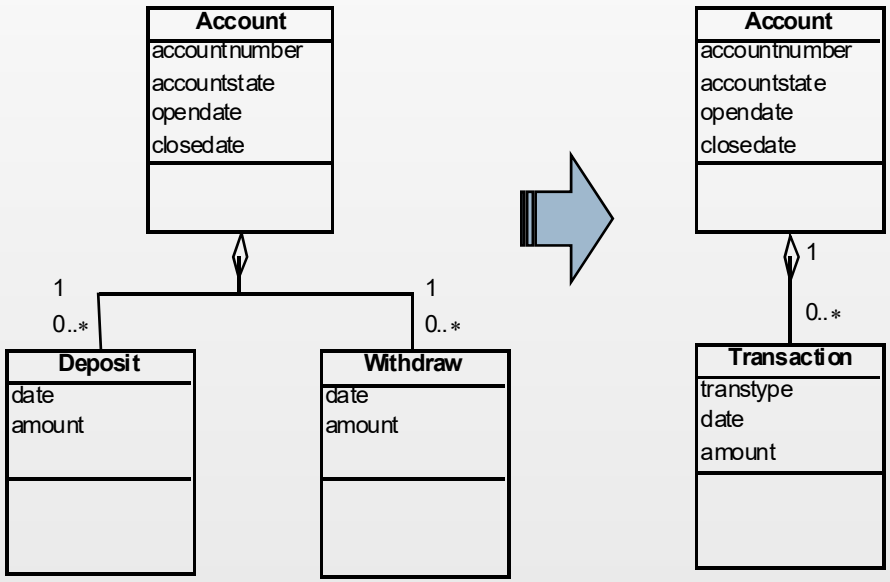
Represent Common Events: Solution A

- The events ‘deposit’ and ‘withdraw’ are iterations in both the Customer and Account classes
- One option is to represent these events as new classes under Account



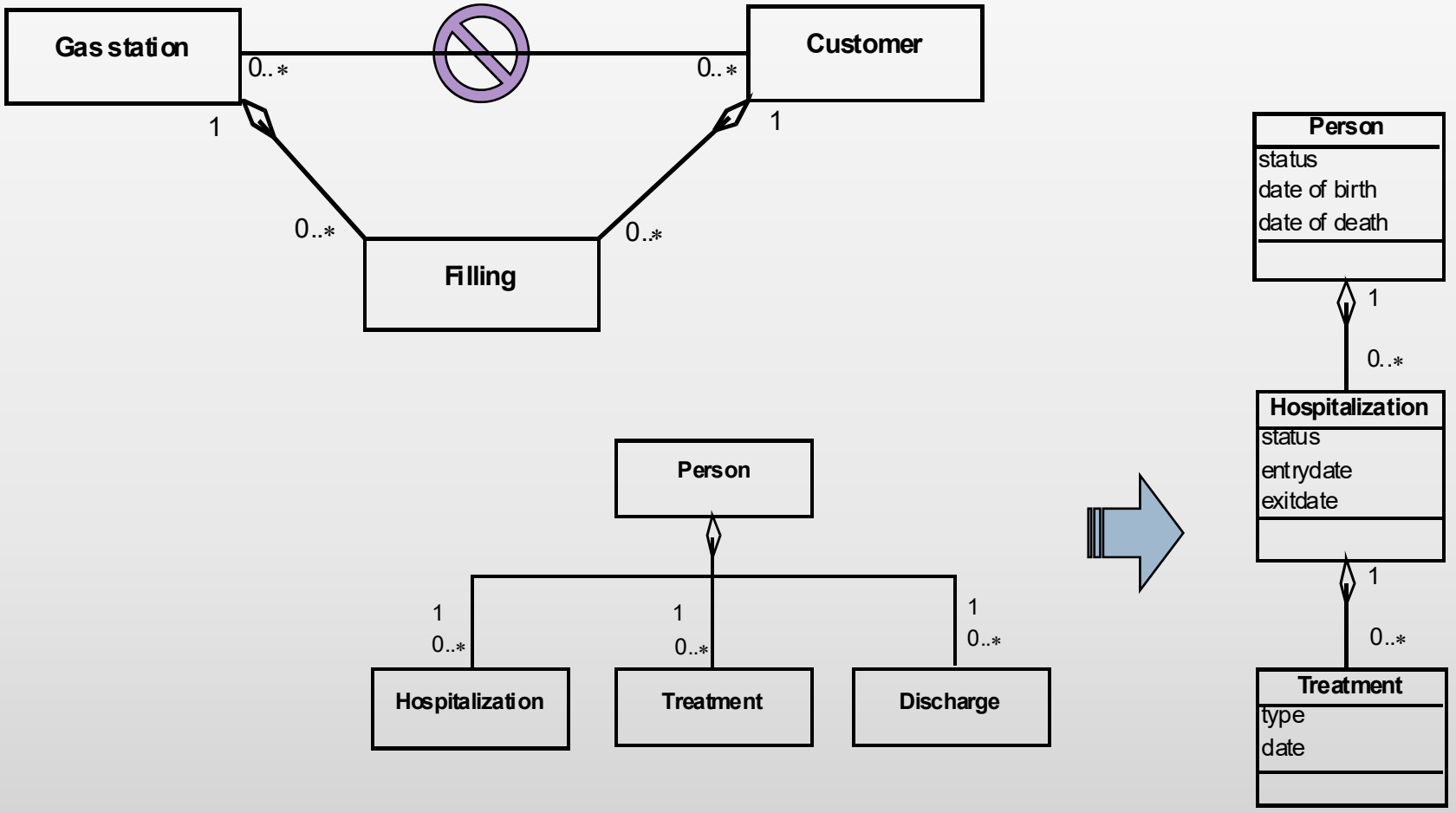
Restructure Classes (1)

- The revised class diagram represents the same information as the original class diagram and the statechart diagrams
- The class diagram can often be simplified without loss of information:
  - Generalization
  - Association
  - Embedded iterations



Restructure Classes (2)

Unnecessary association



Model Component: Summary

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

Quiz 9

Quiz 9

Average

3.42 (of 5.00) of 50 finished attempts (of 159)

Best result (0.67-1.00)

3 (0.92) When designing the model component for private events
5 (0.86) Given the event table; which of the three solutions is/are correct model components?
2 (0.77) Which are activities in component design

Middle result (0.34-0.66)

4 (0.49) Given the event table and class diagram; which of these statements are correct?
1 (0.39) What is design?

Worst result (0.00-0.33)

None

Question 1 and 2

What is design?

Select one or more:

- [x] a. Determination of a phenomenon in the IT-system
- [ ] b. Determination of the events an object performs or experiences
- [x] c. Determining how to get access to an object
- [ ] d. Identification of objects
- [x] e. Determining the operations an object can carry out and make available to other objects in the system
- [ ] f. Determination of a phenomenon in the context of the IT-system

Which are activities in component design:

Select one or more:

- [x] a. design of the model component
- [ ] b. designing the component architecture
- [ ] c. determining the design criteria
- [x] d. design of the connection between components
- [ ] e. designing the process architecture
- [x] f. design of the function component

Question 3 and 4

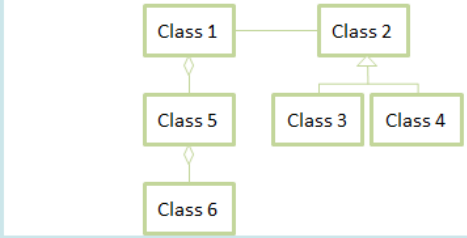
When designing the model component for private events:

Select one or more:

- ☑ a. for an iteration, create a new class
- ☐ b. for a selection, create a new class
- ☐ c. for a sequence, create a new class
- ☐ d. for an iteration, create an attribute
- ☑ e. for a sequence, create an attribute
- ☑ f. for a selection, create an attribute

Given the event table and class diagram; which of these statements are correct?

	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6
Event 1	+	*				
Event 2		+	*	*		
Event 3	*					
Event 4	*				+	
Event 5					*	*



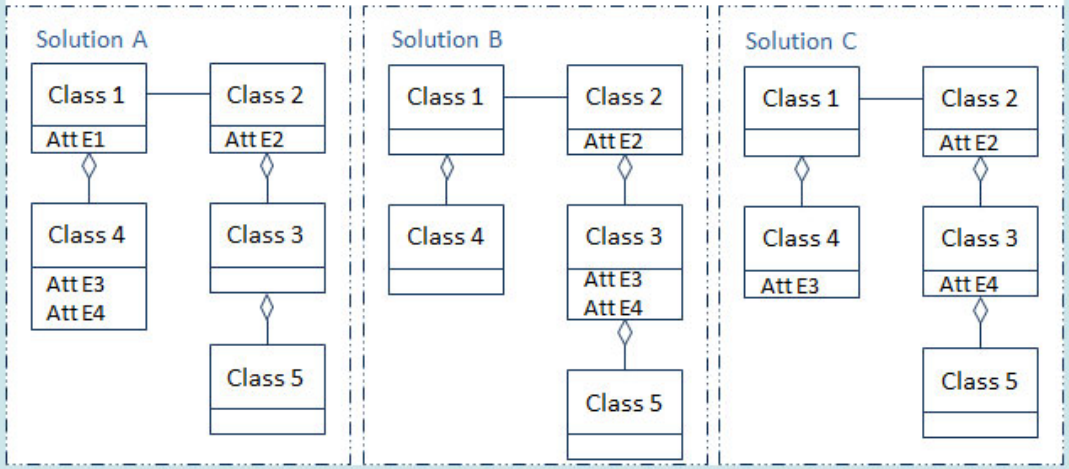
Select one or more:

- ☐ a. To represent event 1 a new class aggregated to class 1 must be created
- ☑ b. To represent event 2 a new attribute must be created on class 2
- ☐ c. To represent event 2 a new attribute must be created on class 3 or 4
- ☑ d. To represent event 3 a new class aggregated to class 1 must be created
- ☐ e. To represent event 4 a new class aggregated to class 1 must be created
- ☑ f. To represent event 4 an attribute must be created on class 5
- ☑ g. To represent event 5 a new class must either be aggregated to class 5 or class 6
- ☐ h. To represent event 5 two new classes each aggregated to class 5 and 6 must be created

Question 5

Given the event table; which of the three solutions is/are correct model components?

	Class 1	Class 2	Class 3
Event 1	*		
Event 2		+	*
Event 3		*	+
Event 4	*		+
Event 5		*	*



Select one or more:

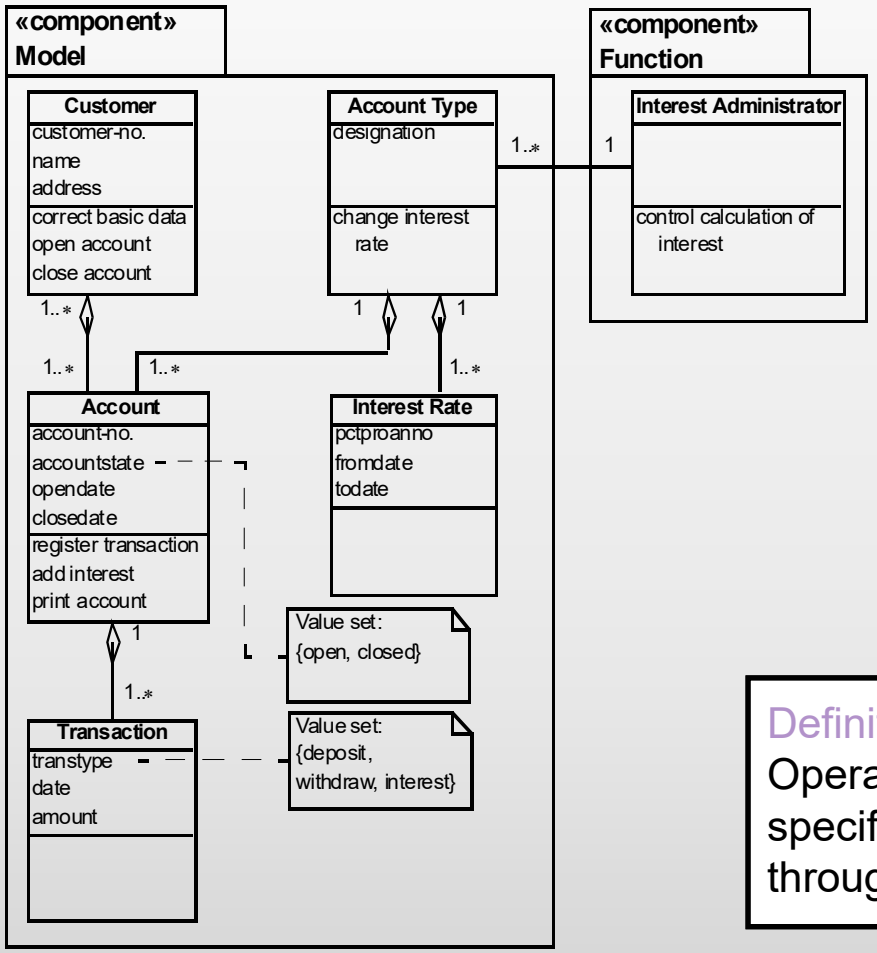
- ☐ a. Solution A
- ☑ b. Solution B
- ☐ c. Solution C

- Remember also a state attribute in each of the "original" classes
- This attribute will have values corresponding to the states in the behavioral patterns (statechart diagrams) from the problem domain analysis

Contents

- Summary of last lecture
- The Function Component activity
  - Results
  - Key concepts
  - Activities
- Solution to Written exam 2018-06

Function Component: Results



- Continue on the class diagram from design of the model component
- Extend with operations realizing requirements to functions from the analysis of the application domain

Definition:
Operation: A process property specified in a class and activated through the class' object

Secondary Result: Specification

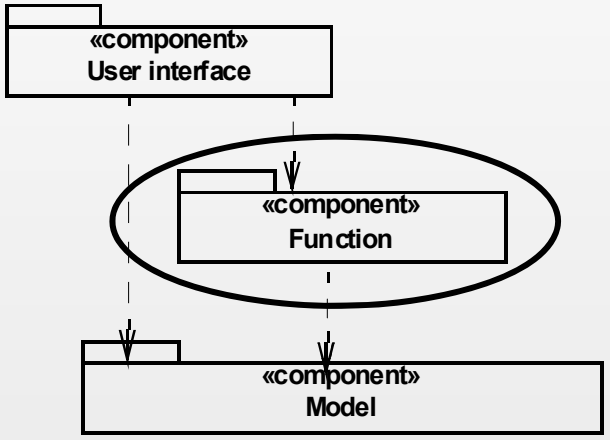
Name	Register transaction	
Category	_ Active x Passive	x Update _ Read _ Compute _ Signal
Purpose	Establishes a new transaction for a specific account.	
Input data	account-no., transtype, date, amount.	
Conditions	An object of the class Account, with the given account number, exists. The attribute accountstate in this object has the value Open.	
Effect	A new object of the class Transaction is established with input data assigned to the attributes. This object is connected to the relevant Account object.	
Algorithm		
Data structures		
Placement	Account.	
Involved objects	Account, Transaction.	
Triggering events	amount deposited, amount withdrawn.	

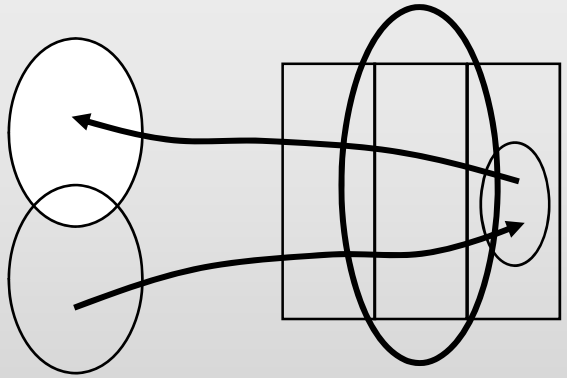
Function Component: Responsibility

Component:
A collection of program parts that constitutes a whole and has well-defined responsibilities

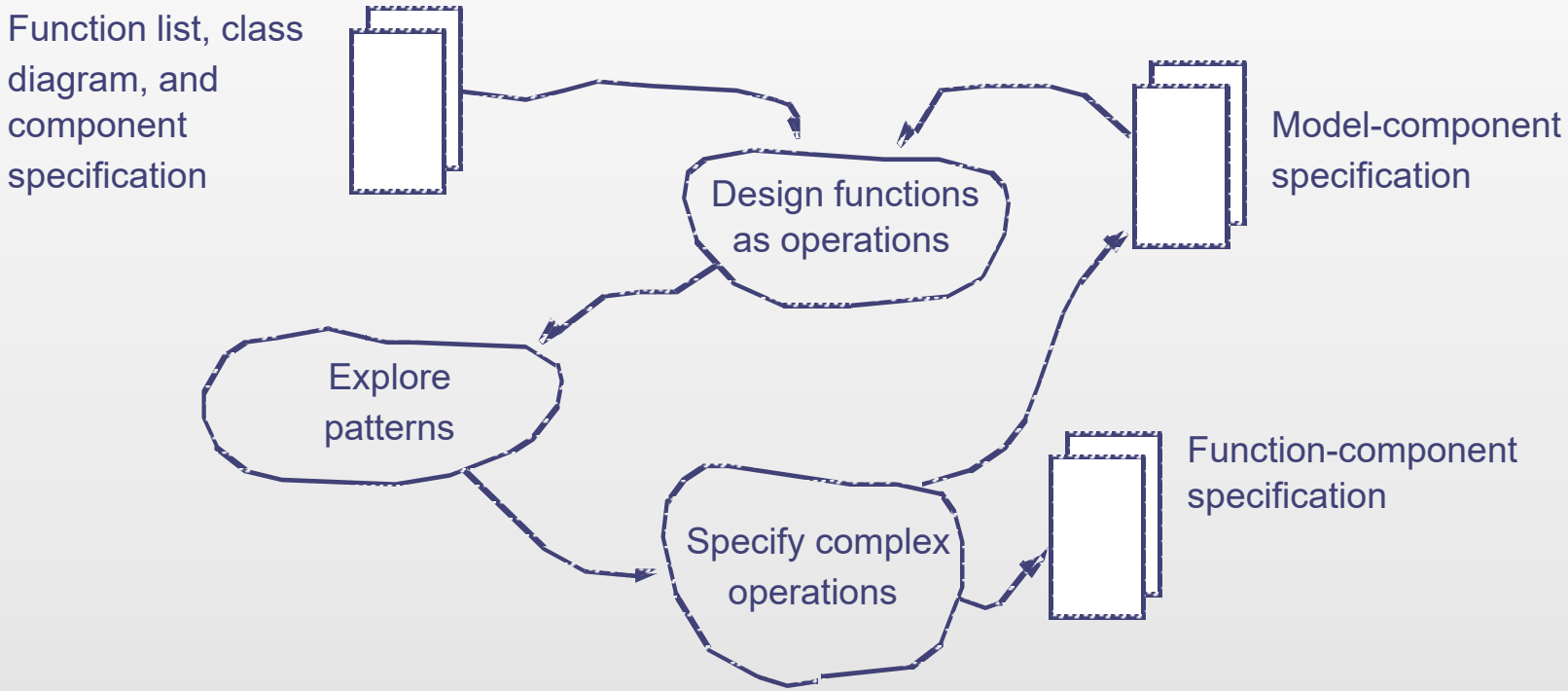
Responsibility of the function component:
Make the model component available as a ressource to actors

… and handle updates from the problem domain





Function Component: Activities

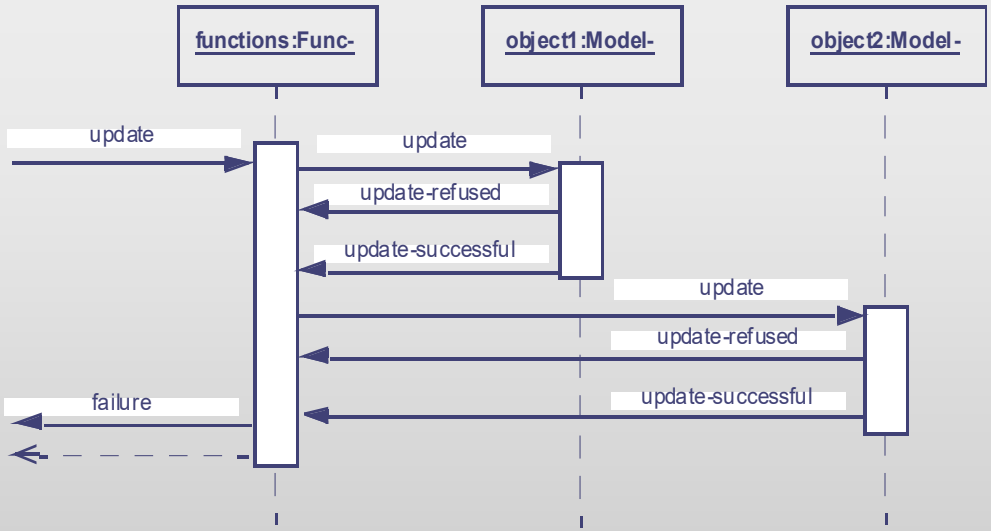


Design Functions as Operations

Function type	*Central questions*
Common to all types	• How should the function be implemented as operations in different classes? • How is the main operation activated and what input data does it use? • Which objects and connections are involved in performing the operations, and how are they identified? • What is the feedback from the main operation?
Update	• How can you determine if the update is legal?
Read	• Which attributes and connections should be read, and how are these found?
Compute	• Which algorithm should be carried out?
Signal	• Which rules apply to the signaling?

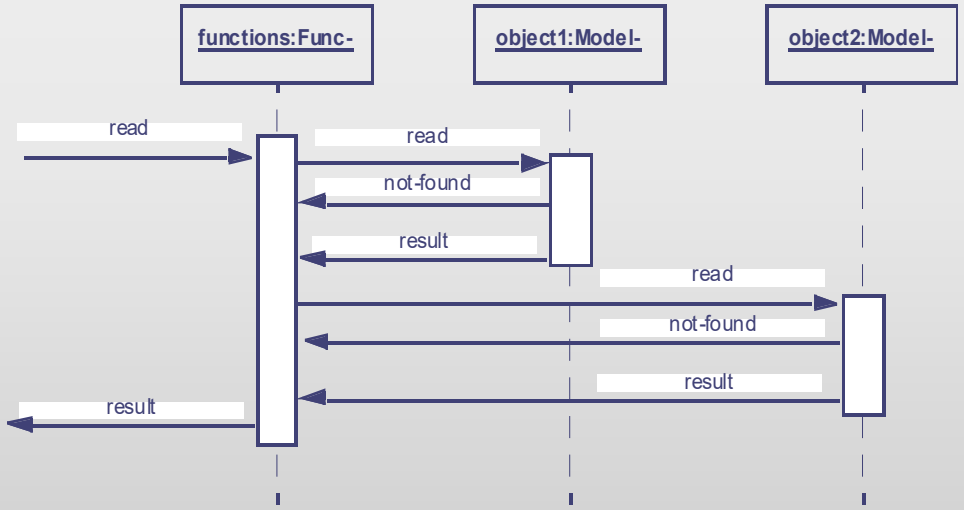
Update

- A relevant event in the problem domain must leave a trace in the model.
- Recieves input describing the event.
- Primary effect is an updated model.
- Identify relevant objects and connections by means of the events.
- The update operation should check that the event is legal.
- This can be described in a sequence diagram (illustration)



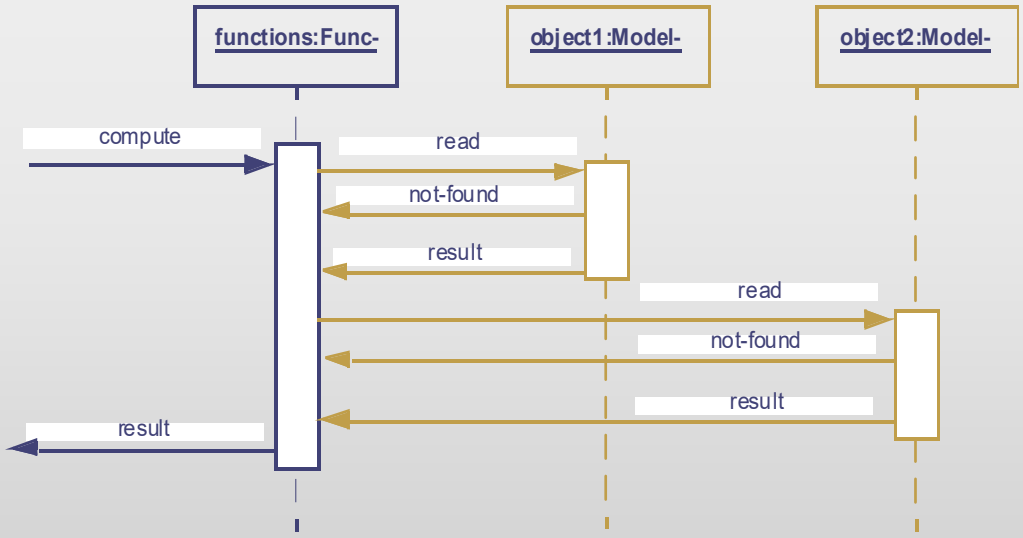
Read

- An actor needs information about the model.
- Input describes the wanted reading and selection of objects.
- Describe the necessary operations.
- Can be done through precedence analysis backward from expected output.



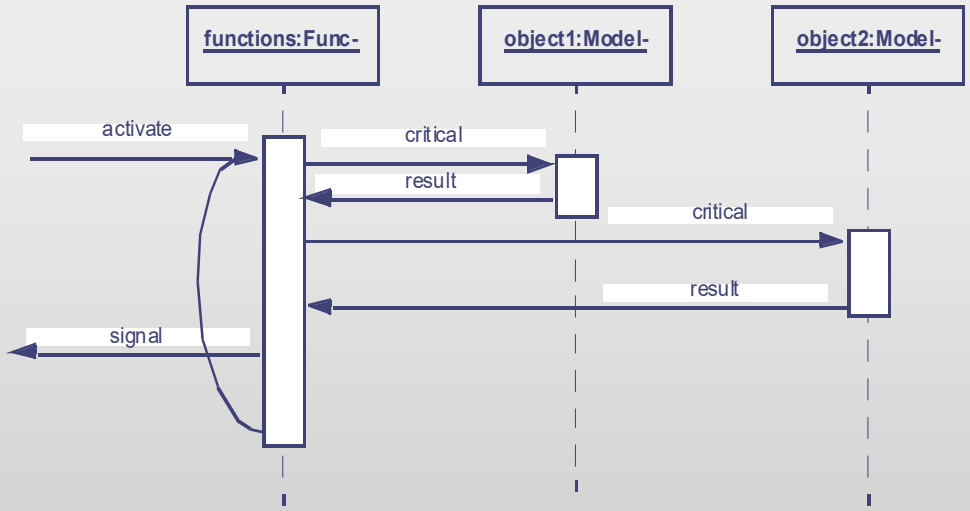
Compute

- An actor needs a computation performed.
- Can also encompass reading and updating
- Describe input and readings of the model.
- Describe algorithm possibly through decomposition.



Signal

- An actor needs to monitor or control a part of the problem domain.
- The critical state is read from the model.
- Typically few inputs.
- Identify the state transitions that might require a signal.
- Determine how to signal.
- Is the signaling active or passive?



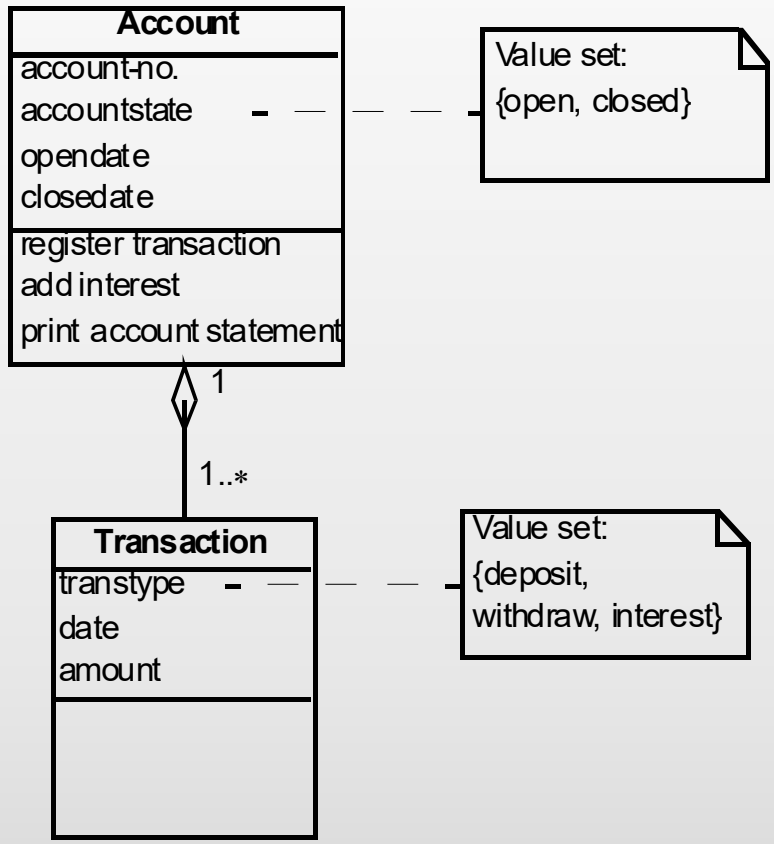
Explore Patterns

The patterns specify how functions can be realized as a set of operations:

- Model-Class Placement
- Function-Class Placement
- Strategy
- Active Function

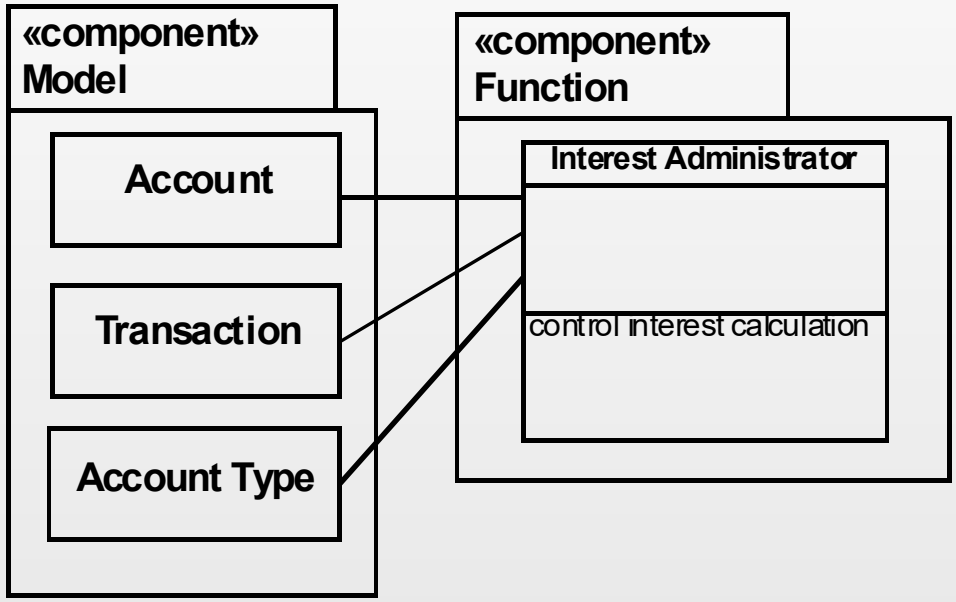
Pattern: Model-Class Placement

- A number of operations are specified on class Account. That again is realized through several operations:
  - Transaction registration (update)
  - Calculate interests (compute) and deposit interests (update)
  - Print account statement (read)



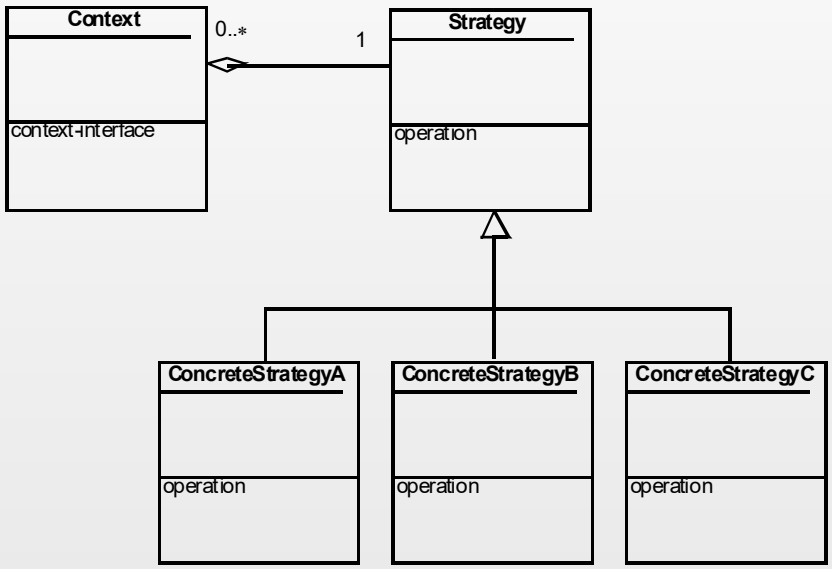
Pattern: Function-Class Placement

- Some operations cannot be placed on a class in the model.
- Typically functions that operate on several objects.
- A new class is then designed in the function component. That class contains the operation that realizes the function.



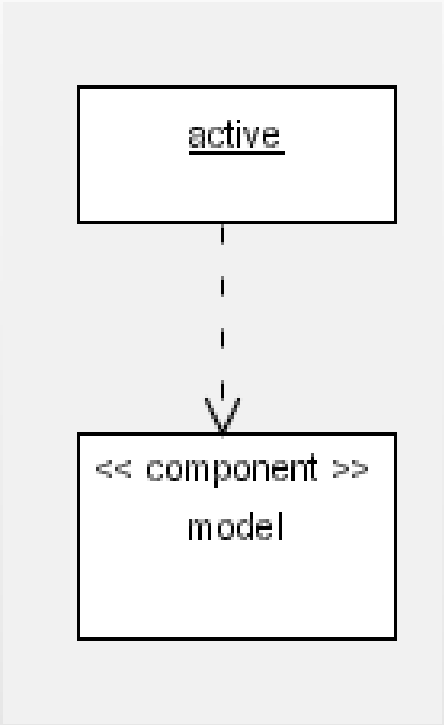
Pattern: Strategy

- If a class has several specializations and a function is performed different dependent on each specialization.
- The Stategy Pattern defines a general operation that is then described in detail in each specialization.



Pattern: Active Function

- A signal function can be active or passive.
- An active function can be realized in an active object.
- The function is then realized with its own control.

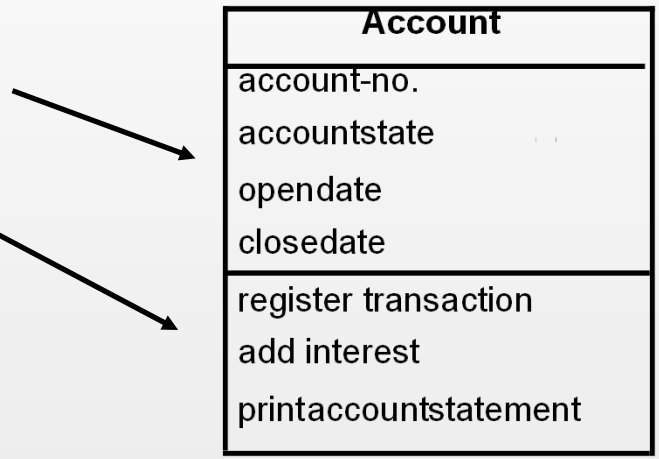


Specify Complex Operations

- Implicit specification by an attribute in the class
- Naming the operation in the class
- Operation specification for complex functions (next slide)

More unusual specifications

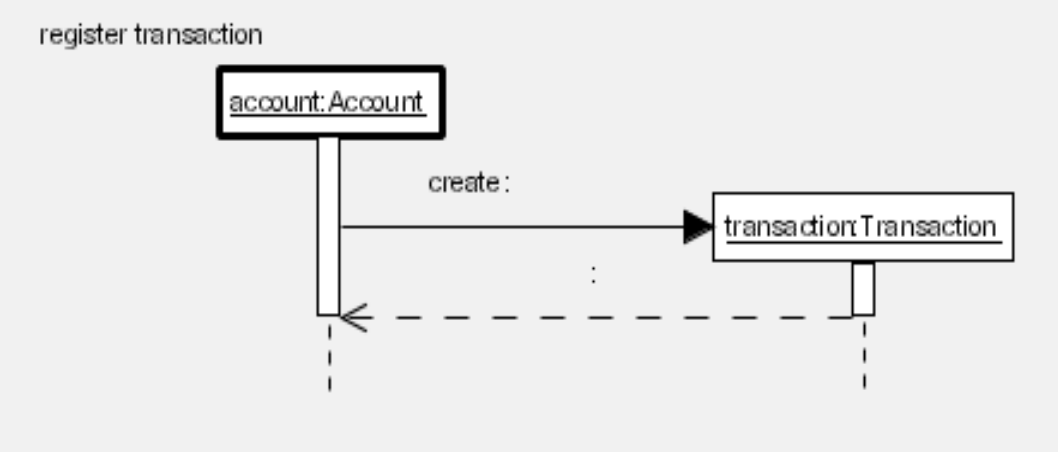
- Sequence Diagram
- Statechart for a class
- Statechart for a system

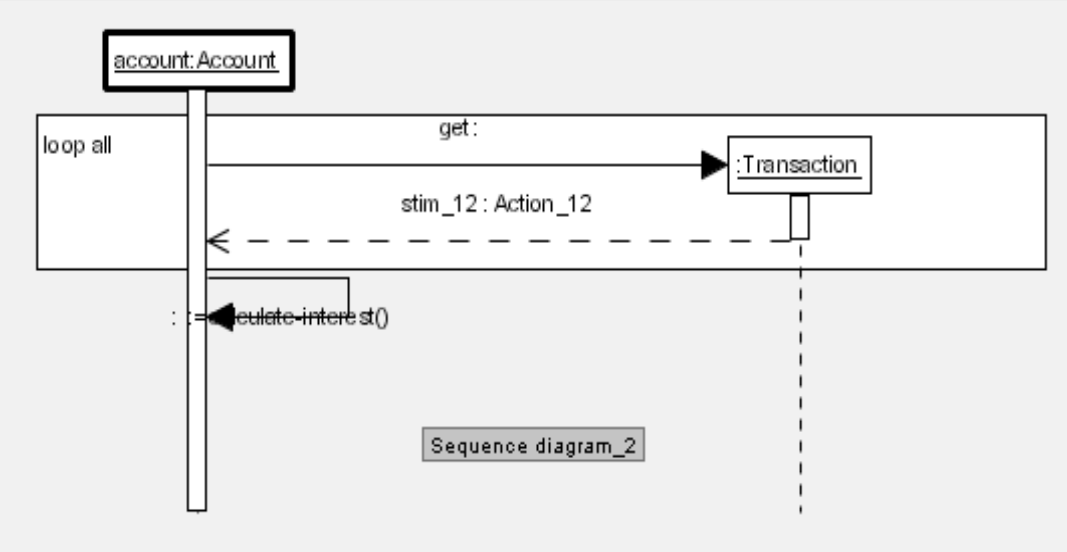


Operation Specification for a Function

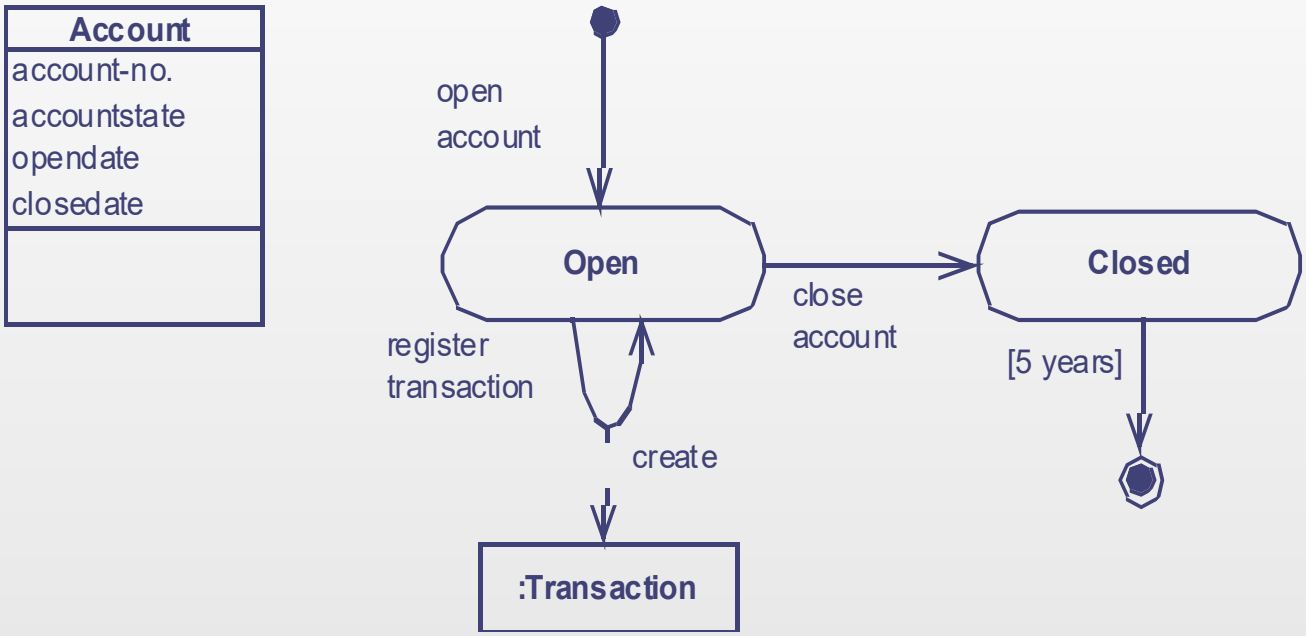
Name	Register transaction	
Category	_ Active x Passive	x Update _ Read _ Compute _ Signal
Purpose	Establishes a new transaction for a specific account.	
Input data	account-no., transtype, date, amount.	
Conditions	An object of the class Account, with the given account number, exists. The attribute accountstate in this object has the value Open.	
Effect	A new object of the class Transaction is established with input data assigned to the attributes. This object is connected to the relevant Account object.	
Algorithm		
Data structures		
Placement	Account.	
Involved objects	Account, Transaction.	
Triggering events	amount deposited, amount withdrawn.	

Sequence Diagram

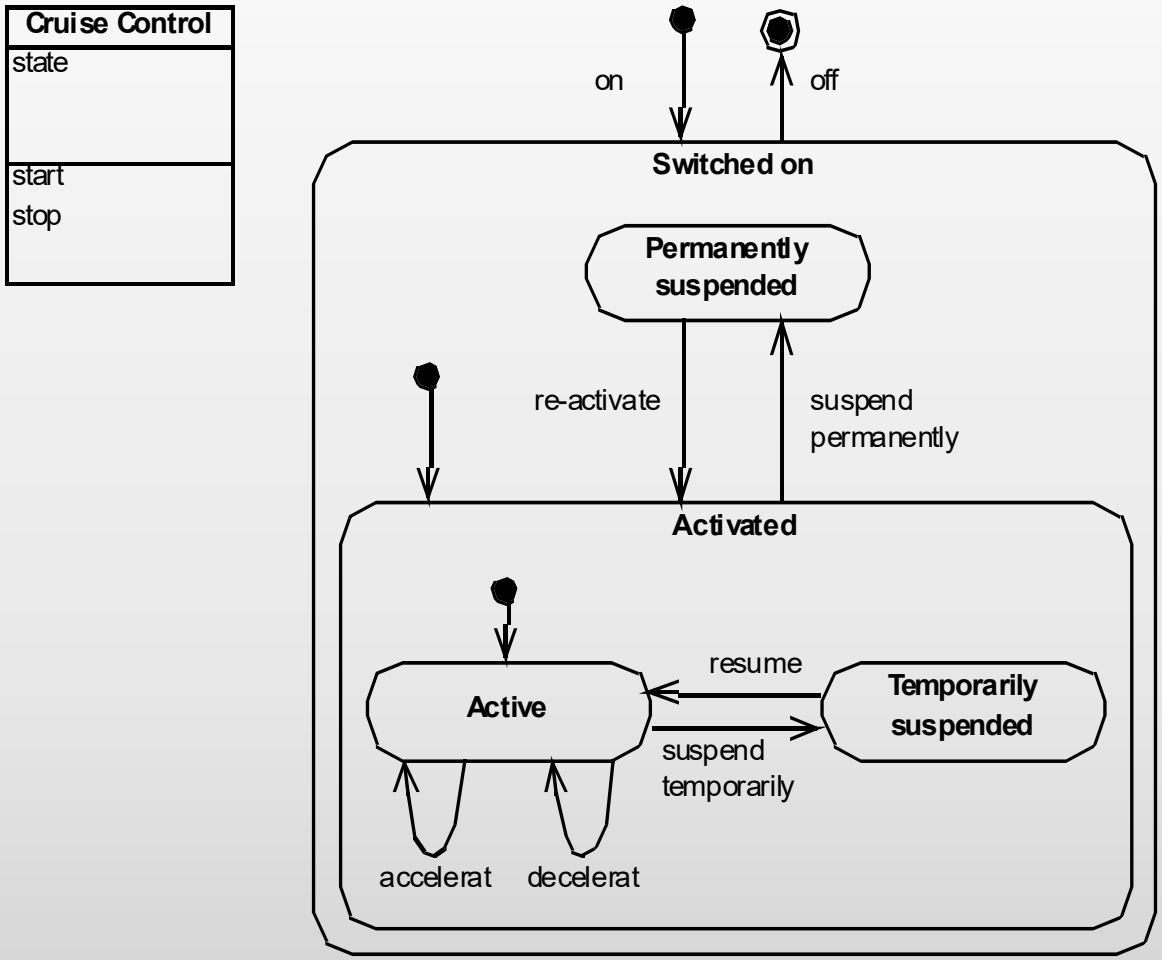




Statechart for a Class



Statechart for the System's Total Behavior



Function Component: Summary

Purpose	• To determine the implementation of functions.
Concepts	• Function component: A part of a system that implements functional requirements. • Operation: A process property specified in a class and activated through the class' objects.
Principles	• Base the design on function types. • Specify complex operations.
Results	• A class diagram with operations and specifications of complex operations.

Contents

- Summary of last lecture
- The Function Component activity
- Solution to Written exam 2018-06