



Systems Development



Lecture 4: Behaviour

Contents

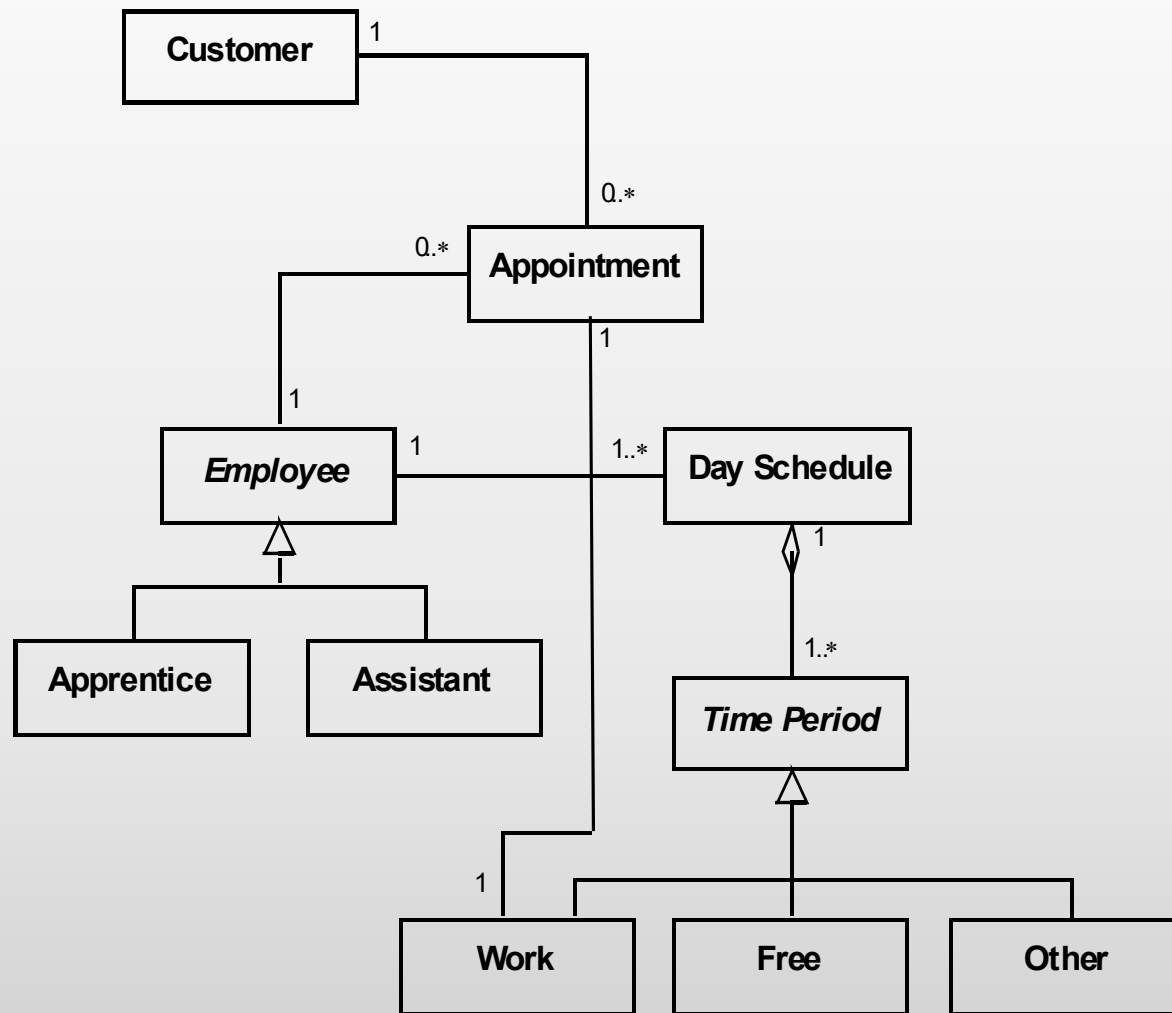
- ▶ Summary of last lecture
- ▶ The Behaviour activity
- ▶ Example: street food and streaming
- ▶ Completing the diagrams
- ▶ Explore patterns
- ▶ Challenges in this activity

Contents

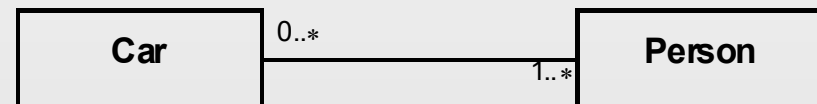
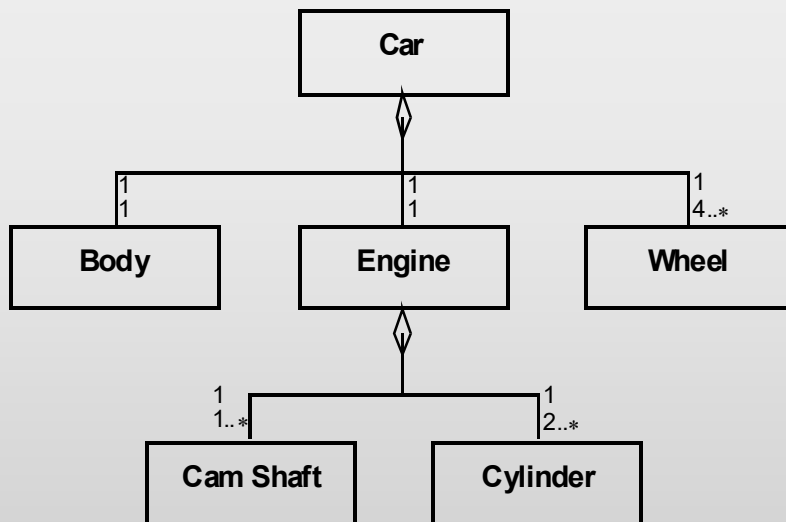
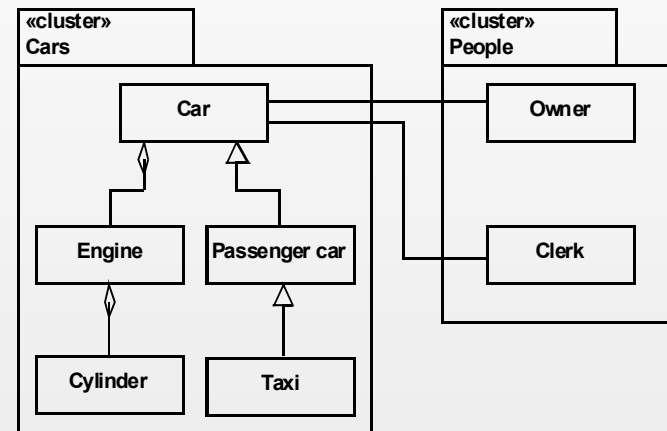
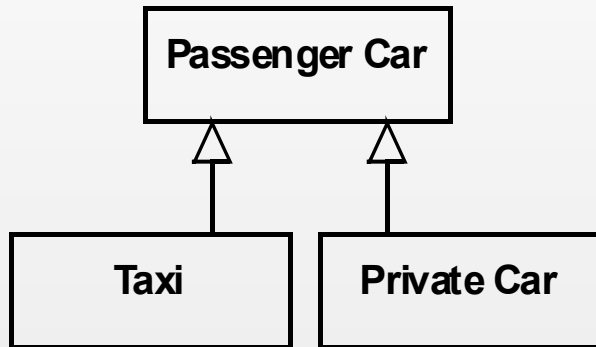
- ▶ Summary of last lecture
 - The Structure activity
- ▶ The Behaviour activity
- ▶ Example: street food and streaming
- ▶ Completing the diagrams
- ▶ Explore patterns
- ▶ Challenges in this activity

Structure: Result

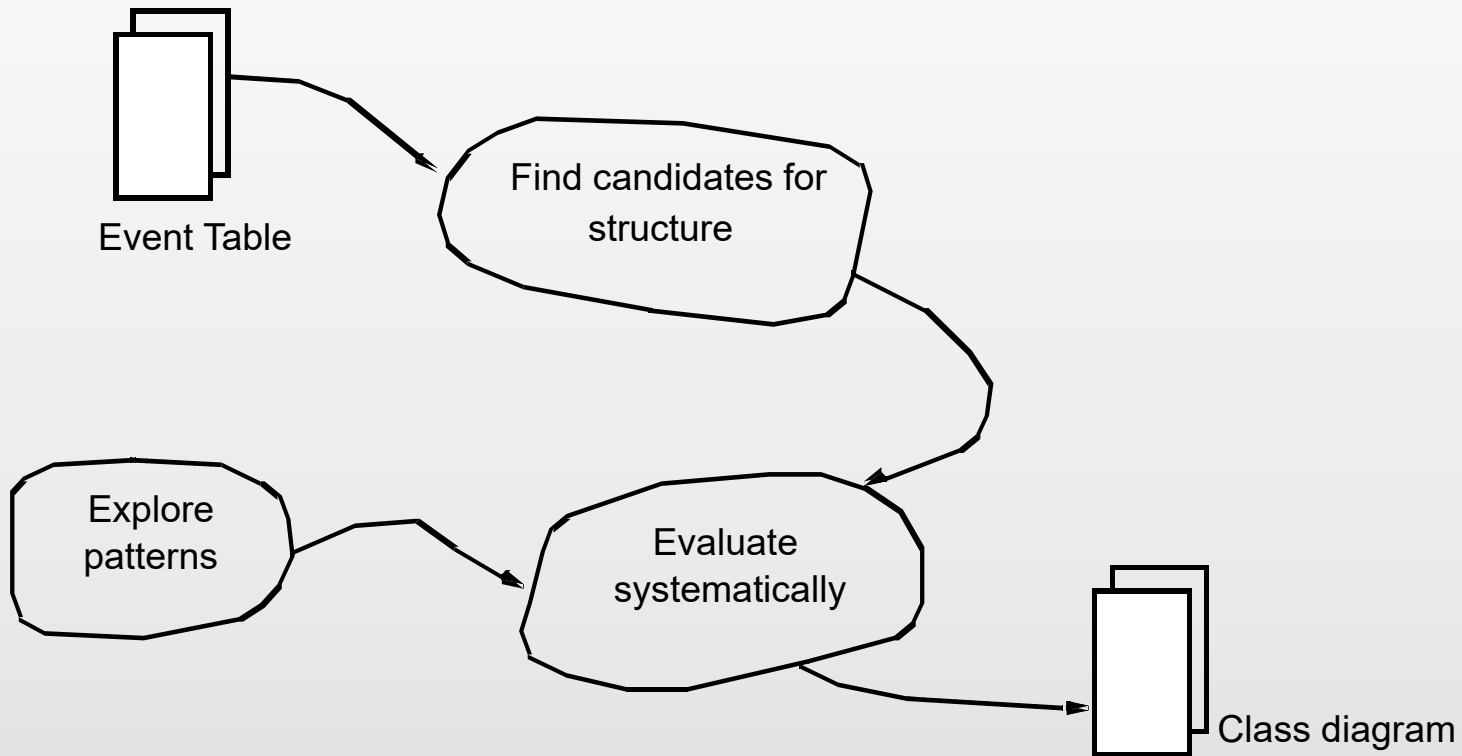
Class
diagram



Structure: Key Concepts



Structure: Activities



Evaluate Systematically

Structures must be used correctly

- Generalization versus aggregation (is-a / has-a)
- Aggregation versus association (page 87)

Structures must be conceptually true

- Names, concepts, and structures reflect the user's understanding
- The prospective user

Structures must be simple

- Especially at the top levels
- Avoid unnecessary generalizations and aggregations
- Avoid objects changing class
- Check against the system definition

Explore Patterns

- ▶ Role
- ▶ Relation
- ▶ Hierarchy
- ▶ Item-Descriptor

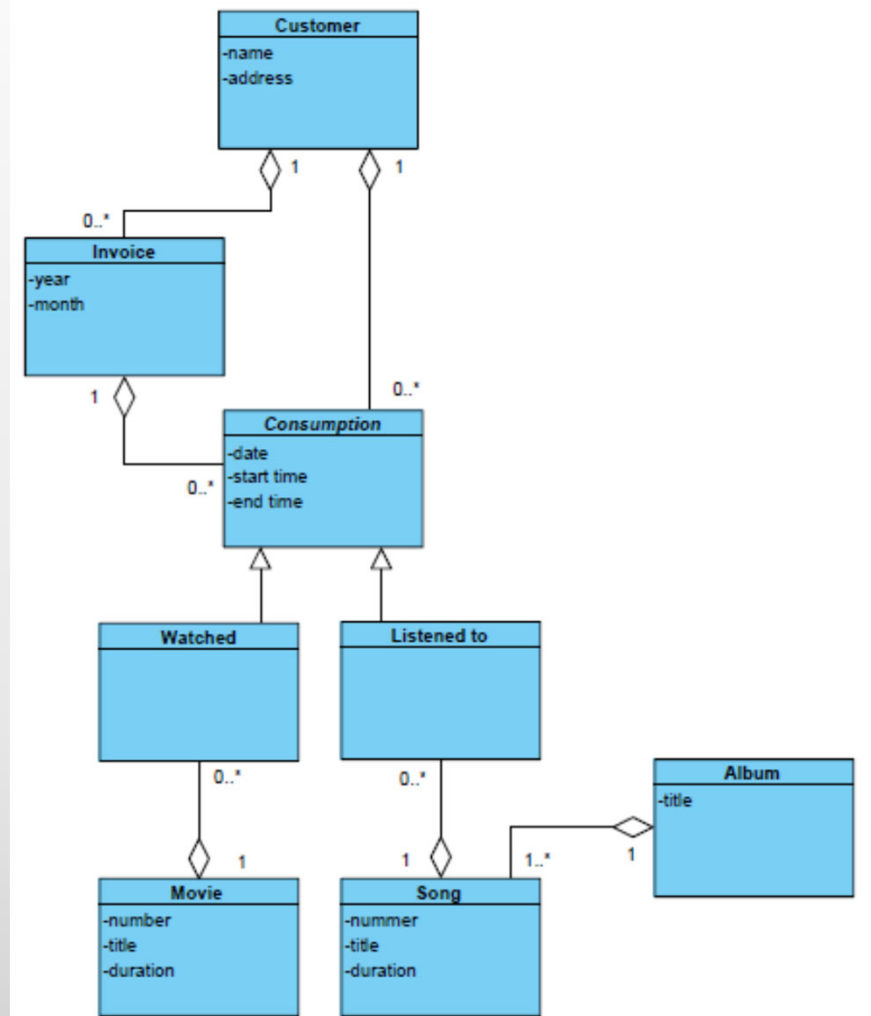
Structure: Summary

Purpose	<ul style="list-style-type: none">• To describe structural relations between classes and objects in a problem domain.
Concepts	<ul style="list-style-type: none">• Class structures• Generalization: A general class (the super class) describes properties common to a group of specialized classes (the subclasses).• Cluster: A collection of related classes.• Object structures• Aggregation: A superior object (the whole) consists of a number of objects (the parts).• Association: A meaningful relation between a number of objects.
Principles	<ul style="list-style-type: none">• Study abstract, static relations between classes.• Study concrete, dynamic relations between objects.• Model only the necessary structural relations.
Results	<ul style="list-style-type: none">• A class diagram with classes and structures.

Streaming Service Class Diagram

► System definition:

- F: register the movies the customers see and the songs they listen to, and support payment by customers for their consumption of movies and songs.
- A: will be used by the administrative personnel that is employed by the organization that provides the streaming service.
- C: developed for the administrative personnel.
- T: PC platform with typical tools.
- O: Customer, Movie, Song.
- R: registration, administration and payment of customers' consumption.



Quiz 3 Overview

Quiz 3

Average

4.56 (of 6.00) of 111 finished attempts (of 159)

Best result (0.67-1.00)

- 2 (0.95) Match the three structures with the diagrams:
- 3 (0.78) Generalization is a structure that expresses that:
- 4 (0.75) The following is/are object structure(s):
- 6 (0.74) The following is/are class structure(s):
- 1 (0.74) Which pattern is this?

Middle result (0.34-0.66)

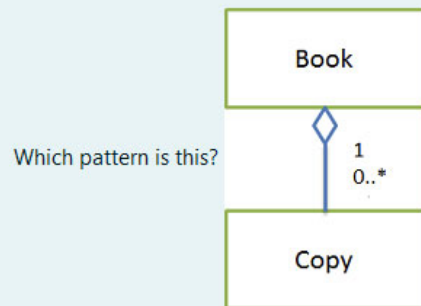
- 5 (0.60) The Role Pattern

Worst result (0.00-0.33)

None

Quiz 3 - Answers

► Which pattern is this?



Select one:

- ☐ a. Relation pattern
- ☐ b. Role pattern
- ☒ c. Descriptor-item pattern
- ☐ d. Hierarchy Pattern

[Clear my choice](#)

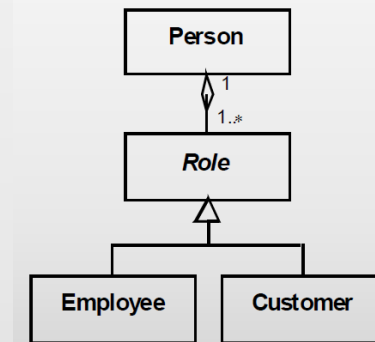
► Some marked Hierarchy, Role and Relation

► The role pattern

The Role Pattern

Select one or more:

- ☒ a. Is a pattern concerned with structure
- ☐ b. Is a pattern concerned with behavior
- ☐ c. Relate two parties to each other
- ☒ d. Let's a person change roles dynamically
- ☐ e. Describes a whole and an item

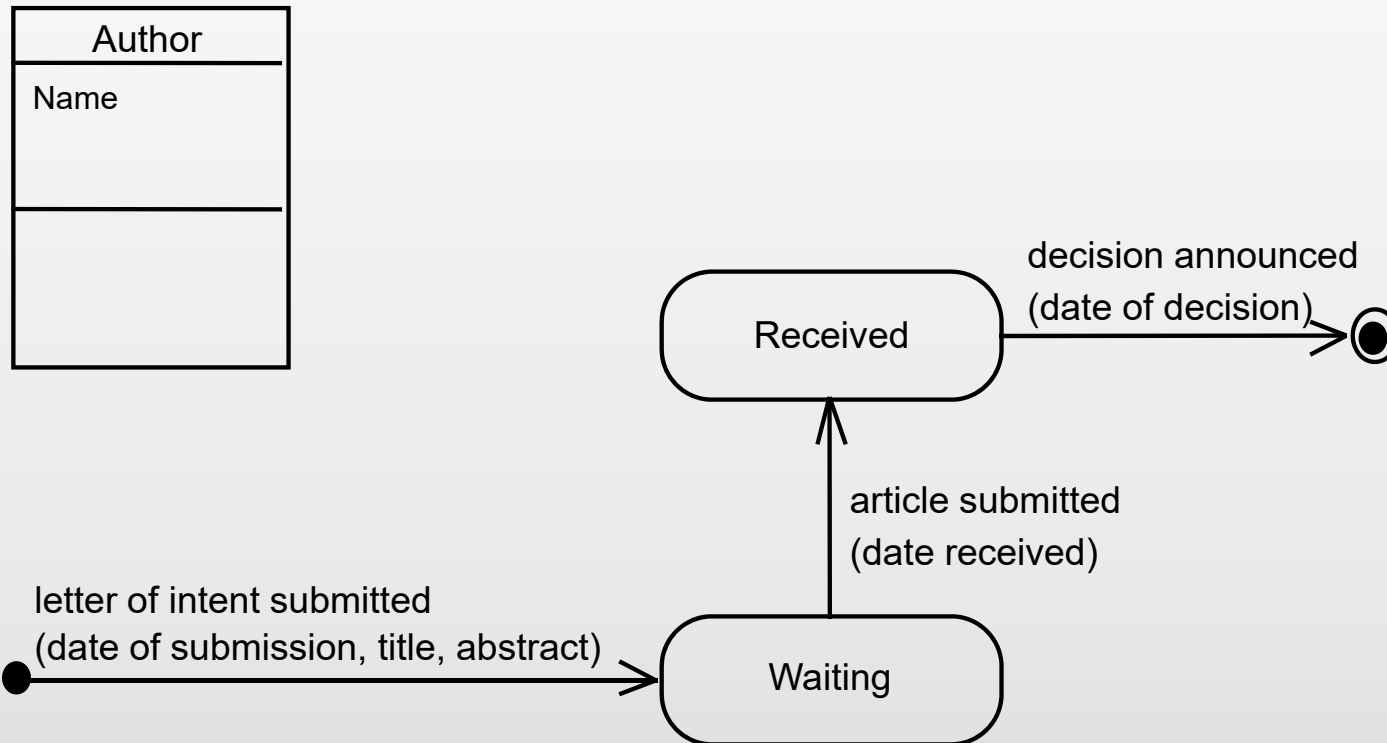


► Some marked “concerned with behavior” and “relate two parties to each other”

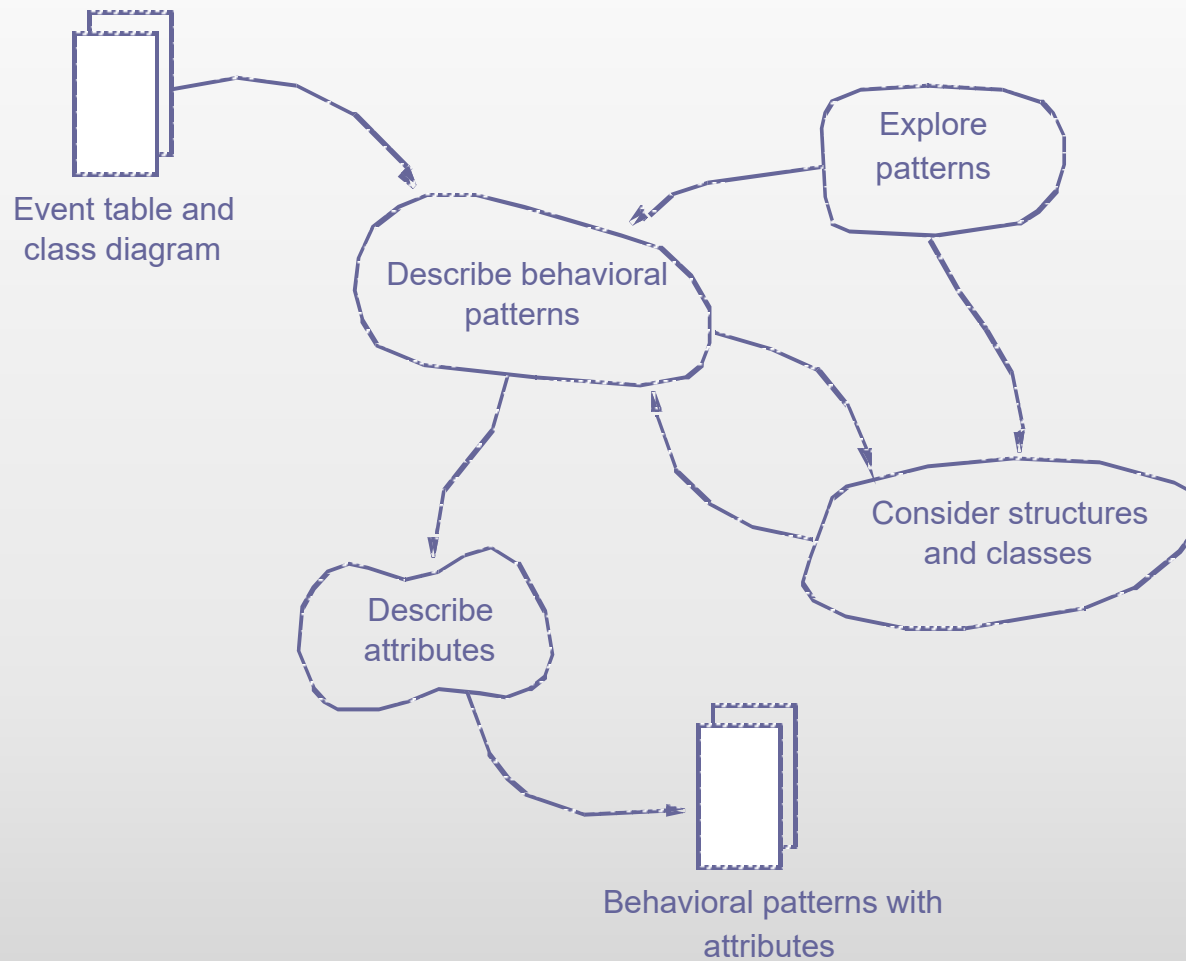
Contents

- ▶ Summary of last lecture
- ▶ The Behaviour activity
 - Result
 - Activities
 - Key concepts
 - Modelling techniques
- ▶ Example: street food and streaming
- ▶ Completing the diagrams
- ▶ Explore patterns
- ▶ Challenges in this activity

Behaviour: Result



Behaviour:Activities



Key Concepts: Event Traces

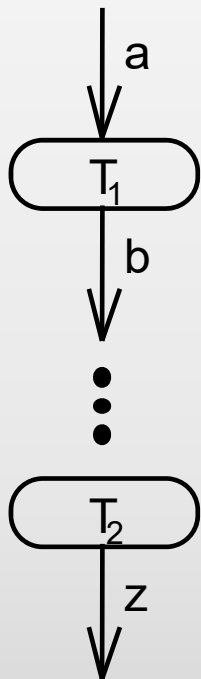
- ▶ A sequence of events involving a specific object
- ▶ Account-1: opened-closed
- ▶ Account-2: opened-deposited-withdrawn-deposited-deposited-deposited-....
- ▶ Account-3: opened-deposited-withdrawn-withdrawn-...
- ▶ Account-n: ...

Describe Behavioral Patterns

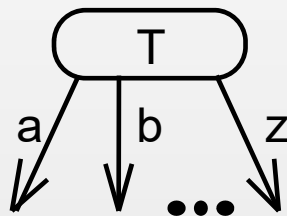
- ▶ Make event traces for each class
- ▶ For each class ask:
 - Which event(s) cause the creation of a problem-domain object?
These events are grouped as selections that can cause the birth of an object.
 - Which event(s) cause the disappearance of a problem-domain object?
These events are grouped as selections that can cause the death of an object.
- ▶ Typical event traces:
 - Which events occur together in a sequence?
 - Are there any alternative events?
 - Can a given event occur more than once?
 - Is the overall form structured or unstructured?

Key Concepts: Control Structures

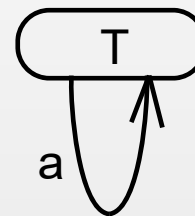
Sequence



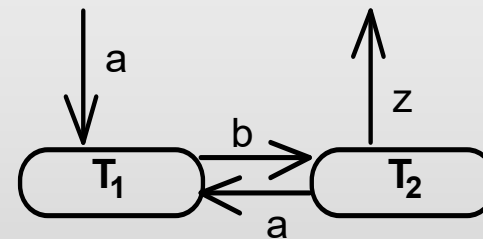
Selection



Iteration

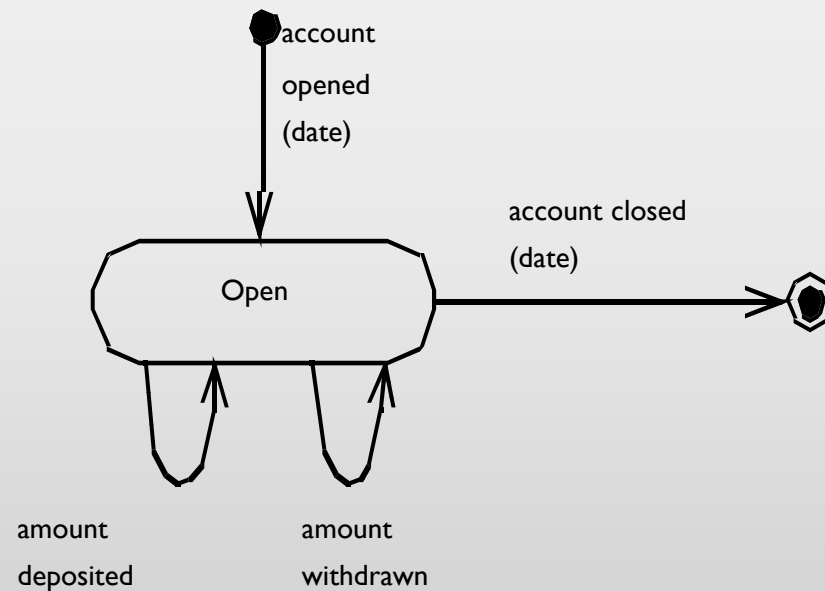
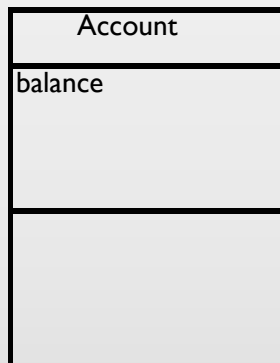


Also iteration

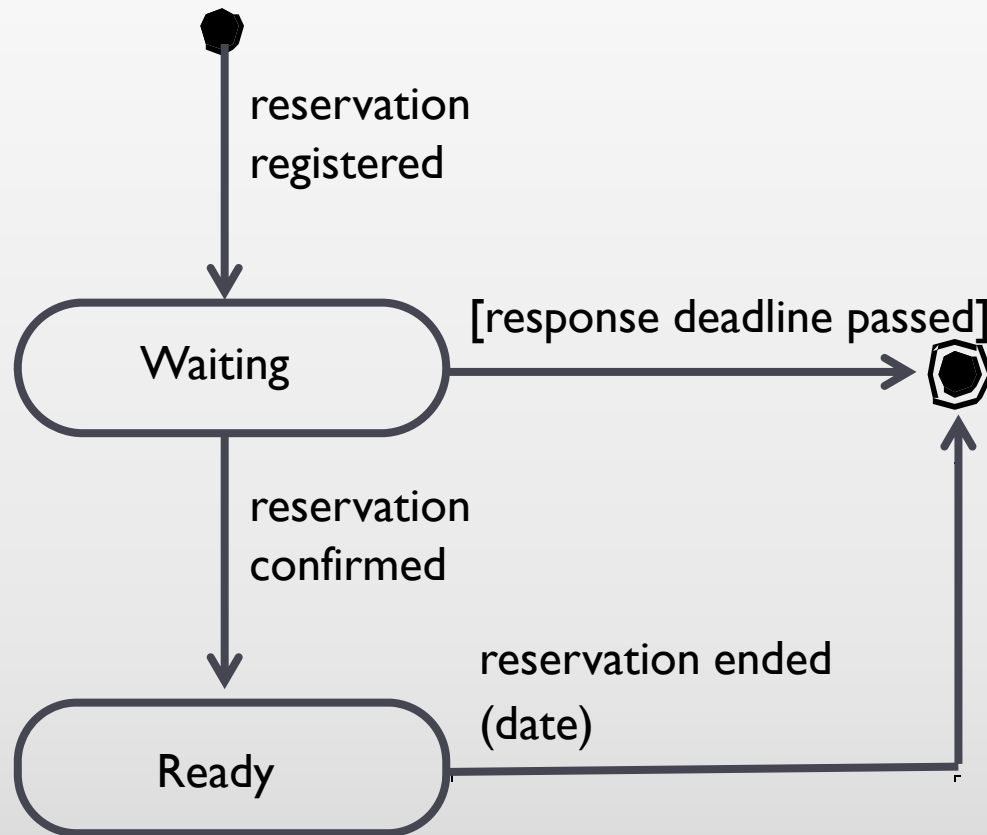


From Event Traces to Behavioural Patterns

- ▶ Account-1: opened-closed
- ▶ Account-2: opened-deposited-withdrawn-deposited-deposited-...-closed
- ▶ Account-3: opened-deposited-withdrawn-withdrawn-...-closed
- ▶ Account-n: ...



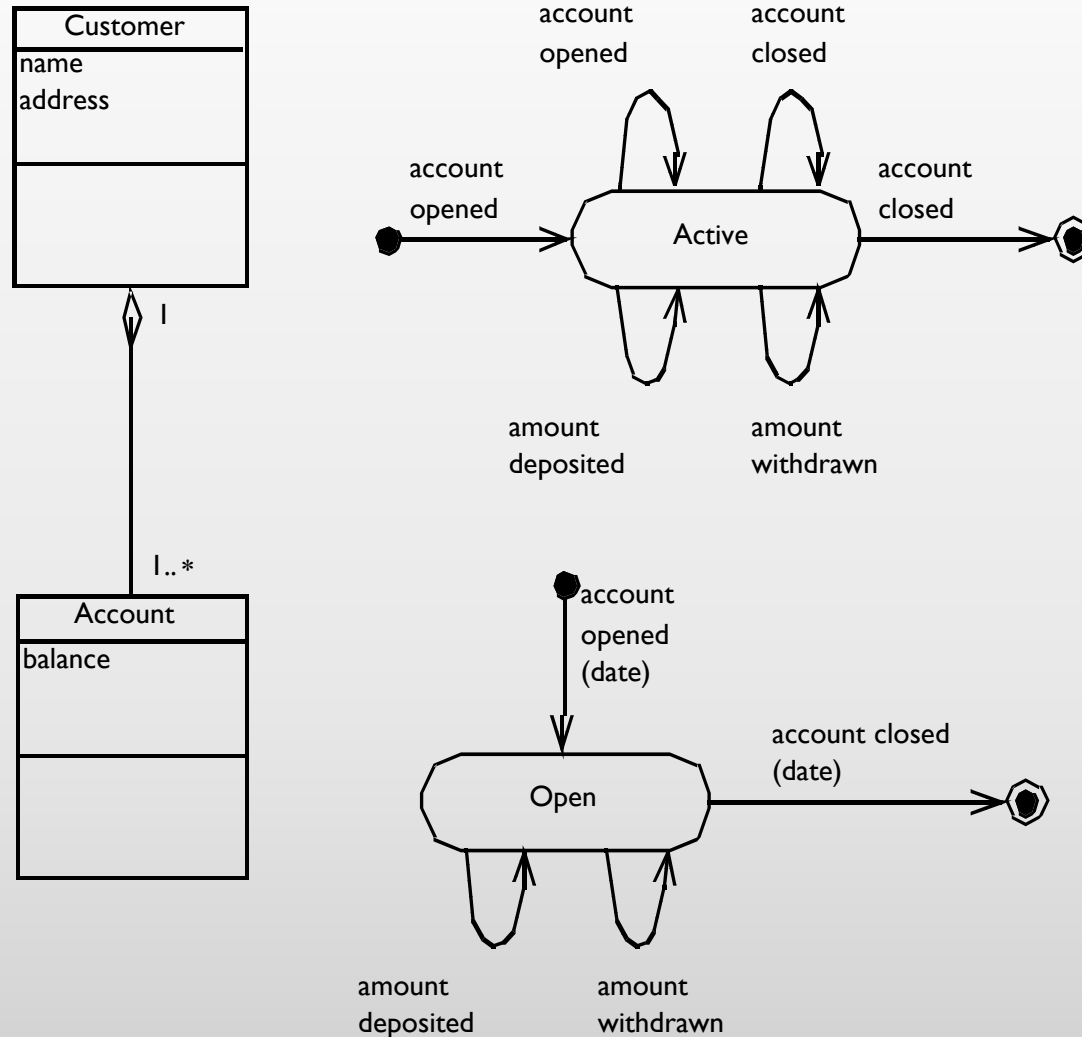
Conditions in Statechart Diagrams



Common Events: Event Table

	Customer	Assistant	Apprentice	Appointment	Plan
reserved	*	*		+	*
cancelled	*	*		+	
treated	*			+	
employed		+	+		
resigned		+	+		
graduated			+		
agreed		*	*		*

Common Events: State Chart Diagrams



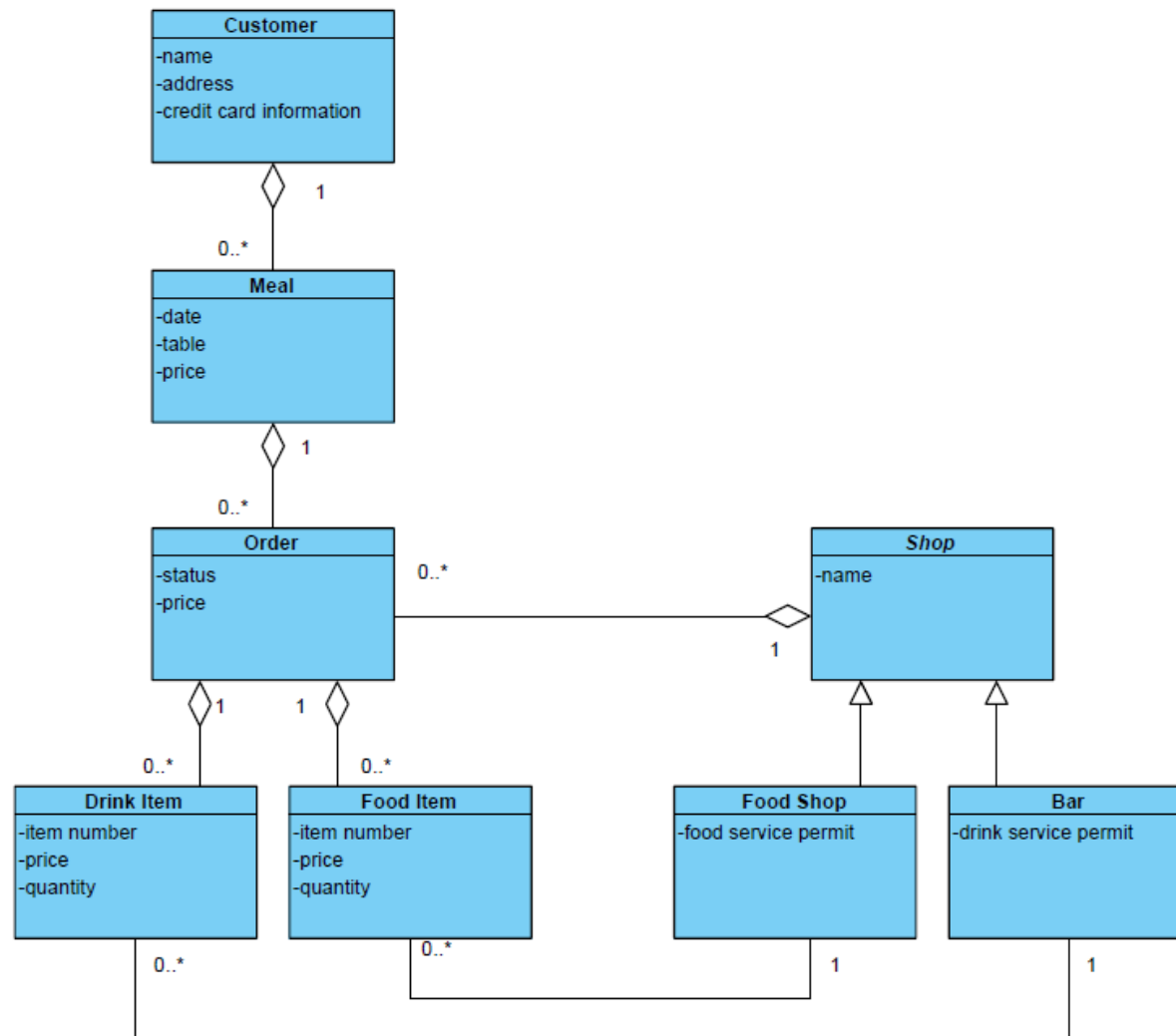
Contents

- ▶ Summary of last lecture
- ▶ The Behaviour activity
- ▶ Example: street food and streaming
- ▶ Completing the diagrams
- ▶ Explore patterns
- ▶ Challenges in this activity

Street Food

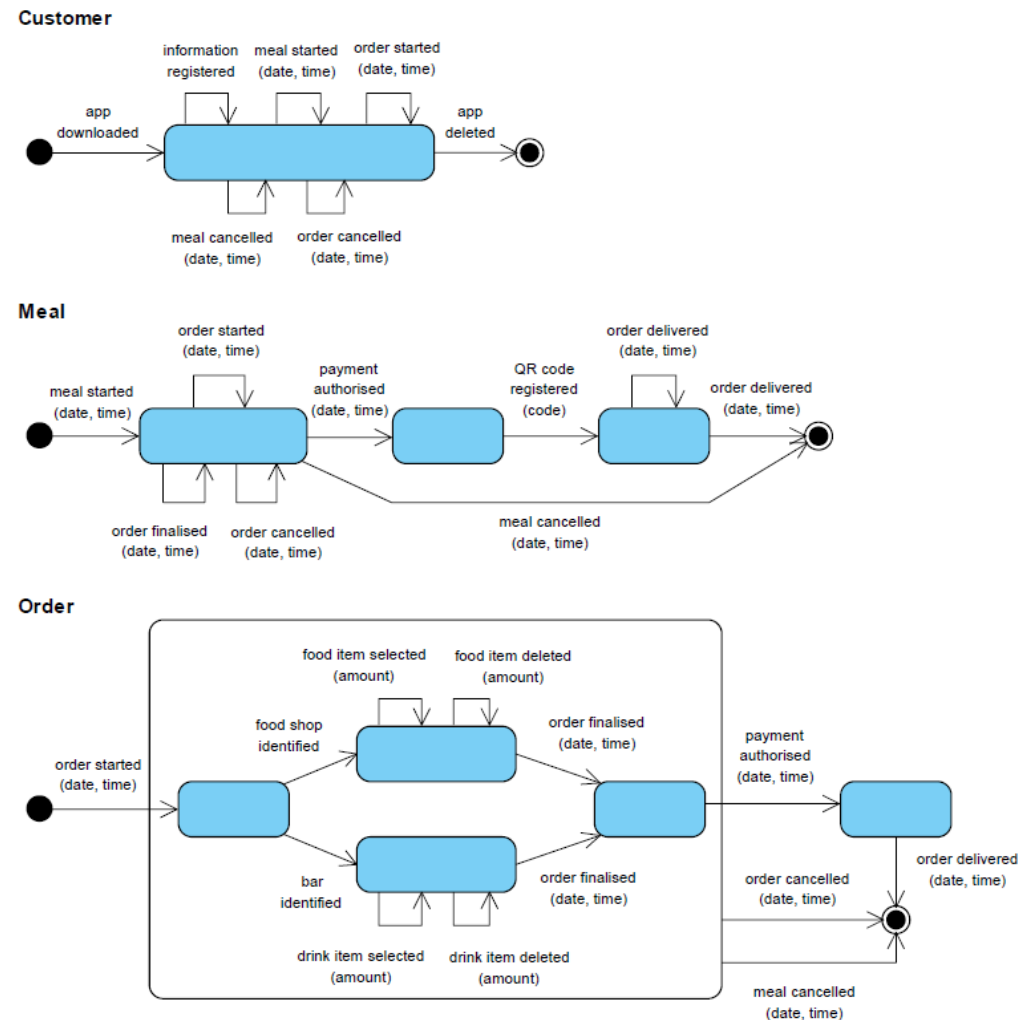
- F: The system can register a new customer with credit card information, compose a meal with a number of orders for food shops and bars, select food or drink items in each order of a meal, make payment of a meal, and register the table where a customer wants his/her orders delivered.
- A: A system provided as an app, which is used by customers who want to order food and drinks at S-Food's street food market. The customers come to S-Food at their own initiative, and their only relation to S-Food is that they download and use their app to order and pay for food and drinks.
- C: The app will be developed by S-Food's own IT department in cooperation with S-Food's sales department, the food shops and bars, and customers that will be selected to represent the whole customer segment. It may be necessary to resolve conflicting requirements between these different groups. The app will be used by users with very different levels of IT skills.
- T: The app is running on each customer's smartphone. It communicates through a wireless network with a server that registers what the individual customer has ordered and paid. On the smartphone, there is always a copy of all meals from the current day, so they are available if the wireless network should fail. The app includes a QR code reader.
- O: Customer, Food Shop, Bar, Order, Meal, Food item, Drink item.
- R: The system is primarily an administrative tool that is responsible for registering all customers and their orders, and facilitating secure payment of these with the customers' credit cards. Secondly, it is a communication medium that customers use to request delivery of orders from the food shops and bars.

Street Food Class Diagram



Statechart Diagrams

- ▶ Make statechart diagrams for the following classes:
- ▶ Customer
- ▶ Meal
- ▶ Order

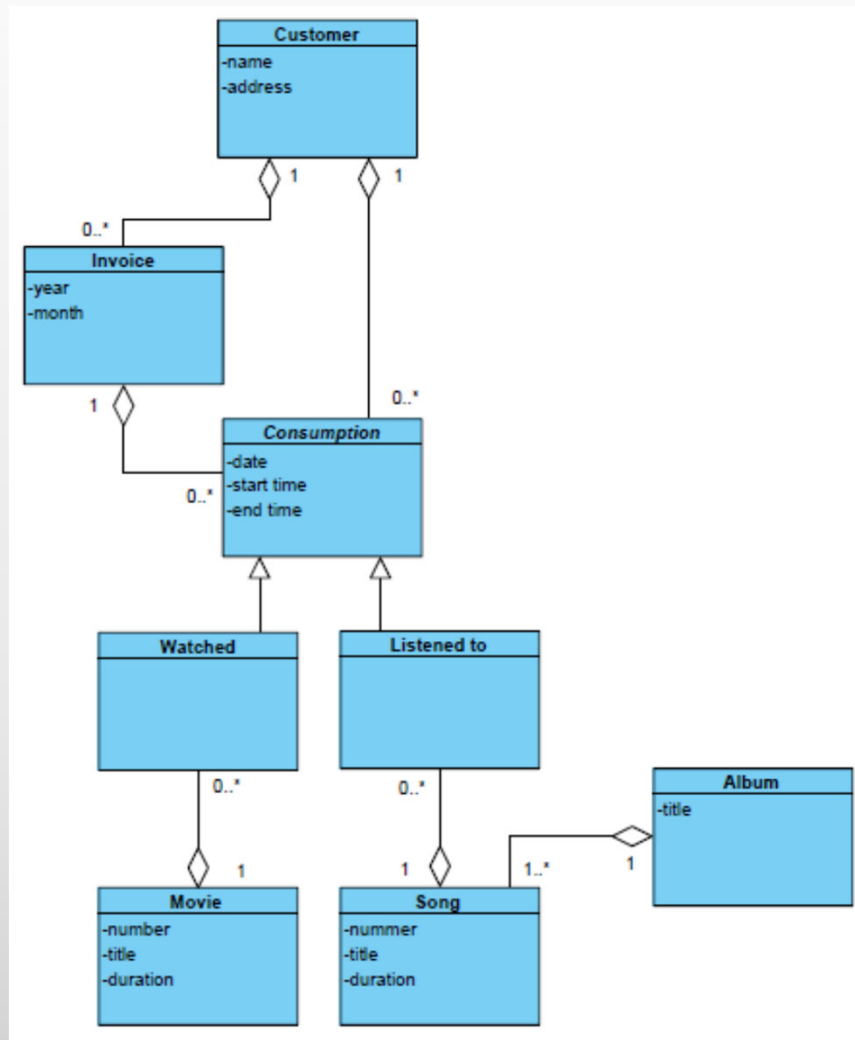


A Streaming Service

► System definition:

- F: register the movies the customers see and the records they listen to, and support payment by customers for their consumption of movies and songs.
- A: will be used by the administrative personnel that is employed by the organization that provides the streaming service.
- C: developed for the administrative personnel.
- T: PC platform with typical tools.
- O: Customer, Movie, Song.
- R: registration, administration and payment of customers' consumption.

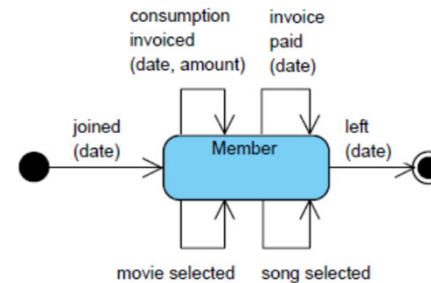
Class Diagram for the Streaming Service



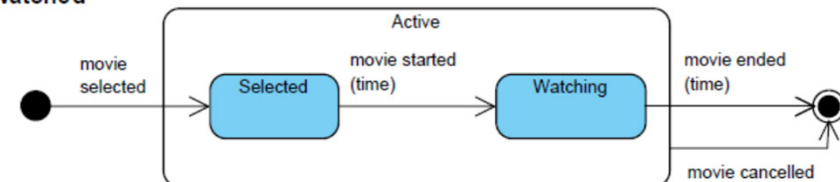
Statechart Diagrams

- ▶ Make statechart diagrams for the following classes:
- ▶ Customer
- ▶ Movie
- ▶ Watched

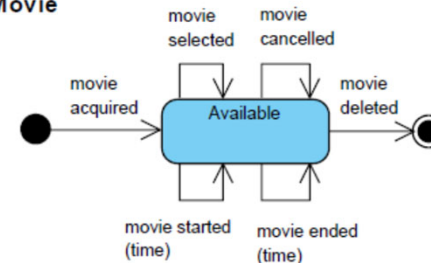
Customer



Watched



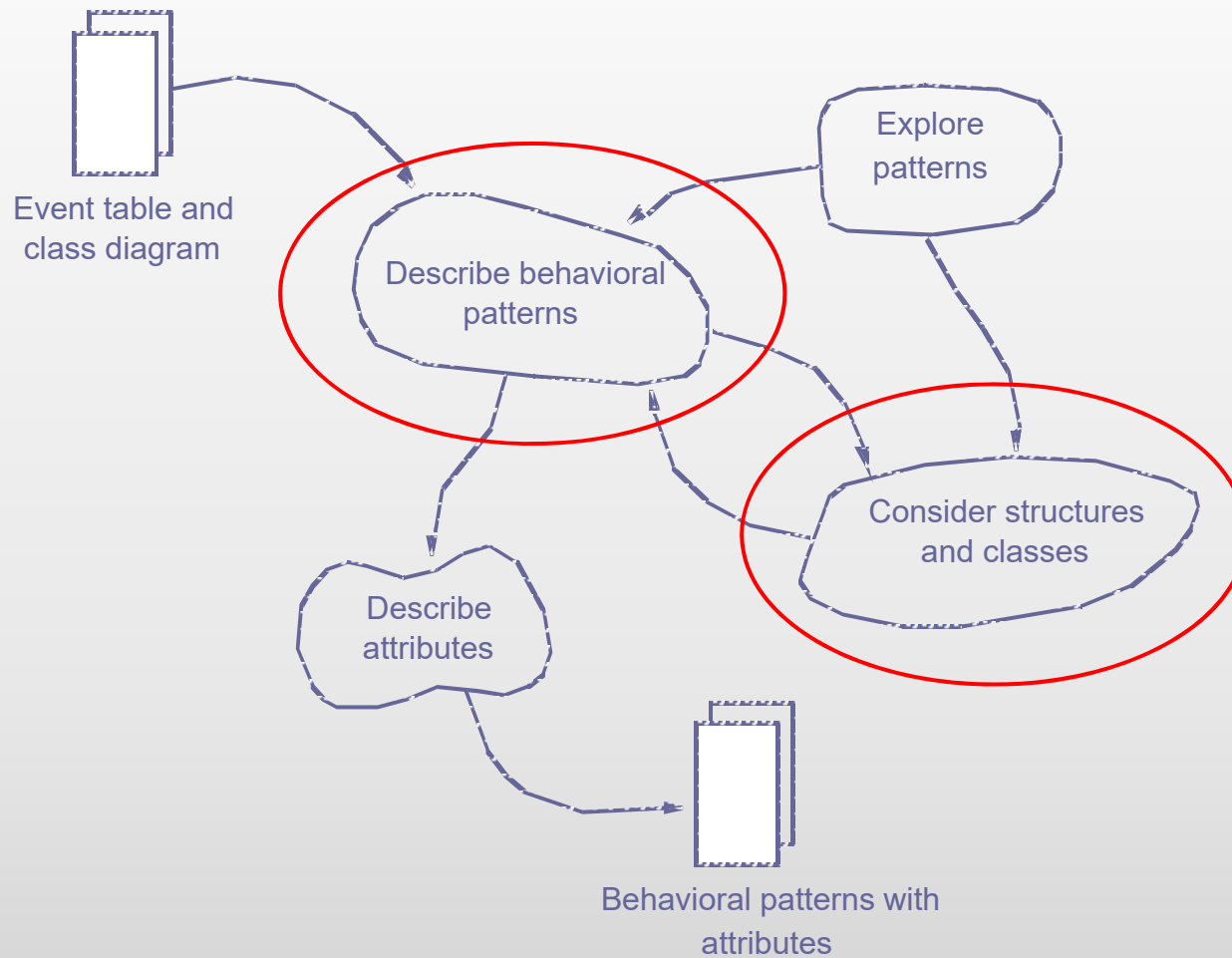
Movie



Contents

- ▶ Summary of last lecture
- ▶ The Behaviour activity
- ▶ Example: street food and streaming
- ▶ Completing the diagrams
- ▶ Explore patterns
- ▶ Challenges in this activity

Behaviour:Activities



Consider Structures

▶ Aggregation and association

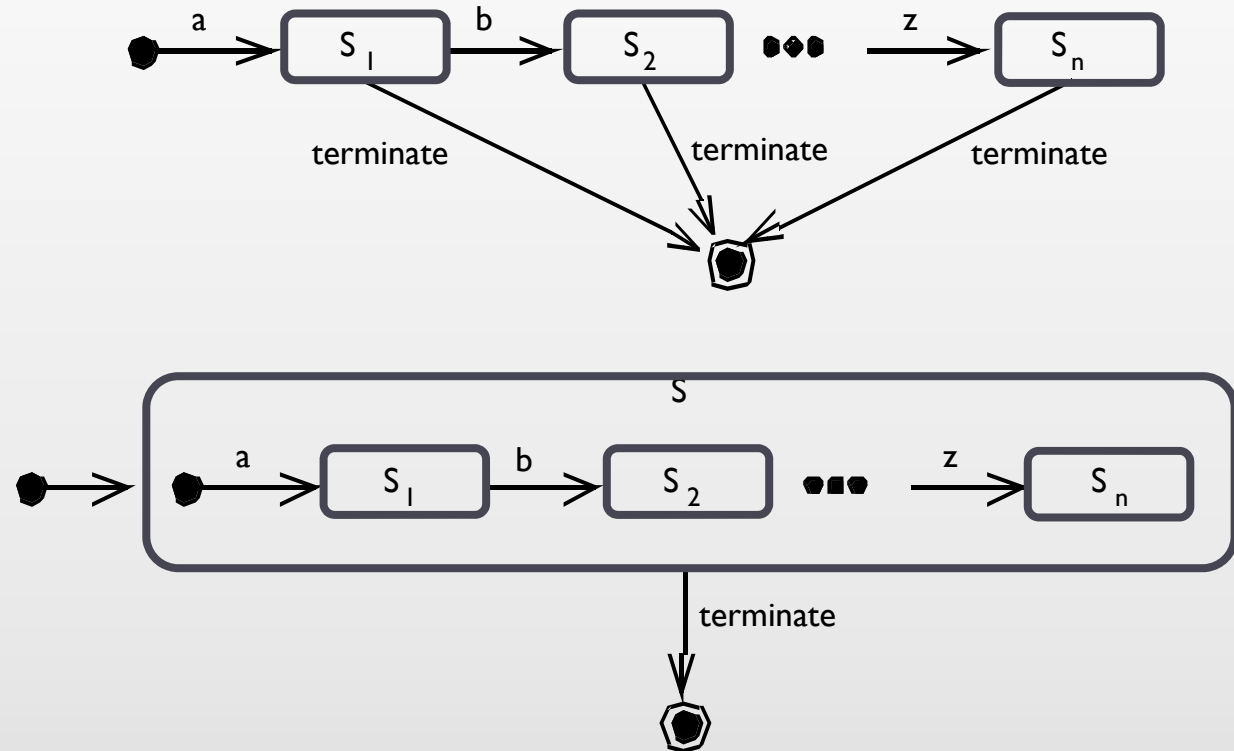
- If two or more objects have common events, consider adding an aggregation or association structure between them.
- If two classes are related by an aggregation or association structure, at least one common event should be considered.

▶ Generalization

- If the same event is tied to two classes, consider whether one class is a generalization of the other.

Hierarchical States

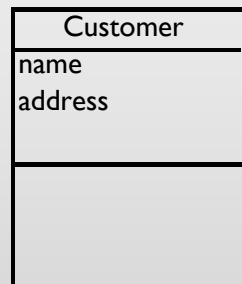
- Consider using hierarchical states



Describe Attributes

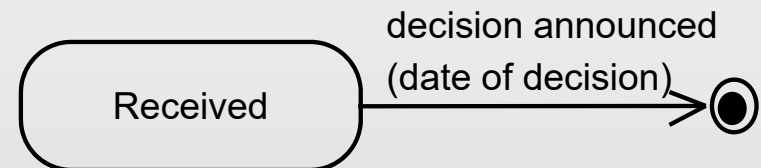
▶ Class attributes:

- What are the general characteristics of the class?
- What basic data must be captured about objects from this class?
- What results from an event trace must be captured?



▶ Event attributes:

- What time did the event occur?
- Which amount did it concern?



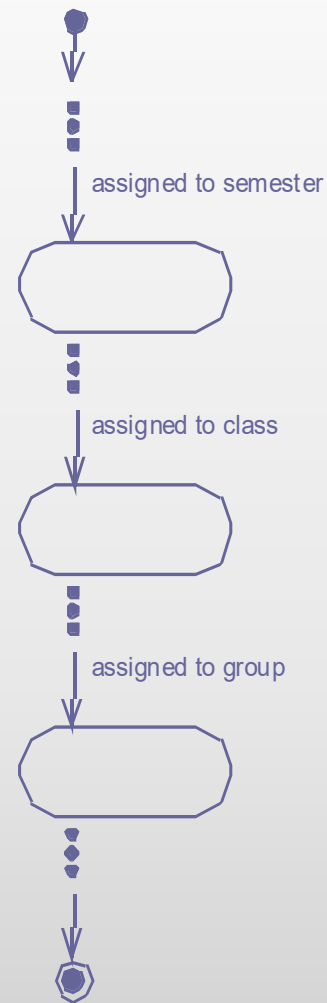
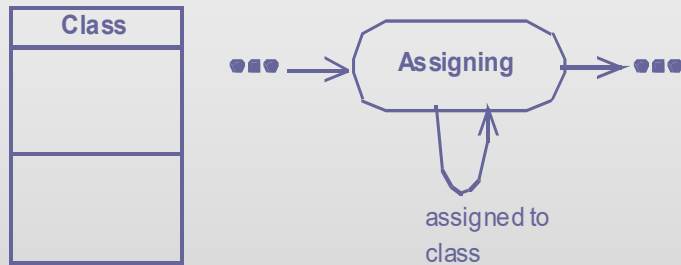
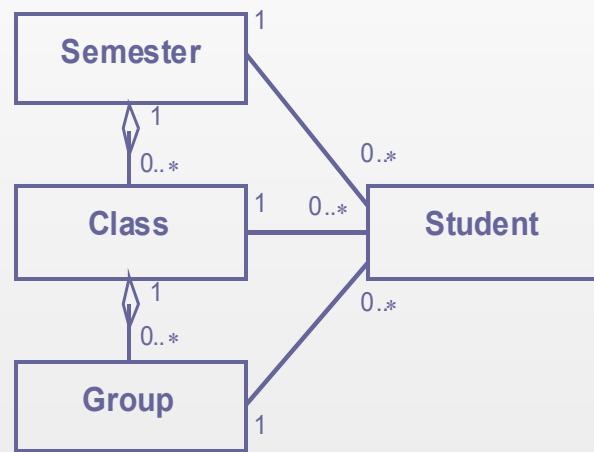
Behaviour: Summary

Purpose	<ul style="list-style-type: none">• To model the dynamics of a problem domain.
Concepts	<ul style="list-style-type: none">• Event trace: A sequence of events involving a specific object.• Behavioral pattern: A description of possible event traces for all objects in a class.• Attribute: A descriptive property of a class or an event.
Principles	<ul style="list-style-type: none">• Create behavioral patterns from event traces.• Study common events.• Derive class attributes from behavioral patterns.
Results	<ul style="list-style-type: none">• A behavioral pattern with attributes for every class in a class diagram

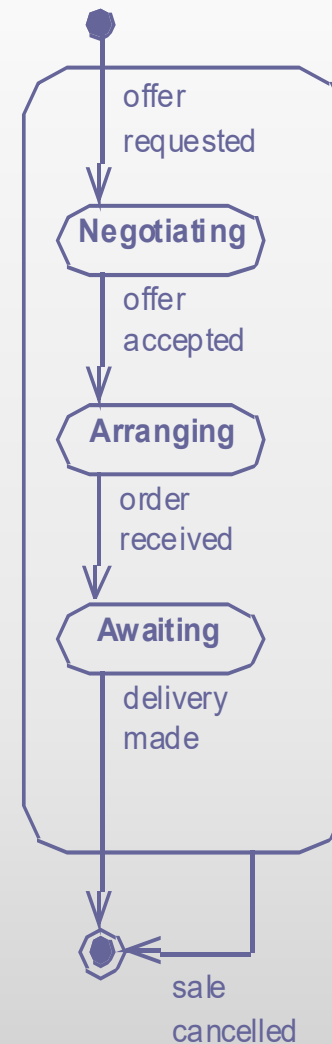
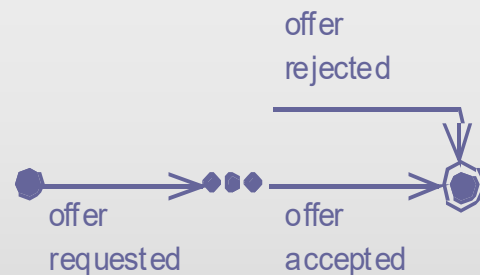
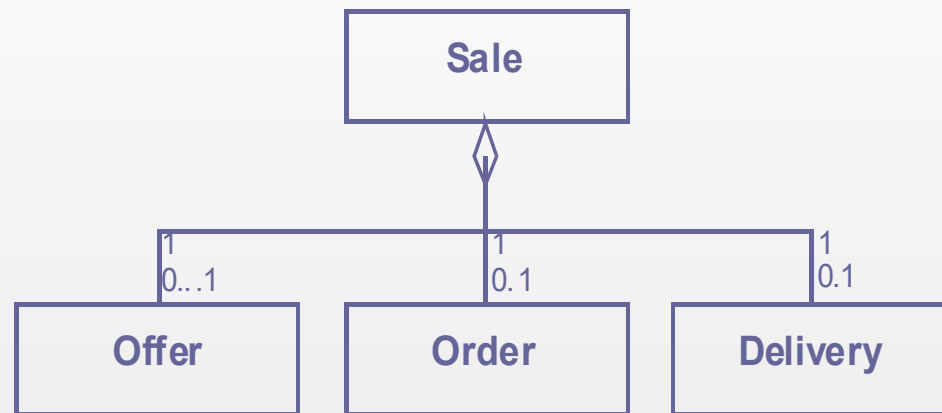
Contents

- ▶ Summary of last lecture
- ▶ The Behaviour activity
- ▶ Example: street food and streaming
- ▶ Completing the diagrams
- ▶ Explore patterns
- ▶ Challenges in this activity

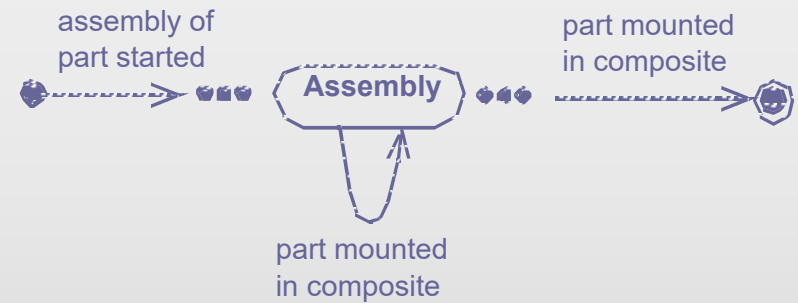
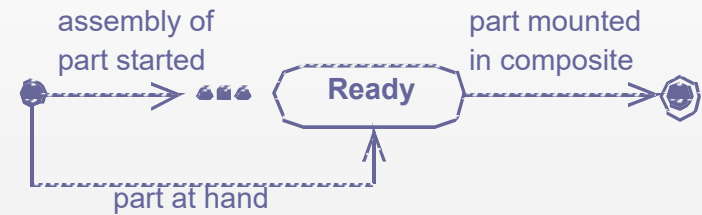
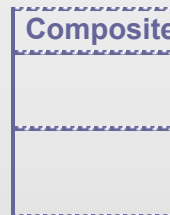
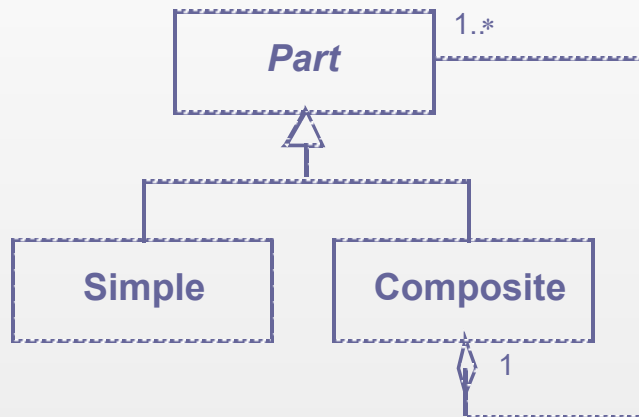
Explore Patterns: Stepwise Relation



Explore Patterns: Stepwise Role



Explore Patterns: Composite



Contents

- ▶ Summary of last lecture
- ▶ The Behaviour activity
- ▶ Example: street food and streaming
- ▶ Completing the diagrams
- ▶ Explore patterns
- ▶ Challenges in this activity

Conceptual: Class, Event or Attribute

- ▶ We have some information from the problem domain
 - Example: a customer buys a chair for 100 kr
- ▶ How can we model that?
 - Attribute:
In the Customer class we introduce an attribute 'bought for' which is increased by the amount that the customer has bought for
 - Event:
For the Customer class we introduce a 'bought' event with the prize of the chair as an attribute
What do we gain here?
We can produce a list of the amounts the user has bought for
 - Class/Object:
We introduce a 'Bought' class that is aggregated by the Customer class, it aggregates an object of the Product class and has some events
What do we gain here?
We can model events and behavioural structure on Bought objects (e.g. complaint)

Work in Exercises for this Activity

- ▶ Start with event traces for the simple classes
- ▶ Describe behavioural patterns from the event traces
- ▶ Continue with the more complex classes
- ▶ If the behavioural pattern becomes too complicated, consider using the stepwise role or stepwise relation pattern – this introduces new classes
- ▶ Make sure there are behavioural patterns that control sequence and some that don't (structured/unstructured)
- ▶ Add attributes to classes and events
- ▶ Check the behavioural patterns against the class diagram