Systems Development

Lecture 9: Component Design and Model Component

Contents

- Summary of last lecture
- Component design
- The Model Component activity
- Solution to Written exam 2018-01

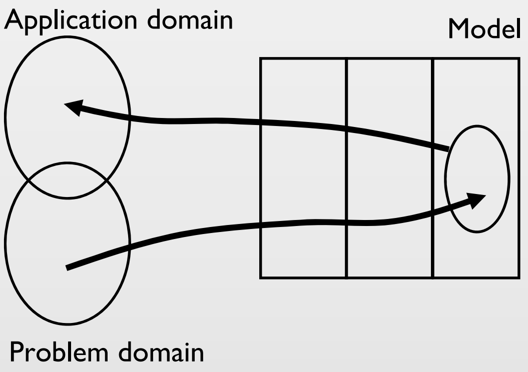
Contents

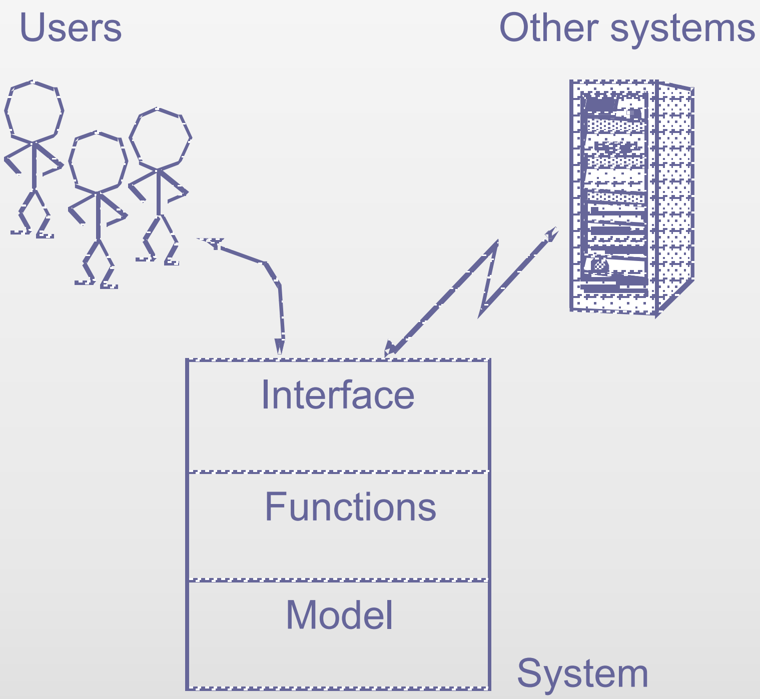
- Summary of last lecture
  - Why are we making the descriptions?
  - Architectural design
  - The Criteria activity
  - The Components activity
- Component design
- The Model Component activity
- Solution to Written exam 2018-01

Architectural Design: Key Concepts

System:

A collection of components that implements modeling requirements, functions, and interfaces.





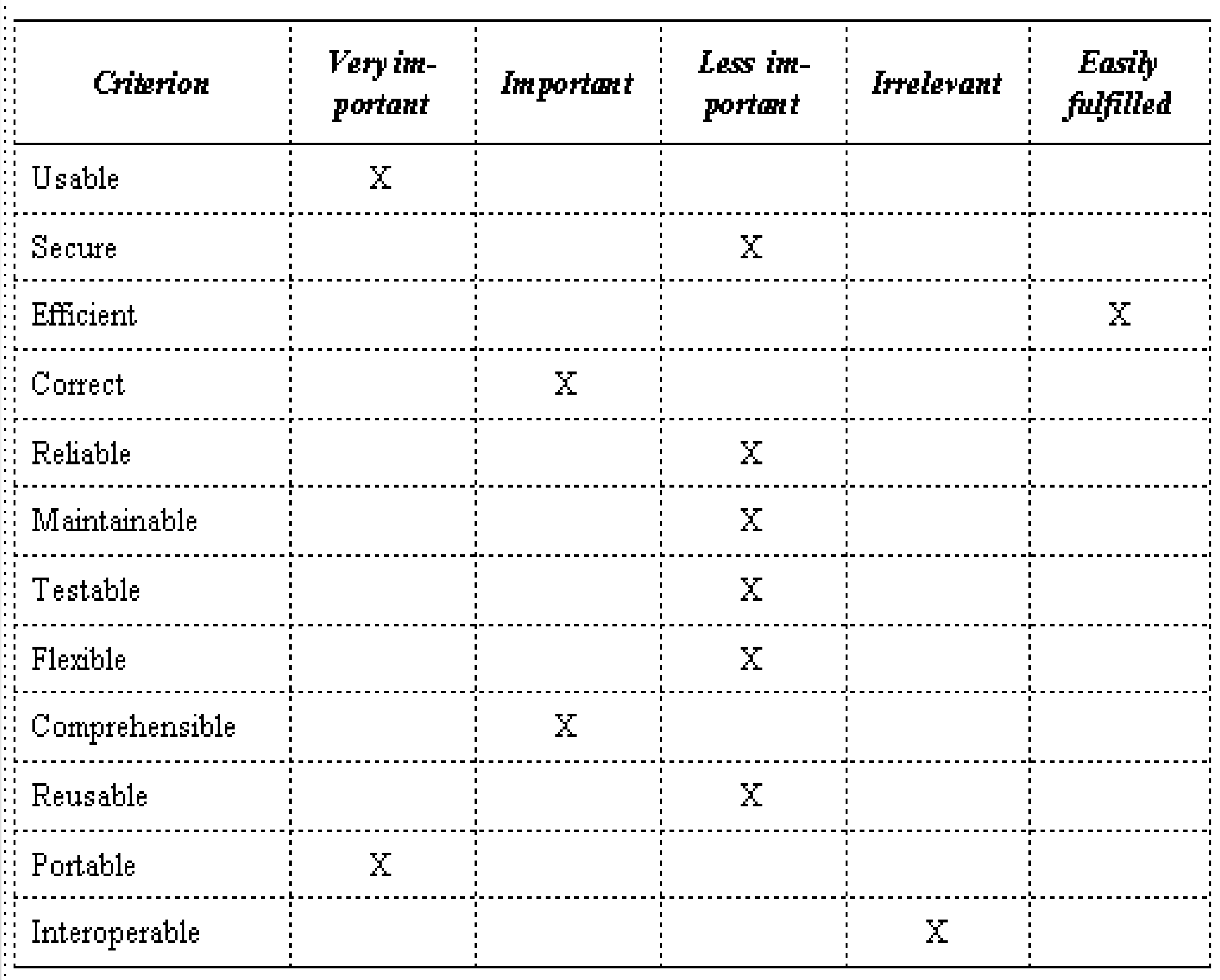
Architecture:
A general structure that is later developed further

Architectural Design: Summary

Purpose	• To structure a computerized system.
Concepts	• Criterion: A preferred property of an architecture. • Component architecture: A system structure composed of interconnected components. • Process architecture: A system-execution structure composed of interdependent processes.
Principles	• Define and prioritize criteria. • Bridge criteria and technical platform. • Evaluate designs early.
Results	• Structures for a system’s components and processes.

Criteria: Result

- A collection of prioritized criteria
- Reasons for the priority of each criterion

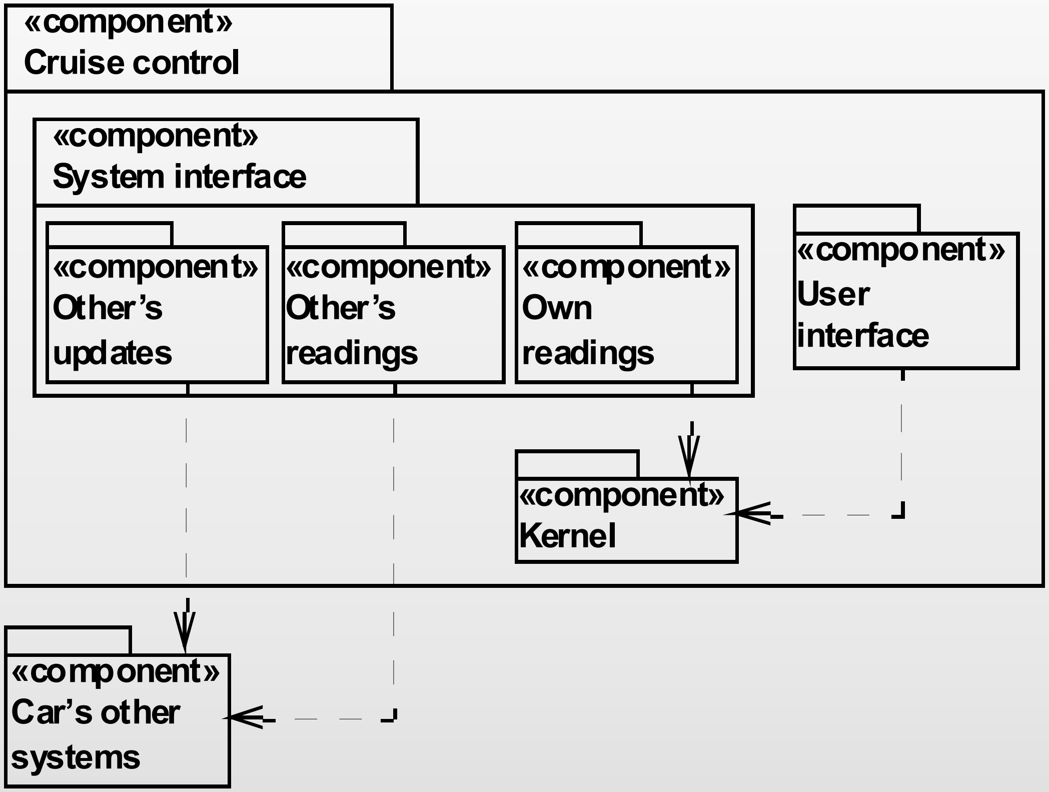


Criterion	*Very important*	*Important*	*Less important*	*Irrelevant*	*Easily fulfilled*
Usable	X				
Secure			X		
Efficient					X
Correct		X			
Reliable			X		
Maintainable			X		
Testable			X		
Flexible			X		
Comprehensible		X			
Reusable			X		
Portable	X				
Interoperable				X	

Criteria: Summary

Purpose	• To set design priorities.
Concepts	• Criterion: A preferred property of an architecture. • Conditions: The technical, organizational, and human opportunities and limits involved in performing a task.
Principles	• A good design has no major weaknesses. • A good design balances several criteria. • A good design is usable, flexible, and comprehensible.
Results	• A collection of prioritized criteria.

Components: Result

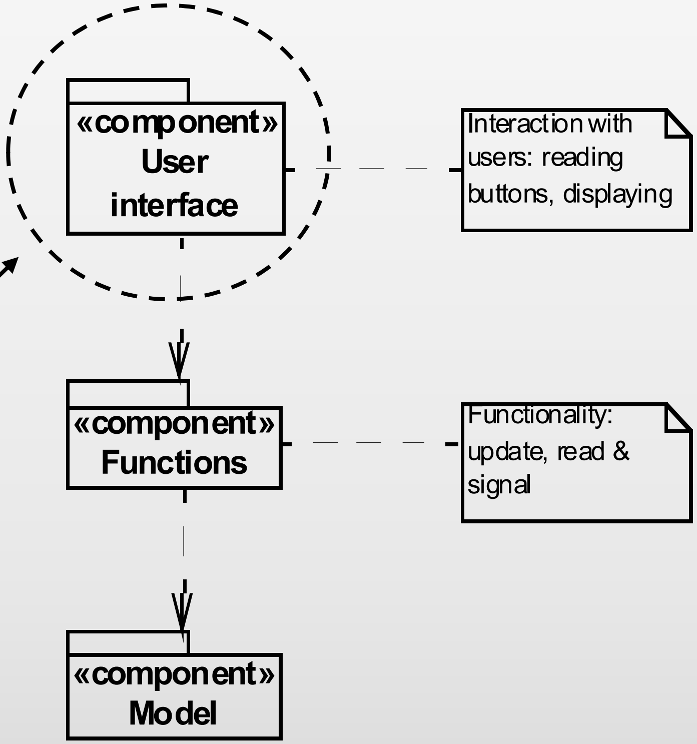


- A structural perspective
- Separates concerns in a system
- Emphasizes comprehensibility and flexibility

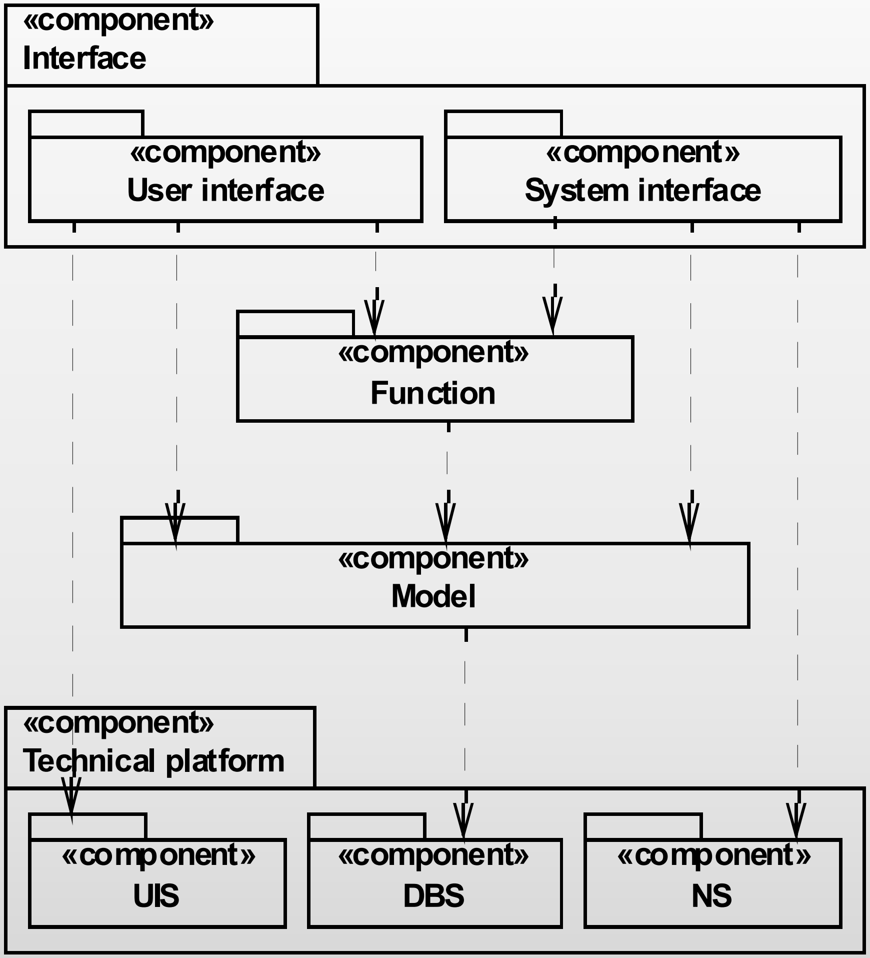
Key Concept: Component

- A collection of program parts
- Constitutes a totality
- Has a well-defined responsibility
- Smallest: a class
- Largest: a system

- Example:
  This component has the responsibility for reading the buttons and updating the display


«component» User interface
Interaction with users: reading buttons, displaying
«component» Functions
Functionality: update, read & signal
«component» Model

Pattern: The Generic Architecure

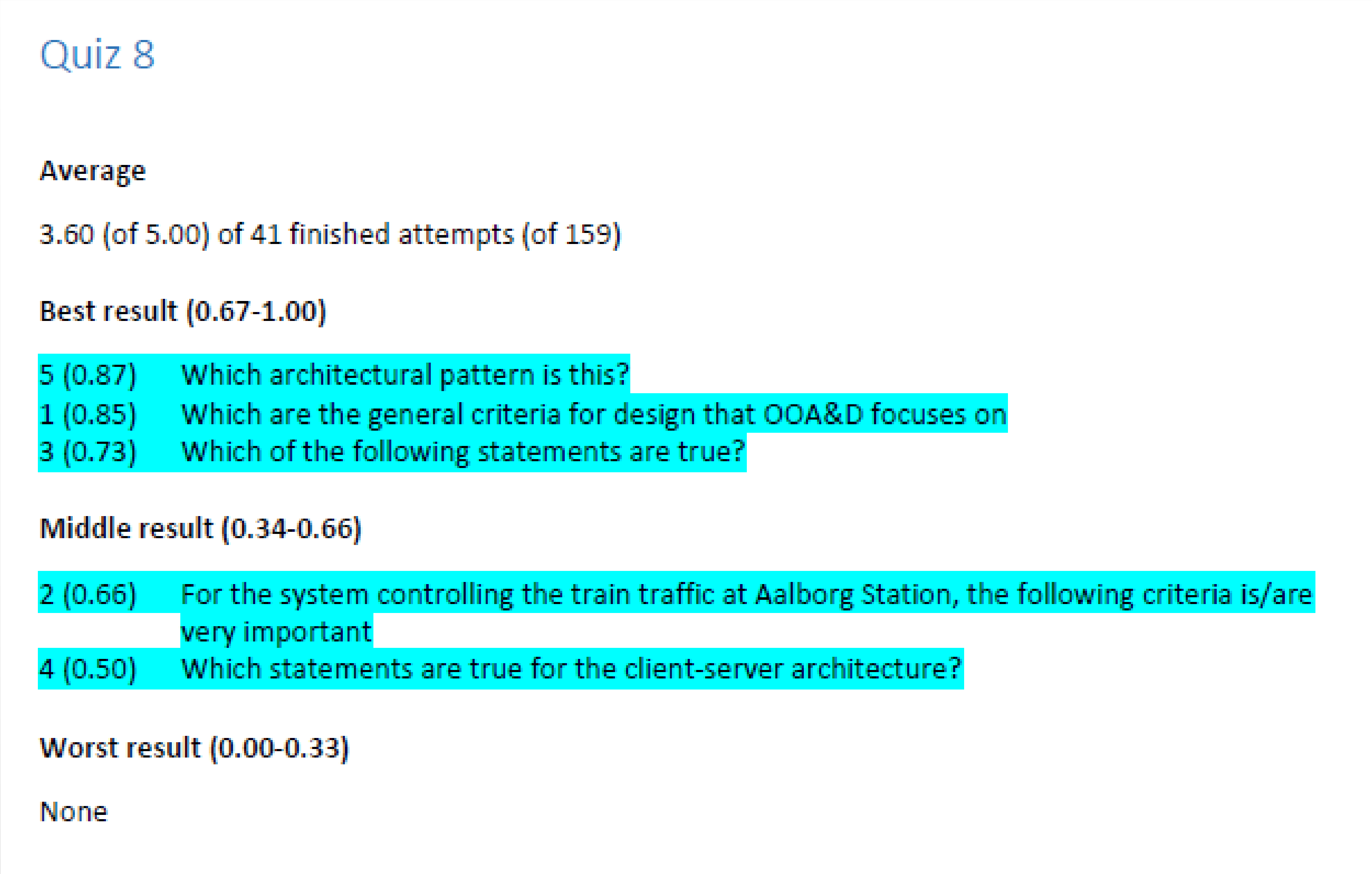


- The generic architecture reflects the division of the context into problem domain and application domain
- "Technical platform" is an extension and encapsulation of the underlying technical platform

Components: Summary

Purpose	• To create a comprehensible and flexible system structure
Concepts	• Component architecture: a system structure of interconnected components. • Component: a collection of program parts that constitutes a whole and has well-defined responsibilities.
Principles	• Reduce complexity by separating concerns. • Reflect stable context structures. • Reuse existing components.
Result	• A class diagram with specifications of the complex components.

Quiz 8



Quiz 8

Average

3.60 (of 5.00) of 41 finished attempts (of 159)

Best result (0.67-1.00)

5 (0.87) Which architectural pattern is this?
1 (0.85) Which are the general criteria for design that OOA&D focuses on
3 (0.73) Which of the following statements are true?

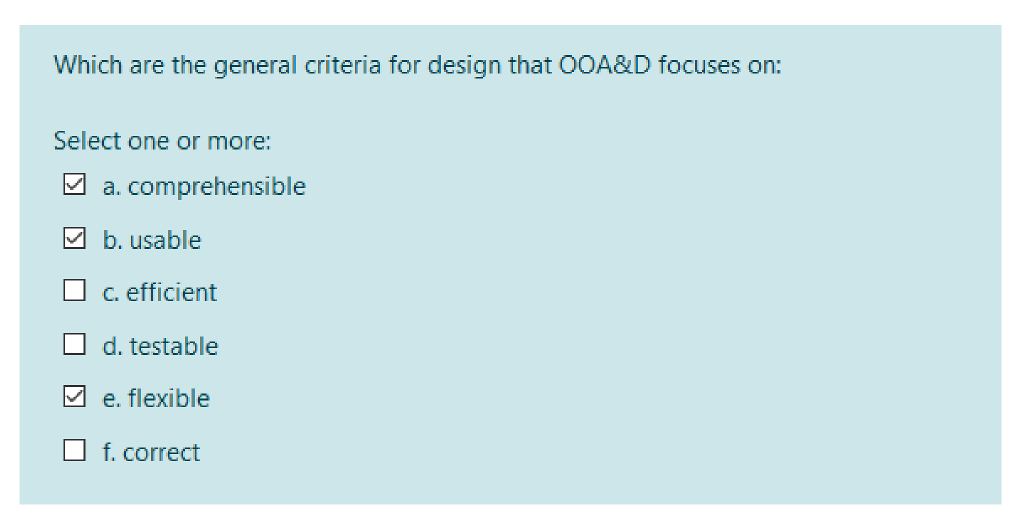
Middle result (0.34-0.66)

2 (0.66) For the system controlling the train traffic at Aalborg Station, the following criteria is/are very important
4 (0.50) Which statements are true for the client-server architecture?

Worst result (0.00-0.33)

None

Question 1 and 2



Which are the general criteria for design that OOA&D focuses on:

Select one or more:

☑ a. comprehensible

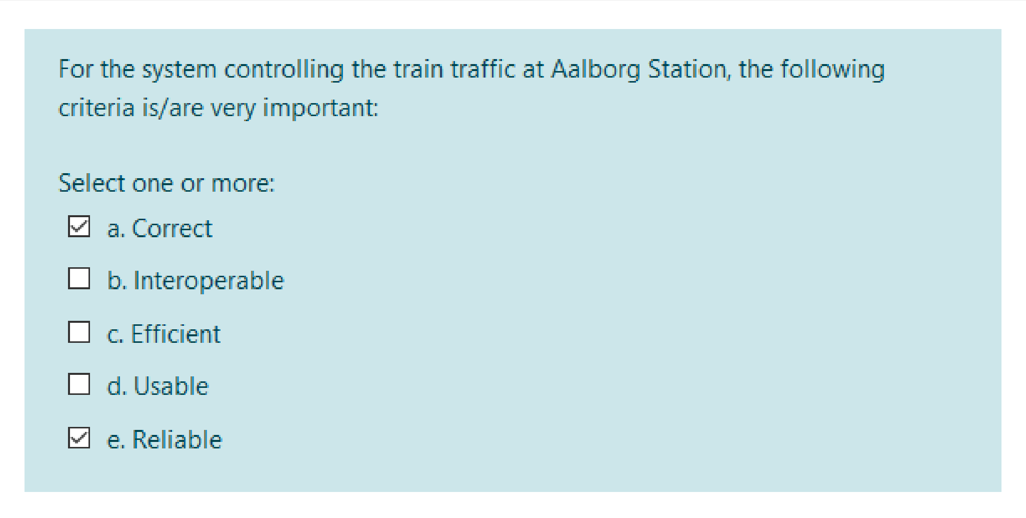
☑ b. usable

□ c. efficient

□ d. testable

☑ e. flexible

□ f. correct



For the system controlling the train traffic at Aalborg Station, the following criteria is/are very important:

Select one or more:

☑ a. Correct

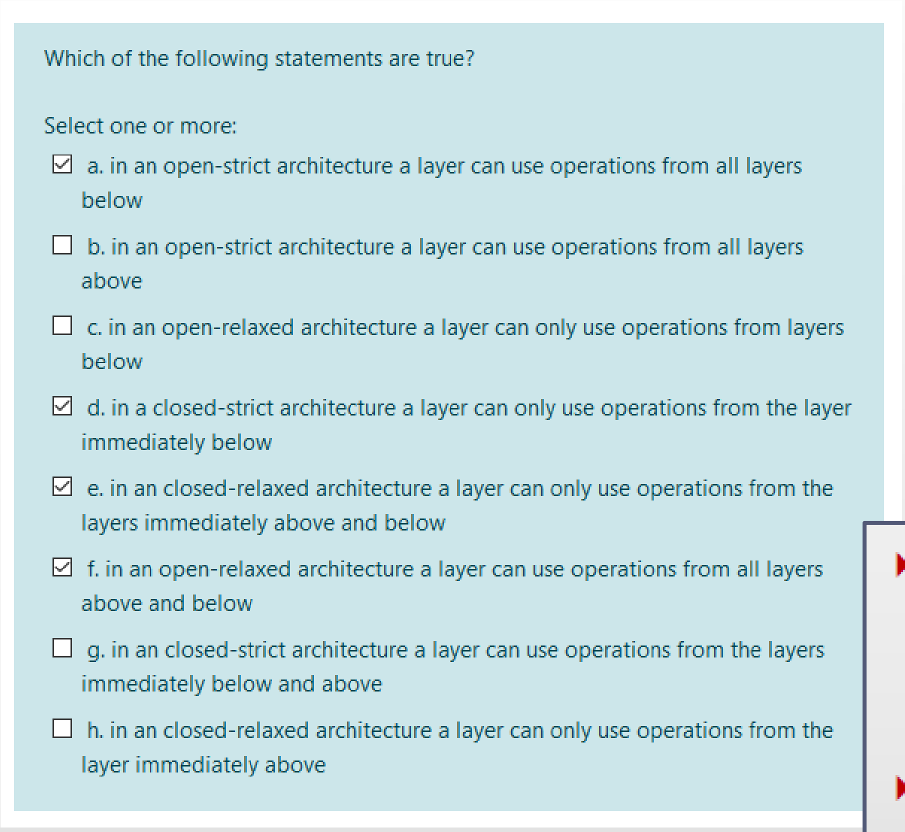
□ b. Interoperable

□ c. Efficient

□ d. Usable

☑ e. Reliable

Question 3 and 4



Which of the following statements are true?

Select one or more:

☑ a. in an open-strict architecture a layer can use operations from all layers below

□ b. in an open-strict architecture a layer can use operations from all layers above

□ c. in an open-relaxed architecture a layer can only use operations from layers below

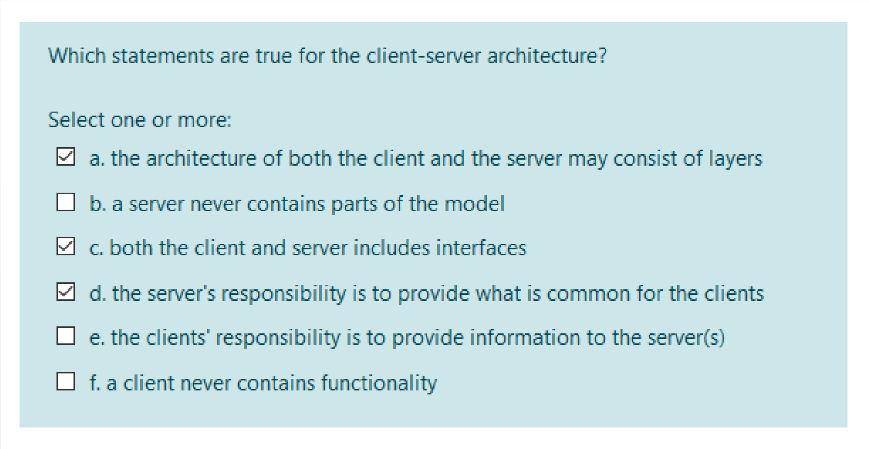
☑ d. in a closed-strict architecture a layer can only use operations from the layer immediately below

☑ e. in an closed-relaxed architecture a layer can only use operations from the layers immediately above and below

☑ f. in an open-relaxed architecture a layer can use operations from all layers above and below

□ g. in an closed-strict architecture a layer can use operations from the layers immediately below and above

□ h. in an closed-relaxed architecture a layer can only use operations from the layer immediately above



Which statements are true for the client-server architecture?

Select one or more:

☑ a. the architecture of both the client and the server may consist of layers

□ b. a server never contains parts of the model

☑ c. both the client and server includes interfaces

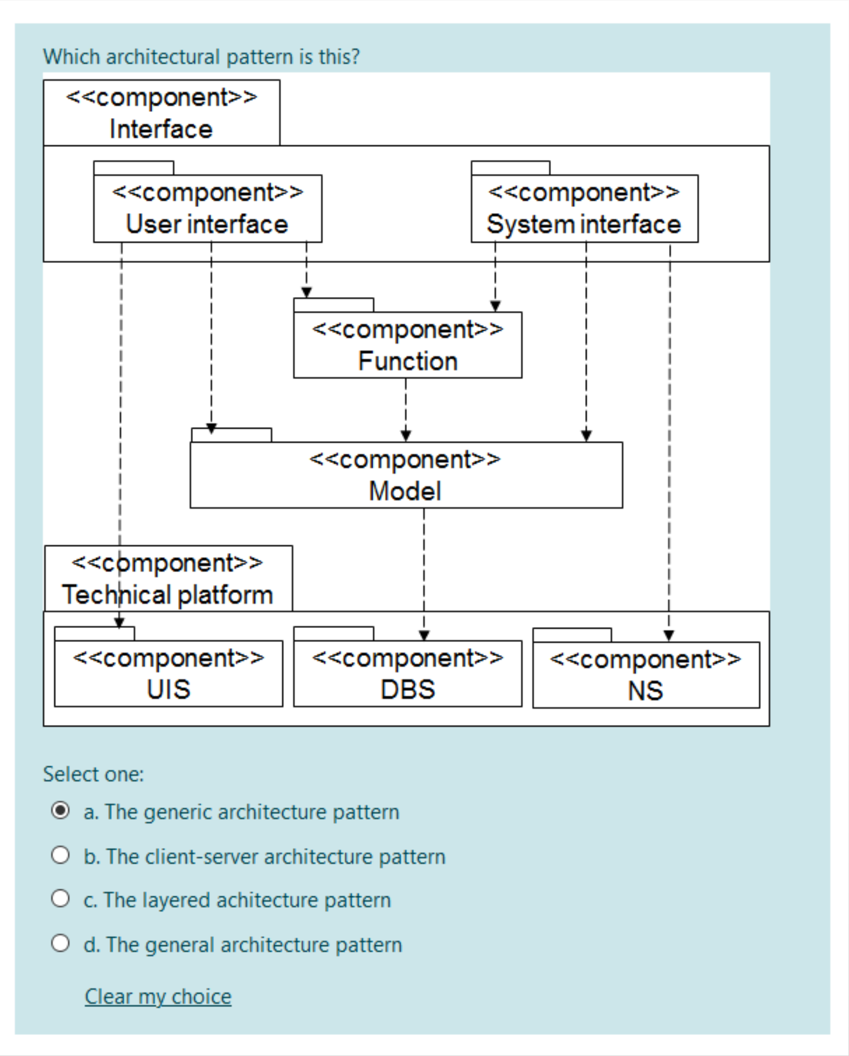
☑ d. the server's responsibility is to provide what is common for the clients

□ e. the clients' responsibility is to provide information to the server(s)

□ f. a client never contains functionality

- Closed architecture: only apply operations from an adjacent layer
  Open arcitecture: apply operations from any other layer
- Strict architecture: only apply operations from a layer below
  Relaxed architecture: apply operation from layer both above and below

Question 5



Which architectural pattern is this?

Select one:

- ● a. The generic architecture pattern
- ○ b. The client-server architecture pattern
- ○ c. The layered achitecture pattern
- ○ d. The general architecture pattern

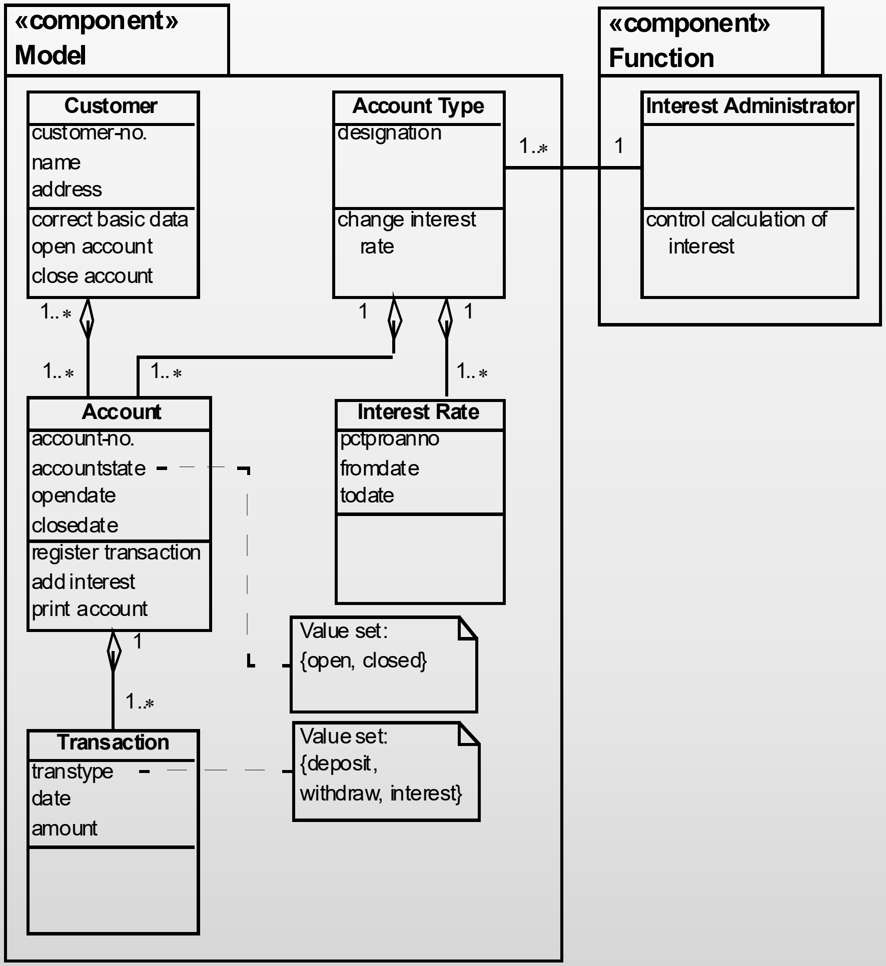
Clear my choice

Contents

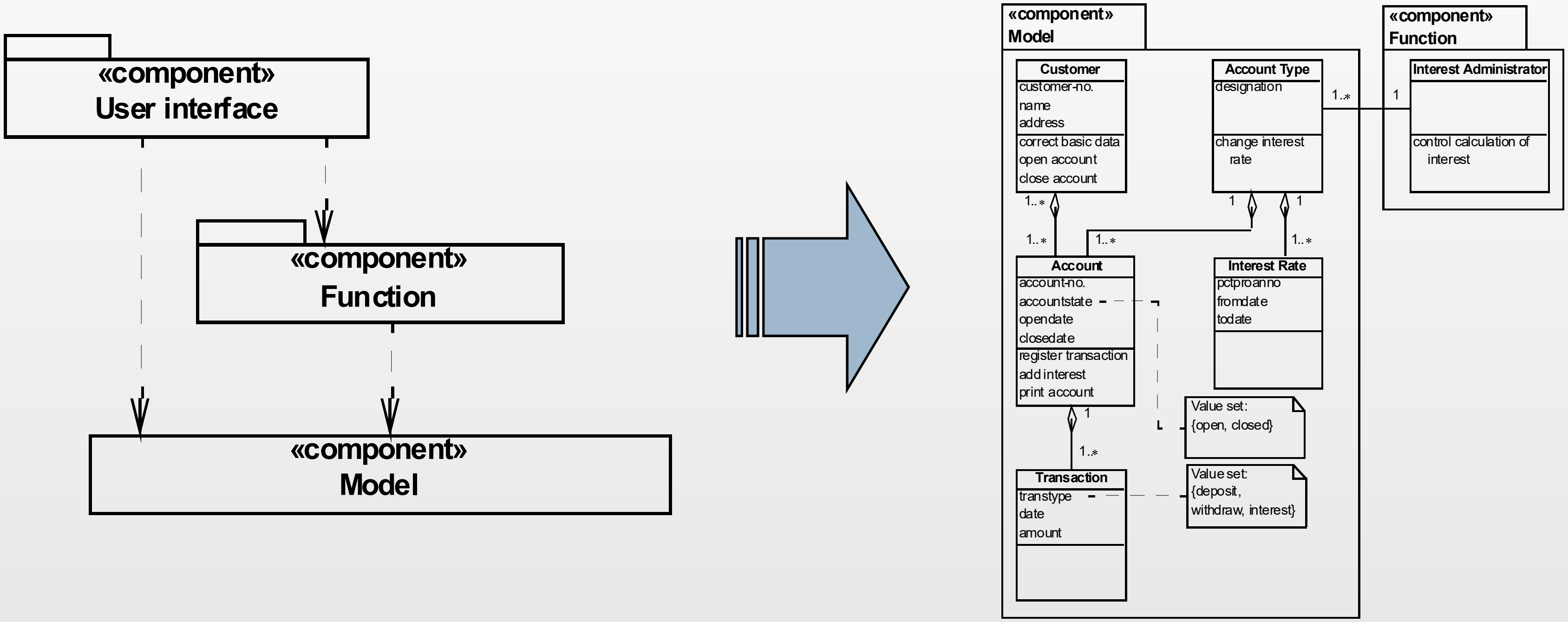
- Summary of last lecture
- Component design
  - Results
  - Key concepts
  - Activities
- The Model Component activity
- Solution to Written exam 2018-01

Component Design: Results

- Details in individual components
- Connections between components
- Iterate architecture
  - Use and revise division into components



Key Concepts: From Architecture to Components



Principles:

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

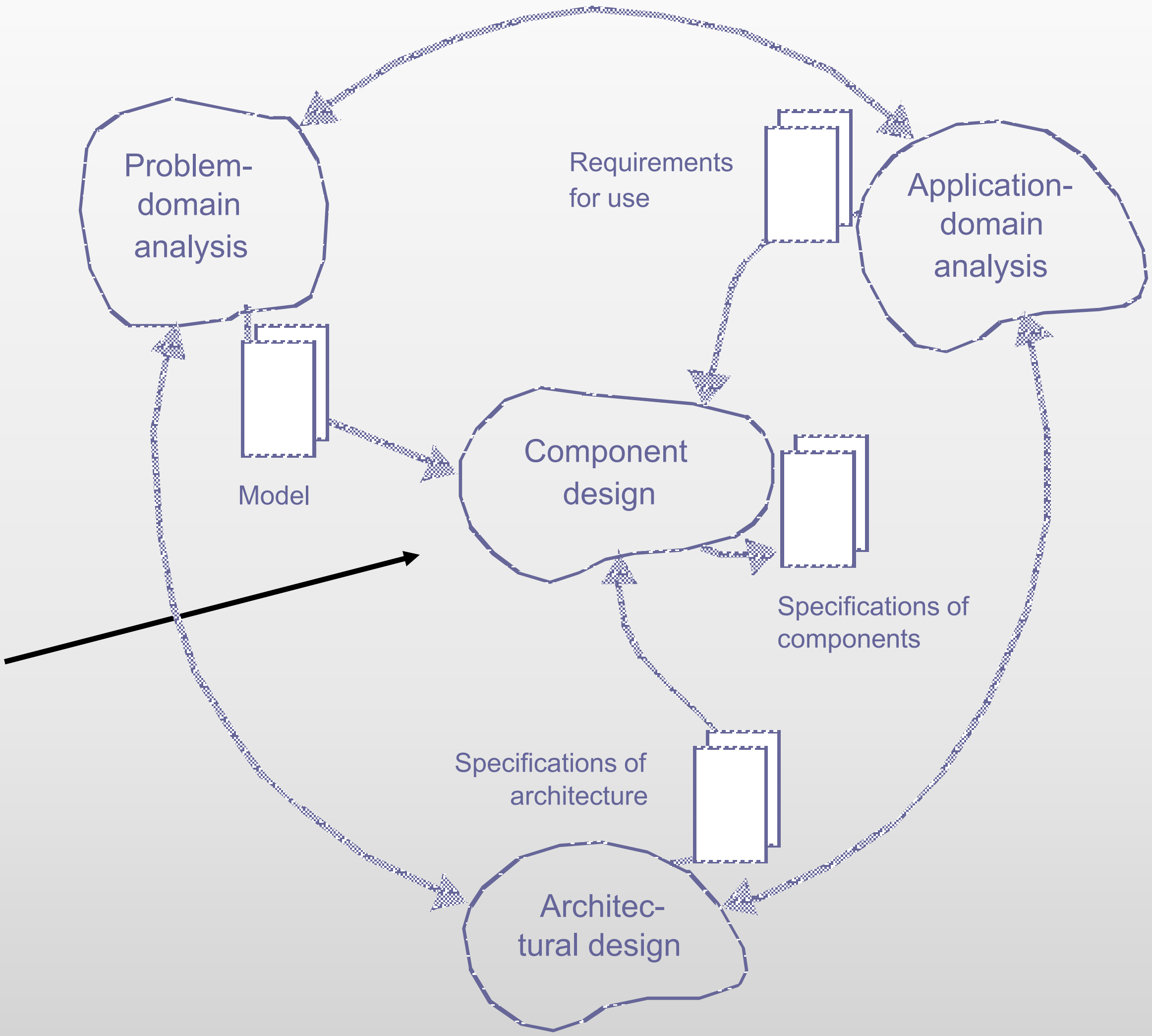
Component Design: Activities

Model component

Function component

Connect components

... more components



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

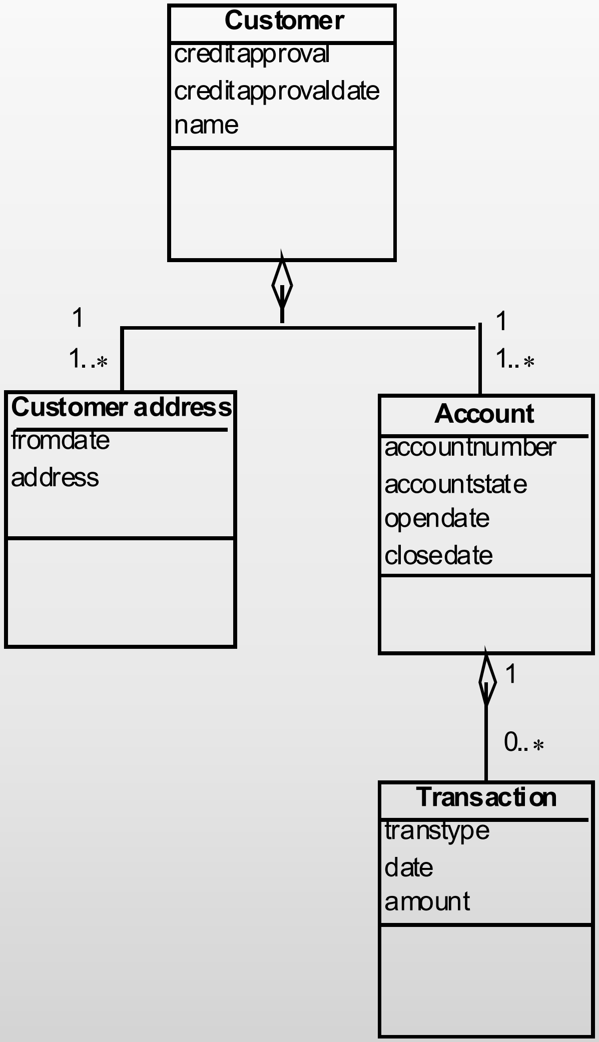
Component Design: Summary

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.

Contents

- Summary of last lecture
- Component design
- The Model Component activity
  - Results
  - Key concepts
  - Activities
- Solution to Written exam 2018-01

Model Component: Results



Customer
creditapproval
creditapprovaldate
name

1
1..*

Customer address
fromdate
address

1
1..*

Account
accountnumber
accountstate
opendate
closedate

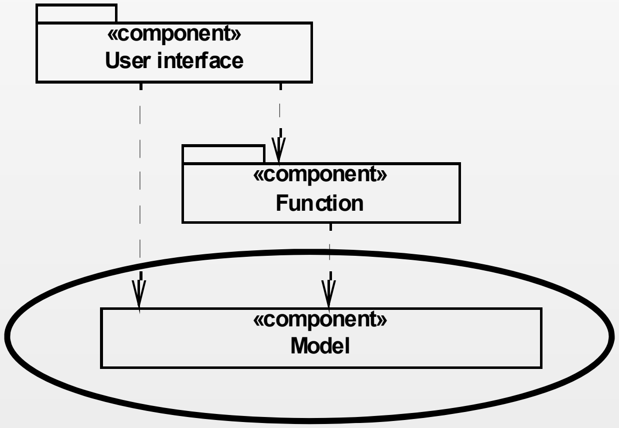
1
0..*

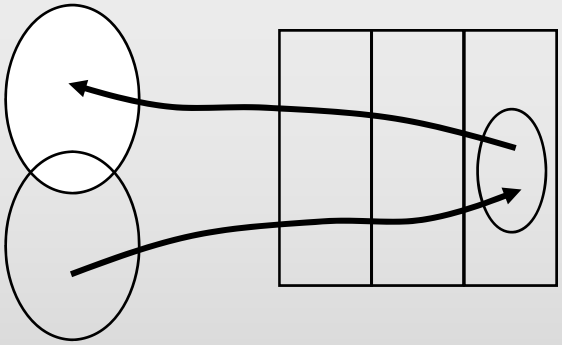
Transaction
transtype
date
amount

- Point of departure in the class diagram from the problem domain analysis
- Extended with representation of the data contents of the behavior described in the statechart diagrams
- The statechart diagrams are no longer needed for implementation of the model component

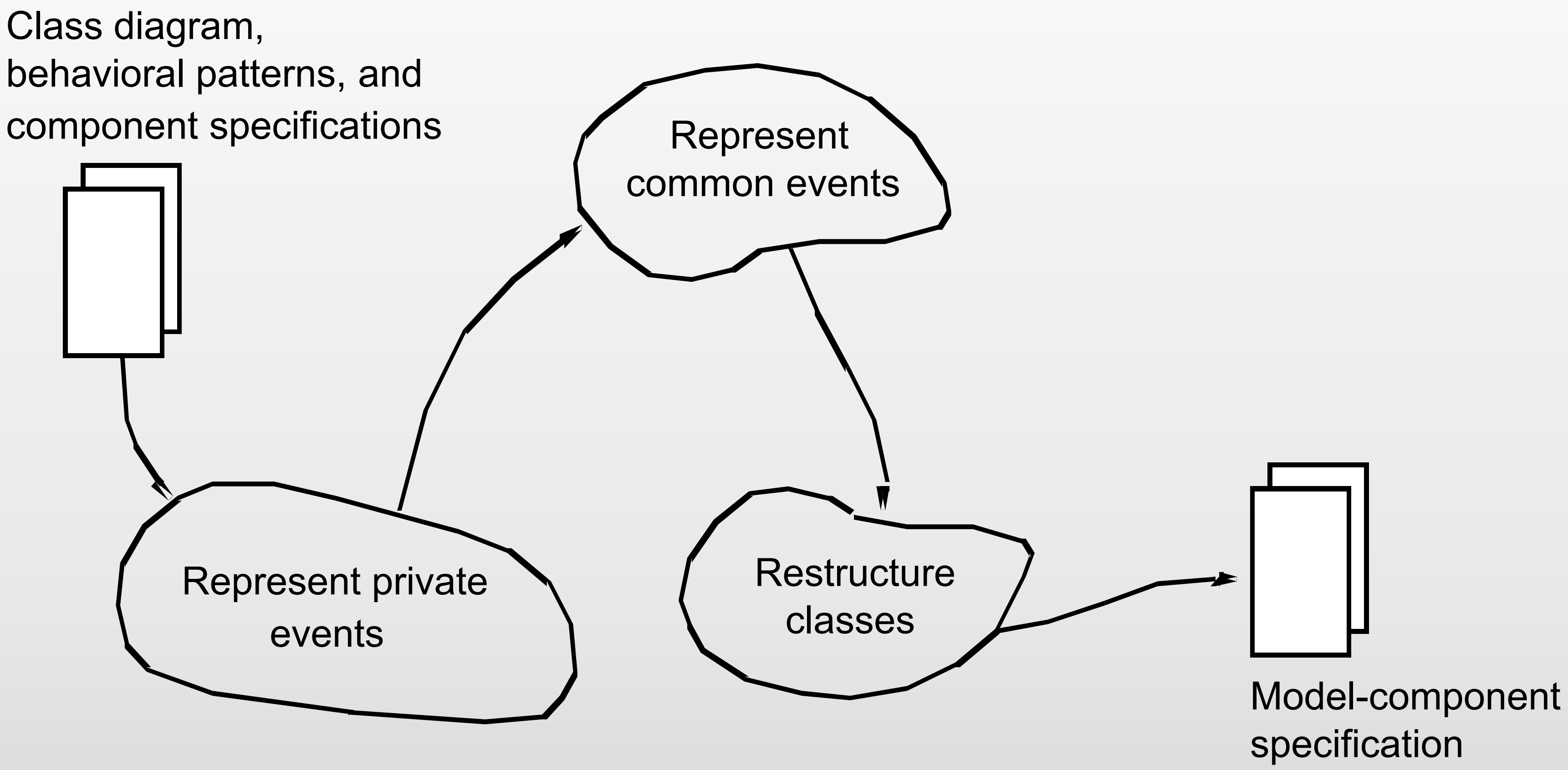
Key Concepts: From Totality to Part

- **Component:**
  A collection of program parts that constitutes a whole and has well-defined responsibilities
- **Responsibility of the model component:**
  Maintain an updated representastion of the problem domain
- ”Remember” what has happened in the problem domain





Model Component: Activities


Class diagram,
behavioral patterns, and
component specifications

Represent common events

Represent private events

Restructure classes

Model-component specification

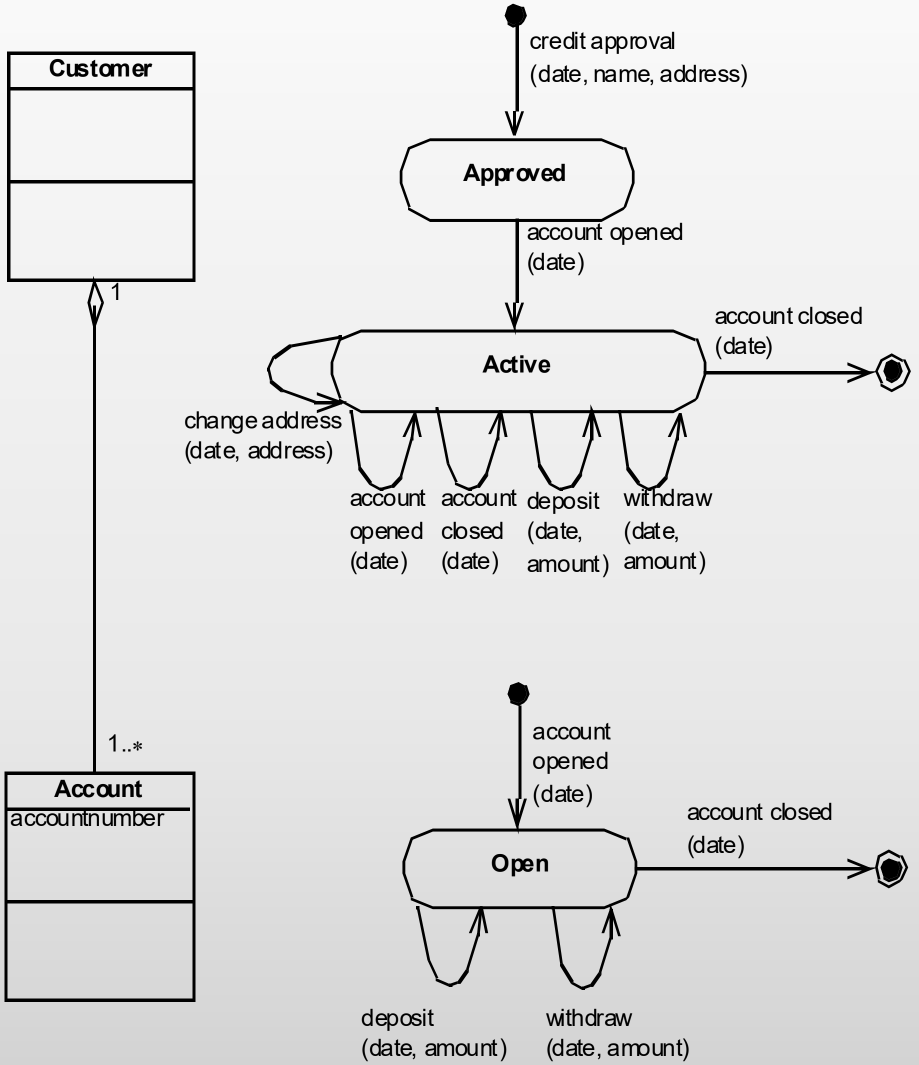
Example: Bank System

Analysis model:

- Class diagram
- Statechart diagrams
- Event table

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

- What do we want to remember here?

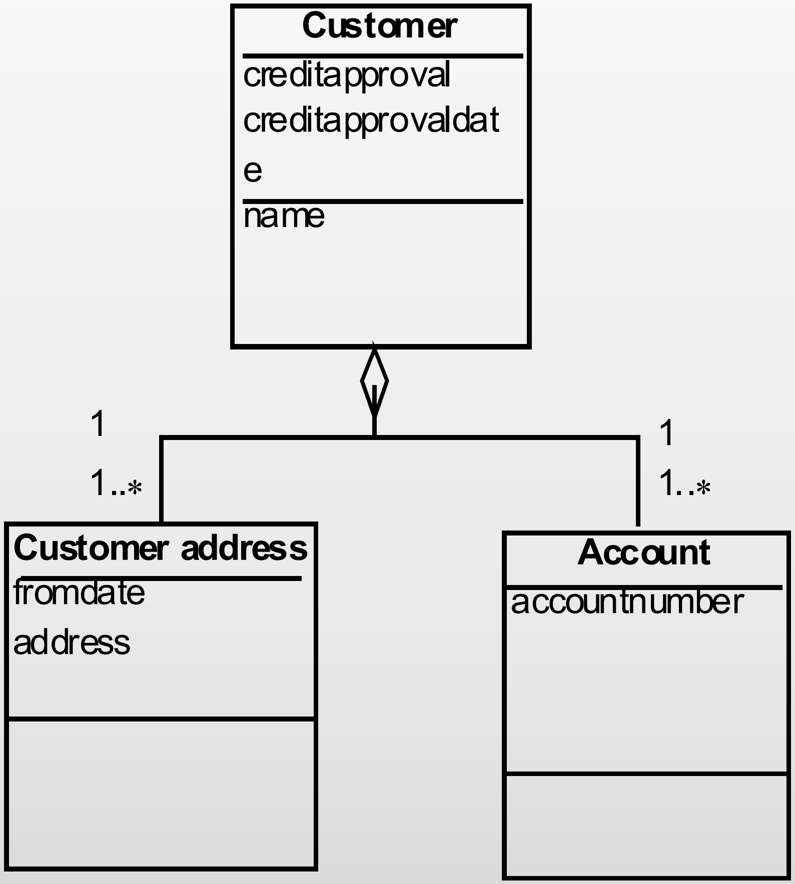


Represent Private Events (1)

- Sequence and selection
  - Represent these events as a state attribute in the class described by the statechart diagram.
  - Every time one of the involved events occurs, the system shall assign a new value to the state attribute.
  - Integrate the attributes of the involved events into the class.
- Iteration
  - Represent these events as a new class; attach it to the class described by the statechart diagram using an aggregation structure.
  - For each iteration that occurs, the system shall generate a new object from the class.
  - Integrate the event attributes into the new class.

Represent Private Events (2)

- 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

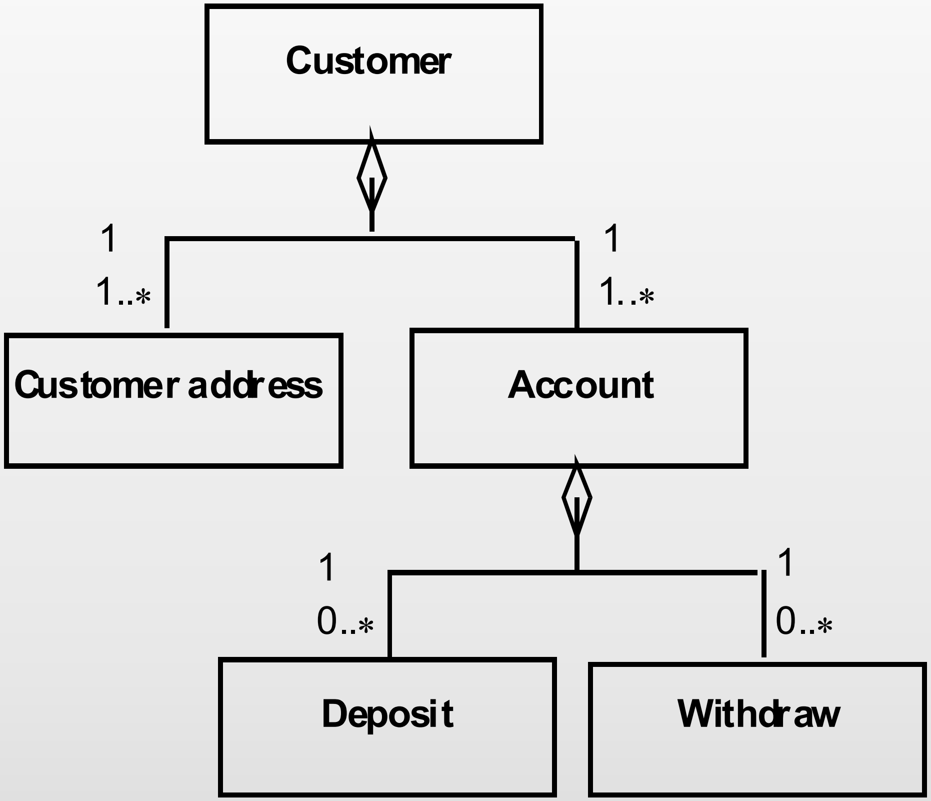


Represent Common Events (1)

- Common events:
  - If the event is involved in the statechart diagrams in different ways, represent it in relation to the class that offers the simplest representation.
  - If the event is involved in the statechart diagrams in the same way, you must weigh possible representations against each other.

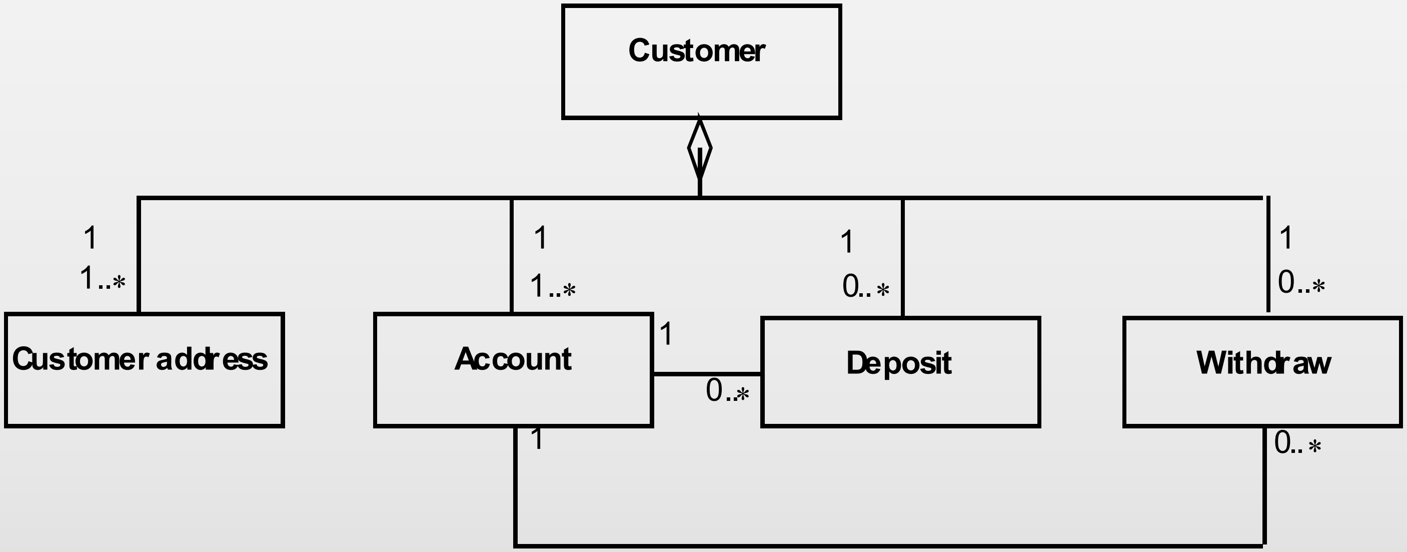
Represent Common Events: Solution A

- 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



Represent Common Events: Solution B

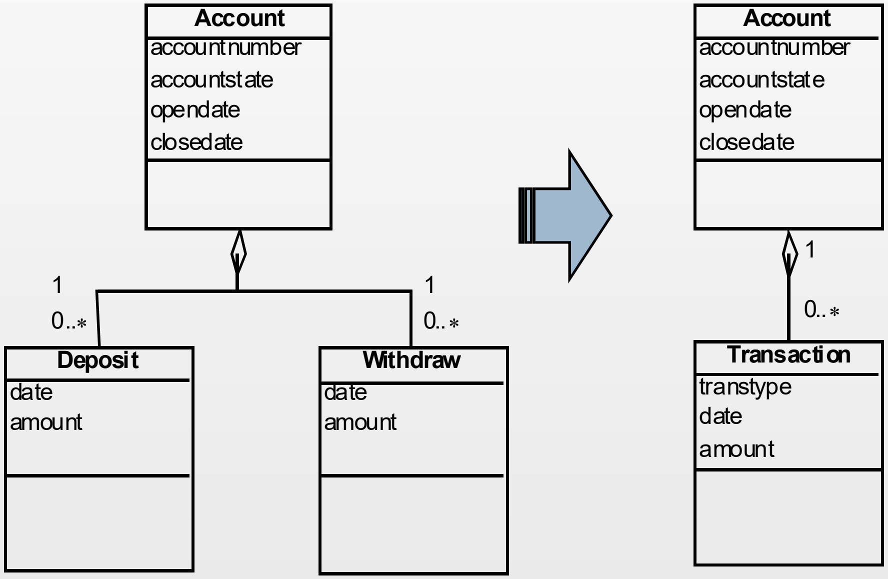
- Alternatively, the events ‘deposit’ and ‘withdraw’ can be represented as new classes under Customer



- B implies a complex structure (two associations across)
- Therefore, we select solution A

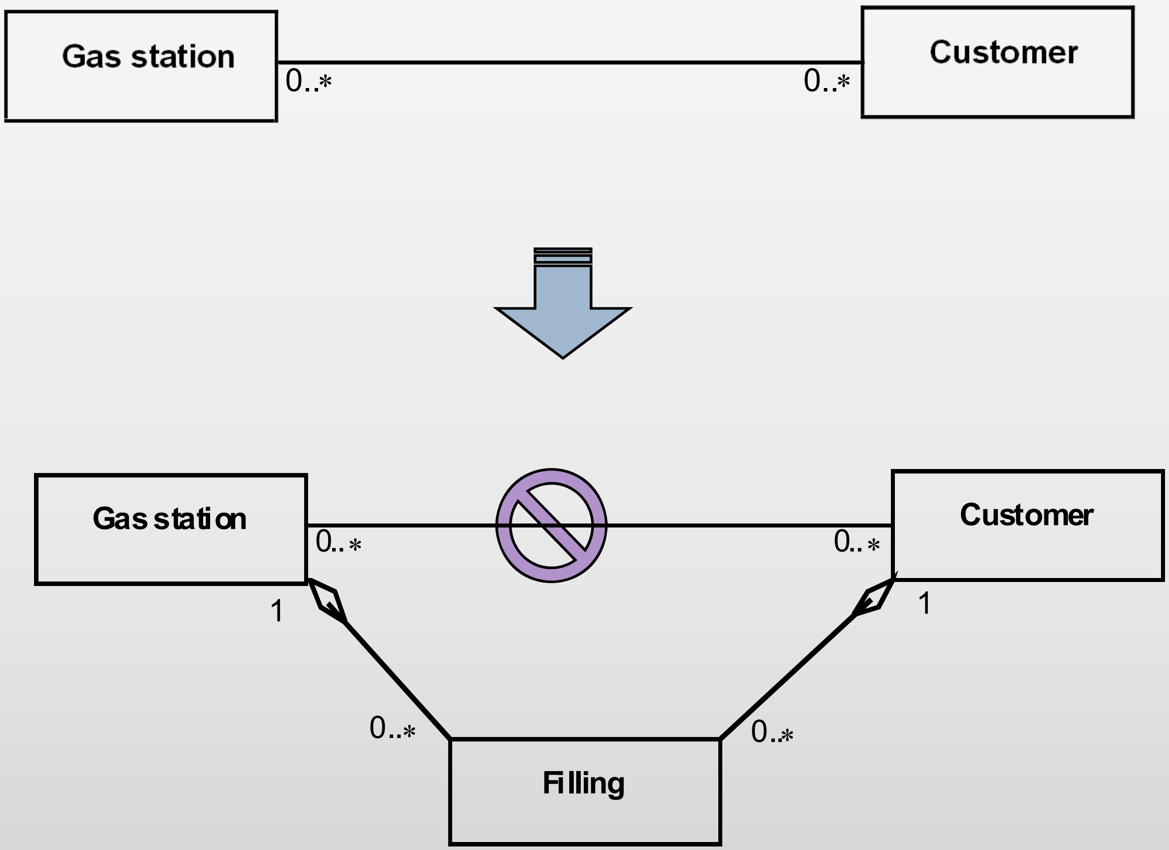
Restructure Classes (1)

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



Restructure Classes (2)

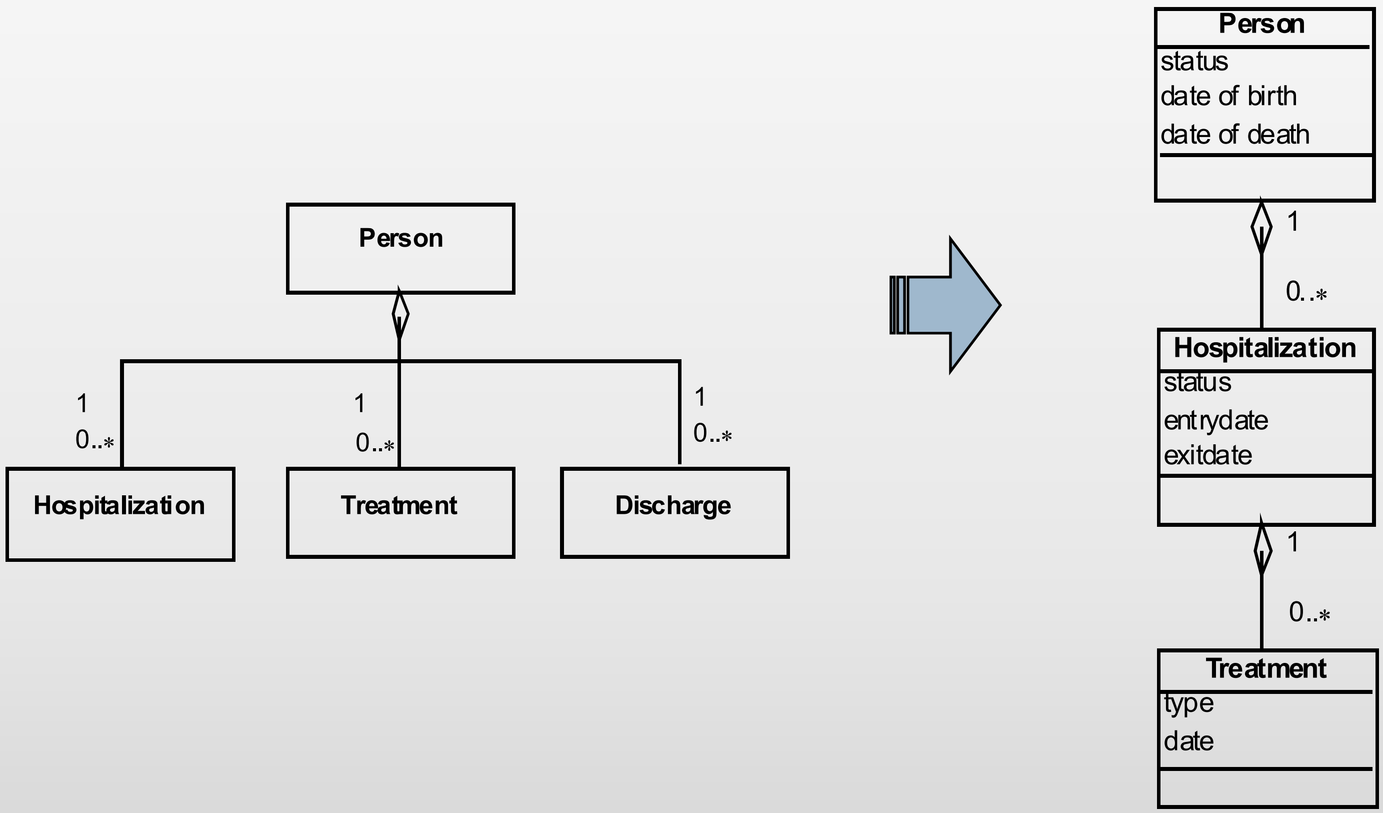
Unnecessary association



How do the statechart diagrams look?

Restructure Classes (3)

Unnecessary classes with embedded iteration



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.

Contents

- Summary of last lecture
- Component design
- The Model Component activity
- Solution to Written exam 2018-01