



Systems Development



Lecture 3: Structure

Contents

- ▶ Hand-in assignment
- ▶ Summary of last lecture
- ▶ The Structure activity
- ▶ Example: street food
- ▶ Explore patterns
- ▶ Challenges in this activity

Contents

- ▶ Hand-in assignment
- ▶ Summary of last lecture
- ▶ The Structure activity
- ▶ Example: street food
- ▶ Explore patterns
- ▶ Challenges in this activity

Hand-In Assignment

- ▶ An opportunity to make analysis and design yourself
- ▶ Check your understanding of the first activities
- ▶ Do it individually and then in groups (your choice)
 - If you solve it in a group, hand in only one solution

Contents

- ▶ Hand-in assignment
- ▶ Summary of last lecture
 - Problem domain analysis
 - The Classes activity
- ▶ The Structure activity
- ▶ Example: street food
- ▶ Explore patterns
- ▶ Challenges in this activity

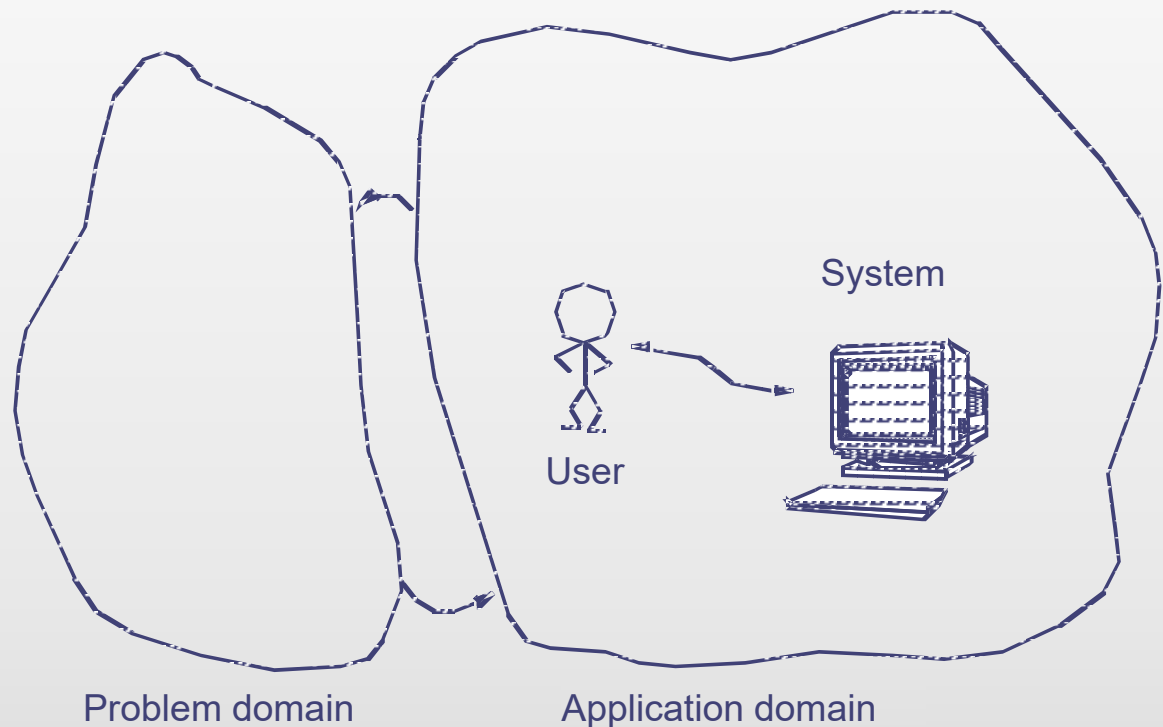
Problem Domain Analysis: Key Concepts

Problem domain:

That part of a context that is administrated, monitored, or controlled by a system

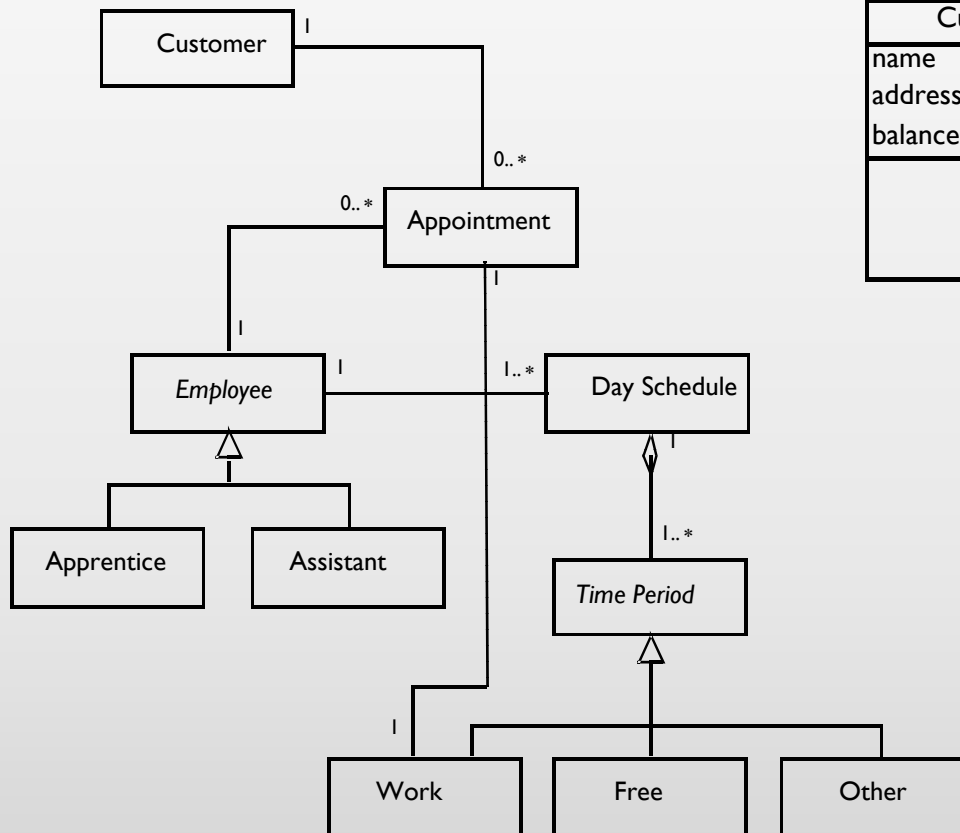
Examples of problem domains:

- ▶ Students in an education institution
- ▶ Items in a warehouse
- ▶ Sand in a gravel pit
- ▶ Customers in a hair salon
- ▶ Computers and employees in a university
- ▶ Customers and meals in a street food place

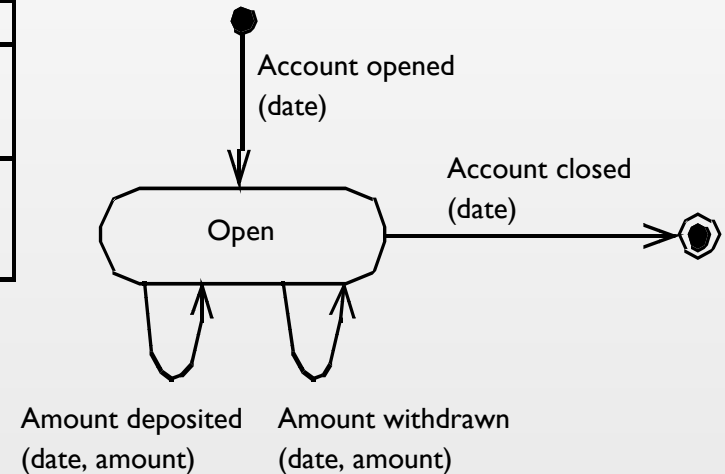
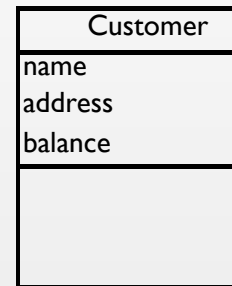


Problem Domain Analysis: Results

Class diagram



Behavioural pattern for each class



Classes: Result

Event table

	reserved	cancel	treated	employed	resigned	grad.	agreed
Customer	✓	✓	✓				
Assistant	✓	✓		✓	✓		✓
Apprentice				✓	✓	✓	✓
Appointment	✓	✓	✓				
Plan	✓						✓

Shows major classes and events in the problem domain

The classes are usually listed horizontally

Key Concepts: Event

- ▶ Event:
An instantaneous incident involving one or more objects
- ▶ Atomic
- ▶ Instantaneous
- ▶ Common to several objects
- ▶ Unique name
- ▶ Example from IKEA – Events:
 - Item received from producer
 - Item stored in warehouse
 - Item selected by customer
 - Item picked in warehouse – OK?
 - Item sold



Classes: Summary

Purpose	<ul style="list-style-type: none">• To select the elements of a problem-domain model.
Concepts	<ul style="list-style-type: none">• Object: An entity with identity, state, and behavior.• Class: A description of a collection of objects sharing structure, behavioral pattern, and attributes.• Event: An instantaneous incident involving one or more objects.
Principles	<ul style="list-style-type: none">• Classify objects in the problem domain.• Characterize objects through their events.• Have an open mind, but select critically.
Results	<ul style="list-style-type: none">• An event table with classes and related events.

Quiz 2: Overview

Quiz 2

Average

4.79 (of 7.00) of 87 finished attempts (of 175)

Best result (0.67-1.00)

- 3 (0.93) An object in the problem domain is defined as an entity with
- 2 (0.81) A system to be used to control train traffic at Aalborg Railway Station: Which of the following are events?
- 6 (0.77) An event in problem domain analysis is:
- 7 (0.70) The system's model of the problem domain (its purpose)

Middle result (0.34-0.66)

- 1 (0.66) The following are core principles of the class activity
- 5 (0.56) What are the results of the problem domain analysis?
- 4 (0.37) A system to be used to control train traffic at Aalborg Railway Station: Which statements are correct? (PD and AD)

Worst result (0.00-0.33)

None

Quiz 2: Answers

▶ Question 4

A system to be used to control train traffic at Aalborg Railway Station: Which statements are correct?

Select one or more:

- ☐ a. A train sensor on the rails is part of the AD
- ☒ b. A train sensor on the rails may be part of the PD
- ☒ c. A signal post and its signal can be part of both PD and AD, but it depends on which functions we require
- ☒ d. A train sensor on the rails is not an object and neither part of the PD nor part of the AD because it is just an input device
- ☐ e. A signal post and its signal cannot be part of PD and AD at the same time
- ☒ f. A signal post and its signal can be part of the PD
- ☒ g. A signal post and its signal is part of the AD

▶ Input device:

- is often only that
- may be in the PD if it is monitored

▶ Output device:

- is always in the AD (with the system and users)
- may be in the PD if it is monitored

▶ Question 5

What are the results of the problem-domain analysis?

Select one or more:

- ☐ a. An actor table
- ☐ b. State-chart diagrams describing use cases
- ☒ c. A class diagram describing classes and structure
- ☒ d. State-chart diagrams describing the behavior of objects in classes
- ☐ e. A component architecture
- ☒ f. An event table

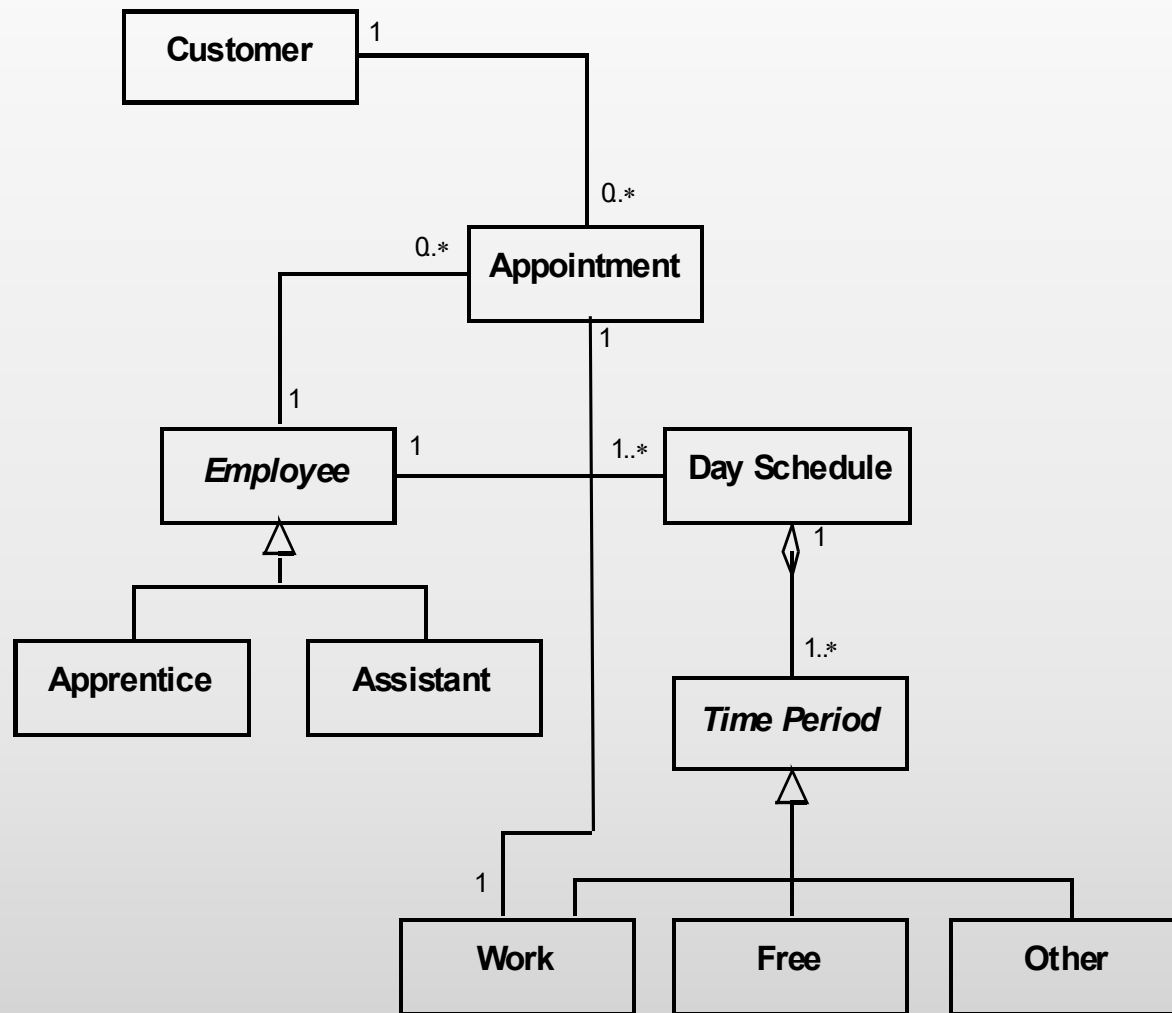
- ▶ Most checked only one or two of the three correct answers

Contents

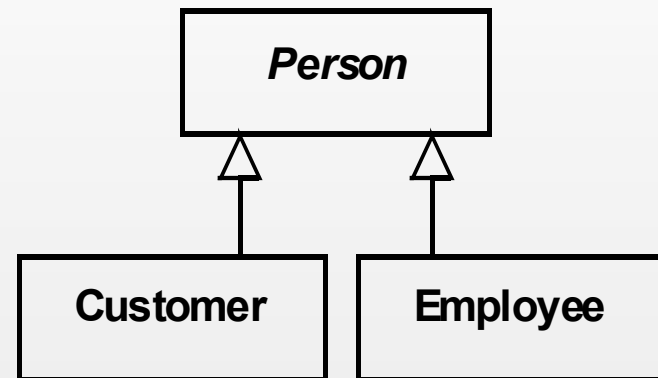
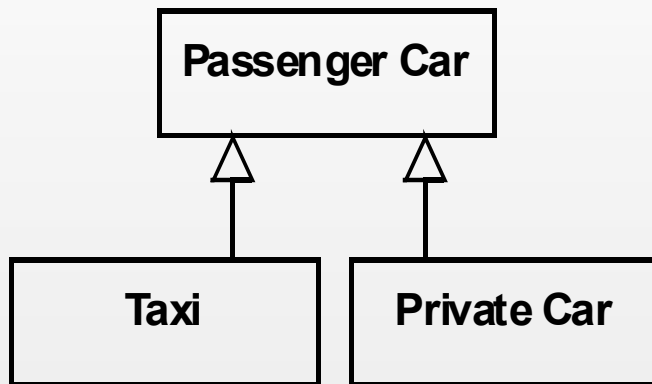
- ▶ Hand-in assignment
- ▶ Summary of last lecture
- ▶ The Structure activity
 - Result
 - Key concepts
 - Activities
- ▶ Example: street food
- ▶ Explore patterns
- ▶ Challenges in this activity

Structure: Result

Class
diagram

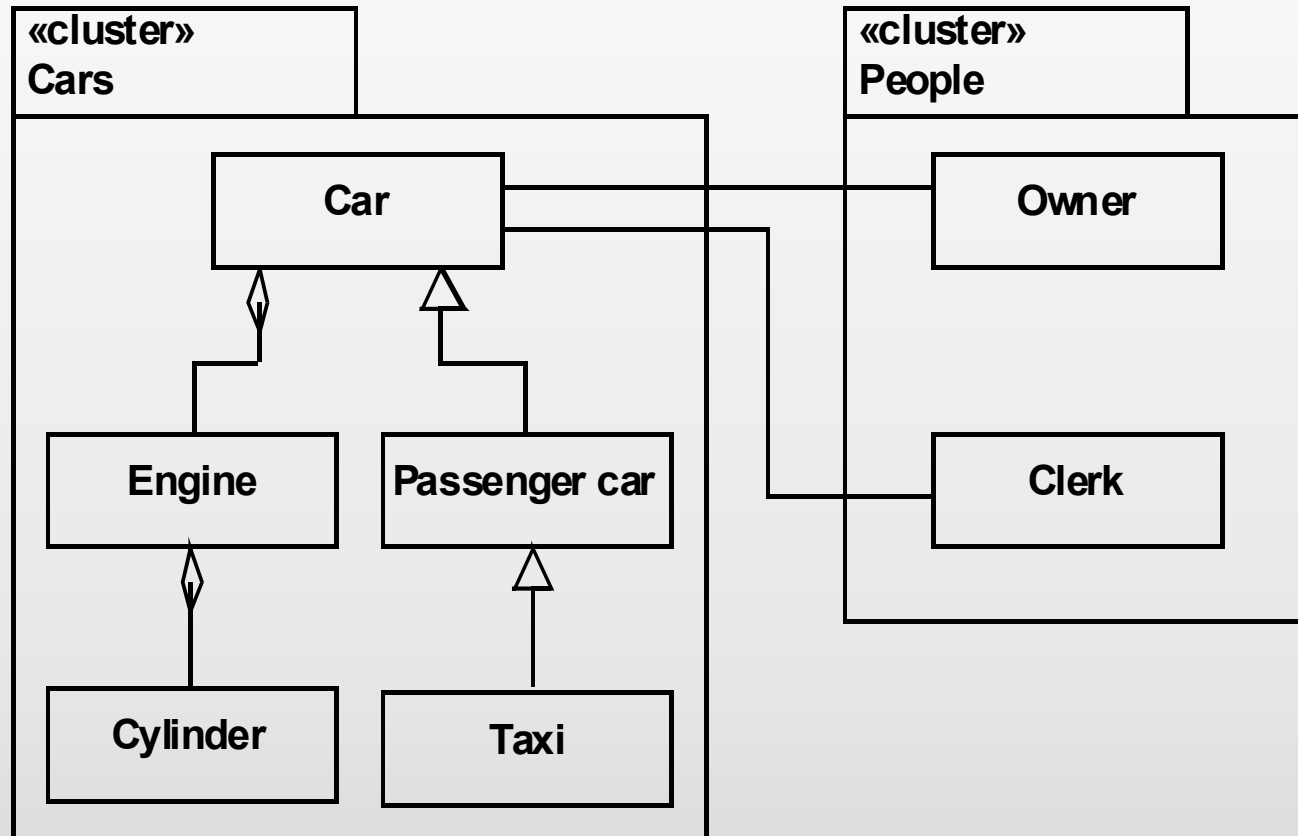


Key Concepts: Generalization

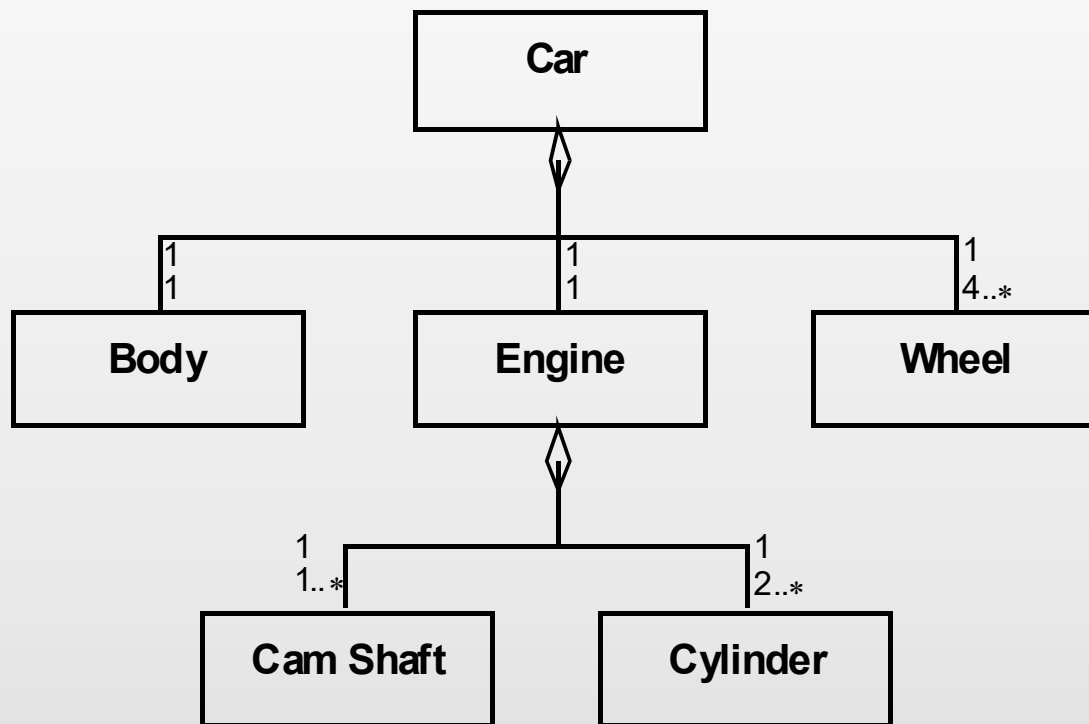


- ▶ 'is-a' relation
- ▶ Generalization class
- ▶ Specialization class
- ▶ Abstract class

Key Concepts: Cluster



Key Concepts: Aggregation



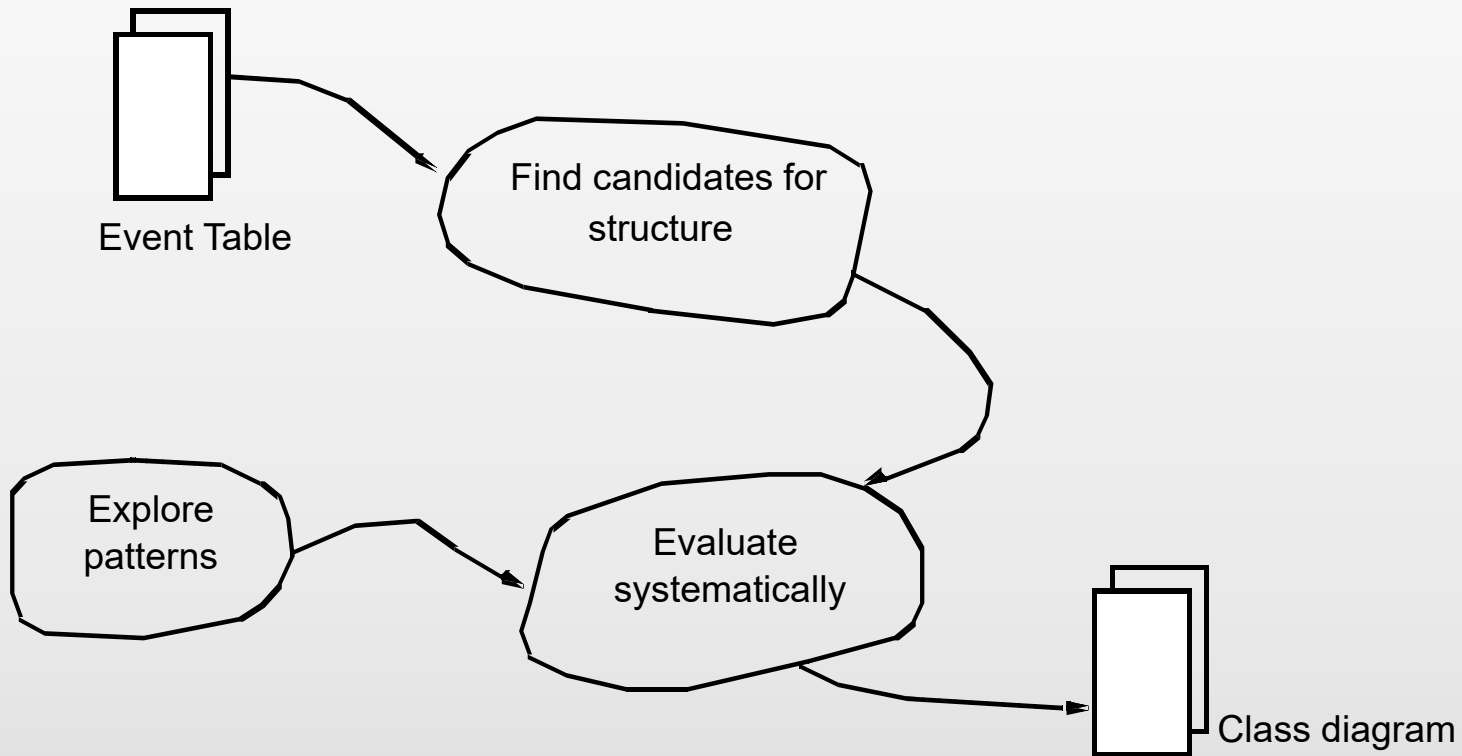
- ▶ Physical aggregation (unusual)
- ▶ 'has-a' or 'owns-a' relation (more common)
- ▶ Usually on different (conceptual) levels
- ▶ Typical aggregations:
 - Whole - part
 - Container - contents
 - Union - member

Key Concepts: Association



- ▶ Relation
- ▶ Usually a loose (non-defining) relation
- ▶ Usually on the same level
- ▶ Can be named (but that is often because a class is missing)

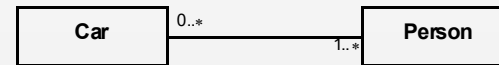
Structure: Activities



Evaluate Systematically

Structures must be used correctly

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



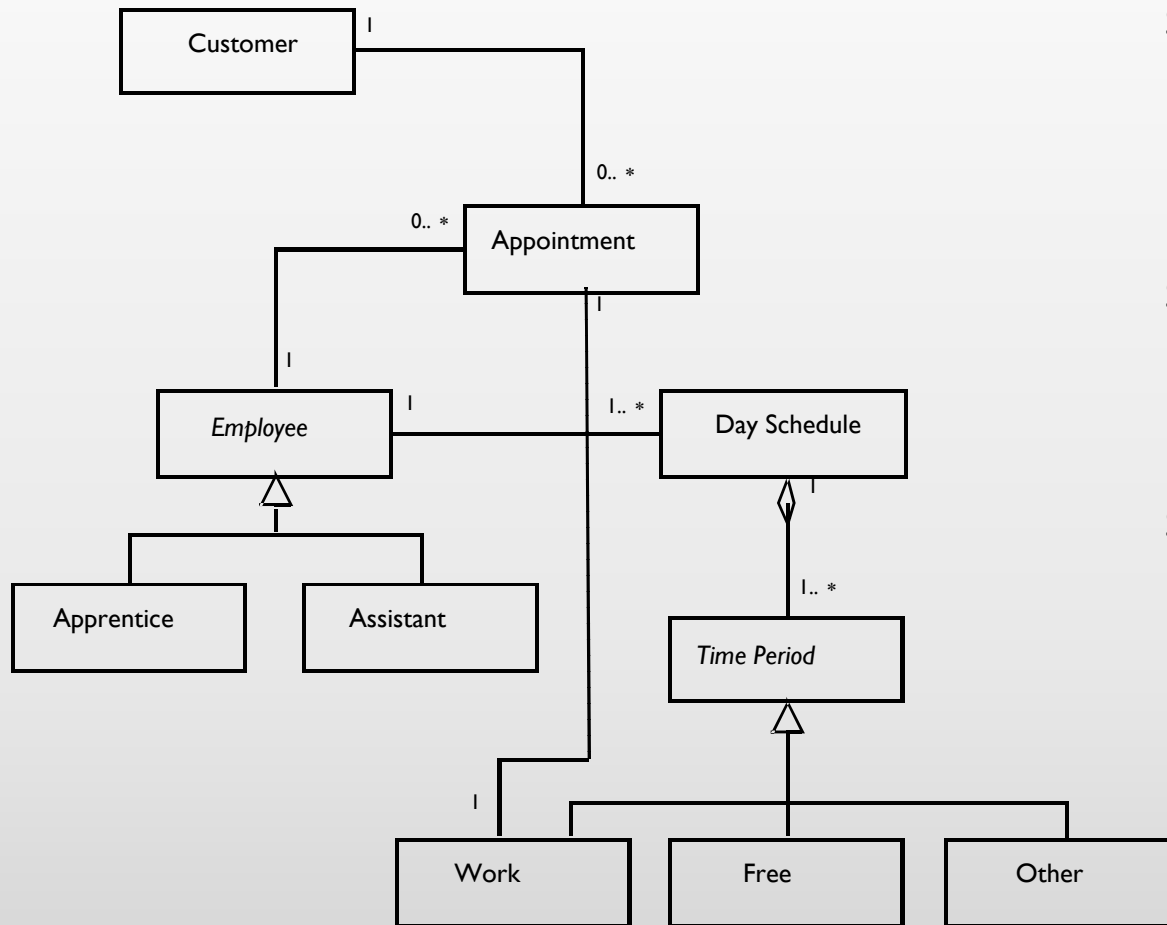
Structures must be conceptually true

- 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

Discuss this Class Diagram



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

Problems in the Class Diagram

- ▶ Customer – Appointment: Aggregation from Customer
- ▶ Appointment – Employee: OK
- ▶ Employee – Day Schedule: Aggregation from Employee
- ▶ Time Period specializations: Different solution, e.g. remove the three specializations

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.

Contents

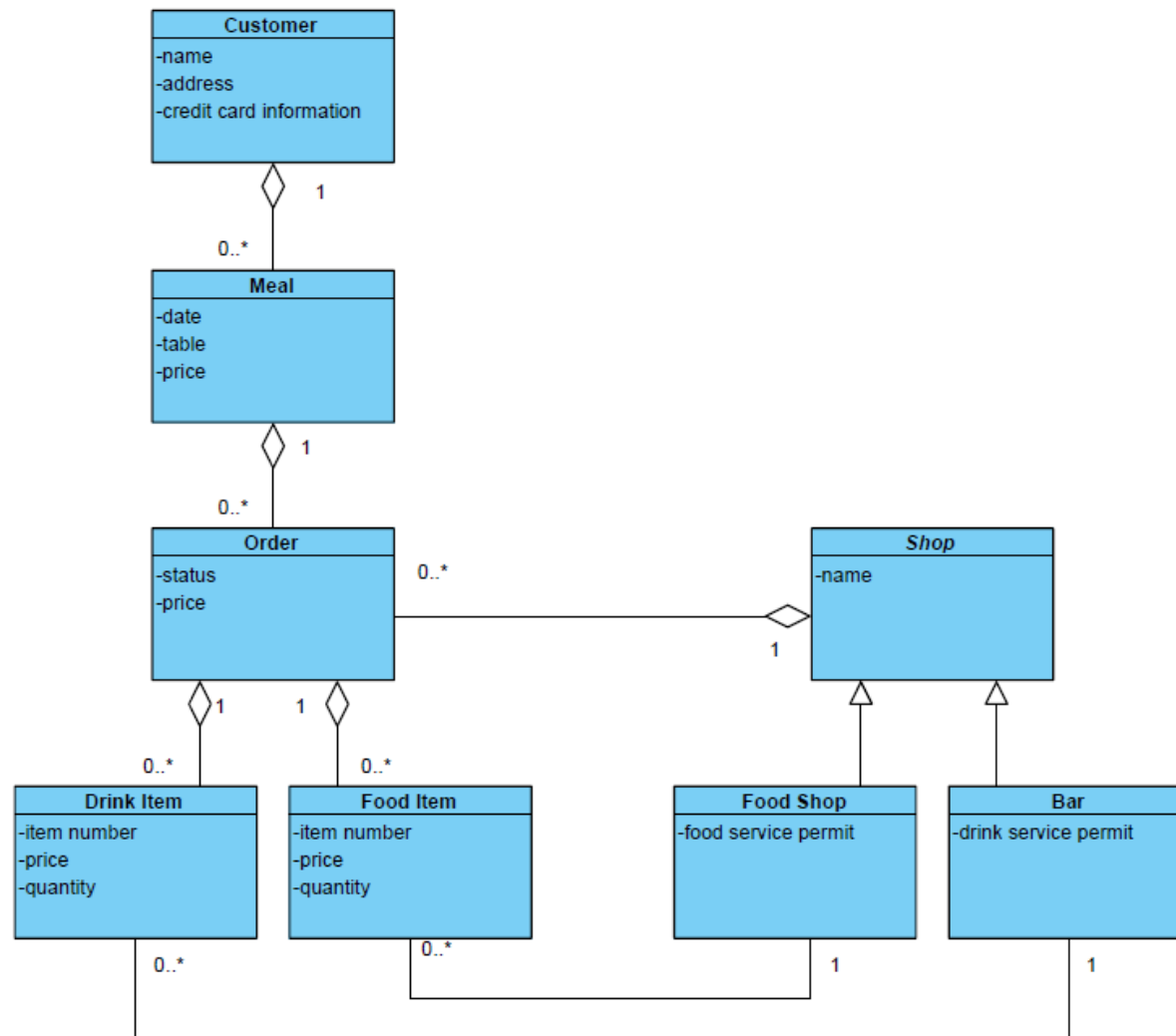
- ▶ Hand-in assignment
- ▶ Summary of last lecture
- ▶ The Structure activity
- ▶ **Example: street food**
- ▶ Explore patterns
- ▶ Challenges in this activity

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.

- ▶ Make a first class diagram

Street Food Class Diagram

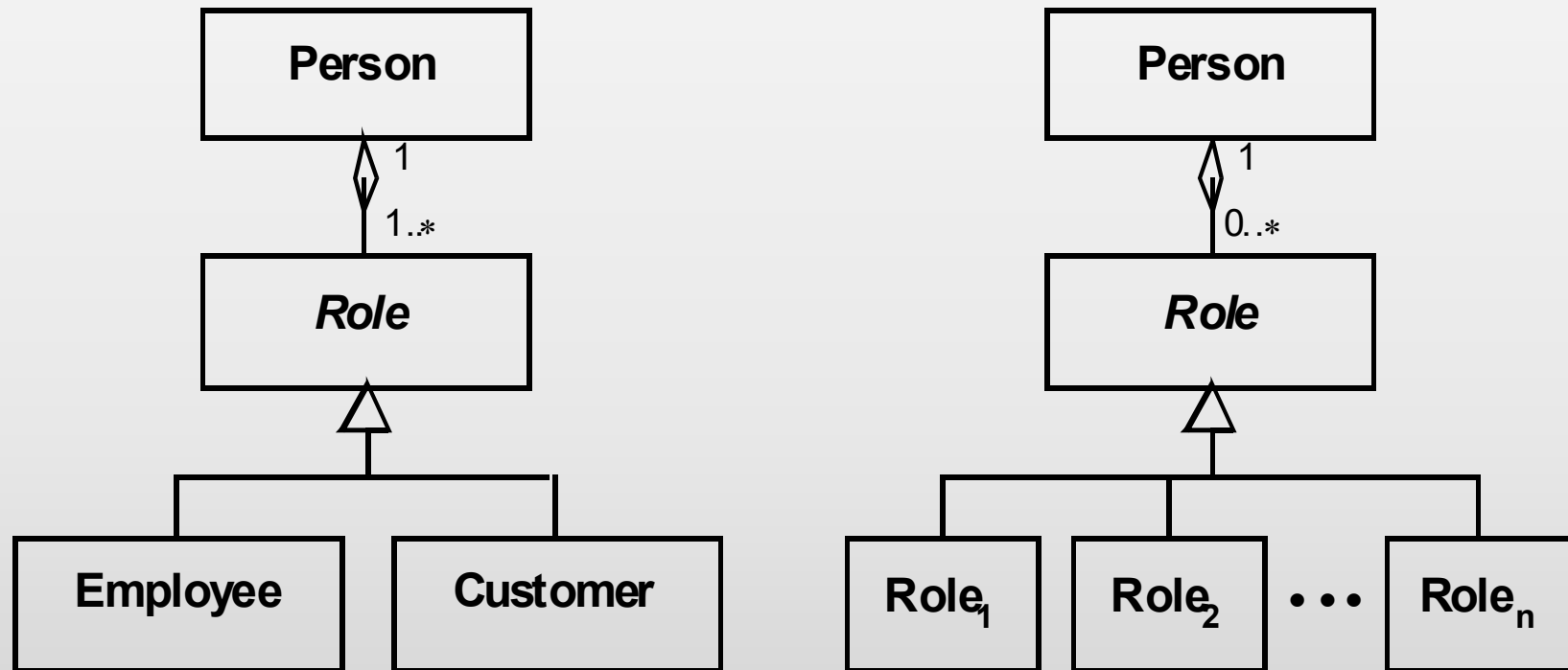


Contents

- ▶ Hand-in assignment
- ▶ Summary of last lecture
- ▶ The Structure activity
- ▶ Example: street food
- ▶ Explore patterns
- ▶ Challenges in this activity

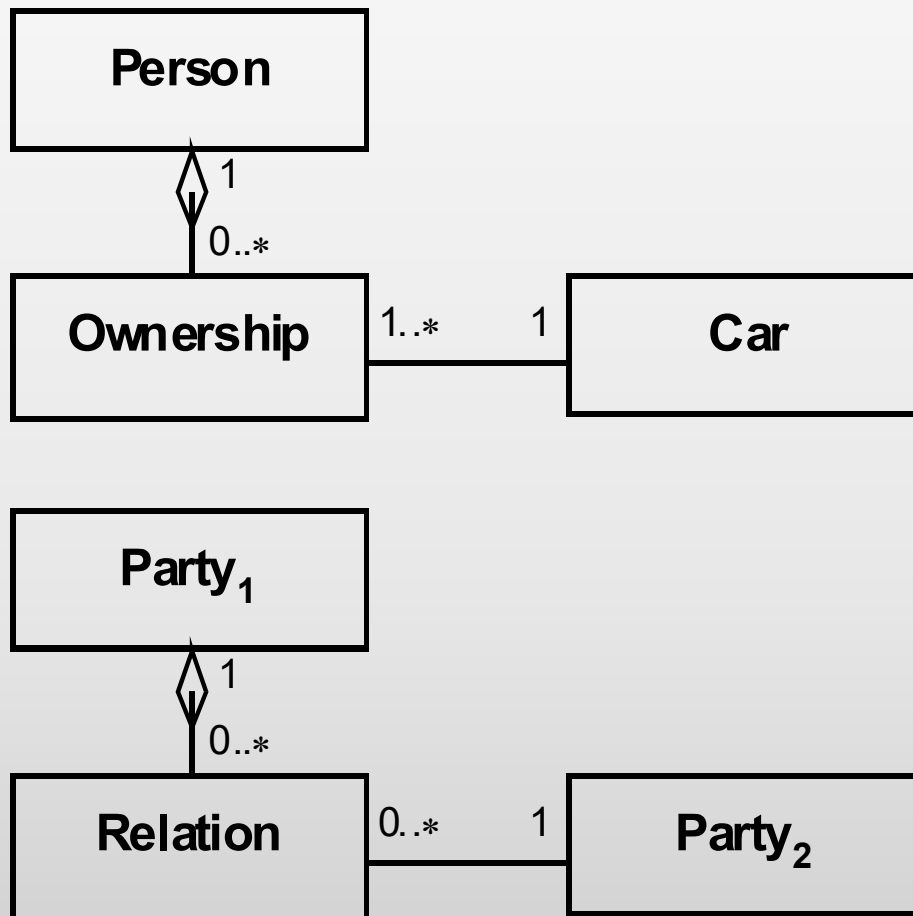
Explore Patterns: Role

- ▶ A person can have different roles
- ▶ The roles for a person change dynamically over time



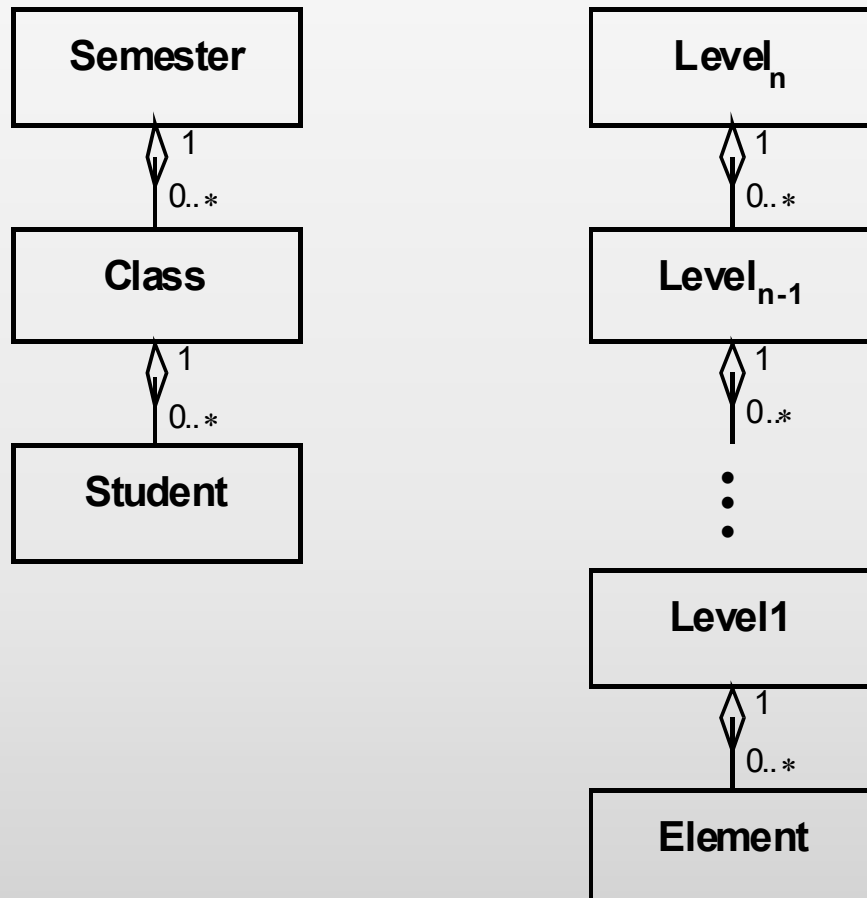
Explore Patterns: Relation

- Objects from two classes are related



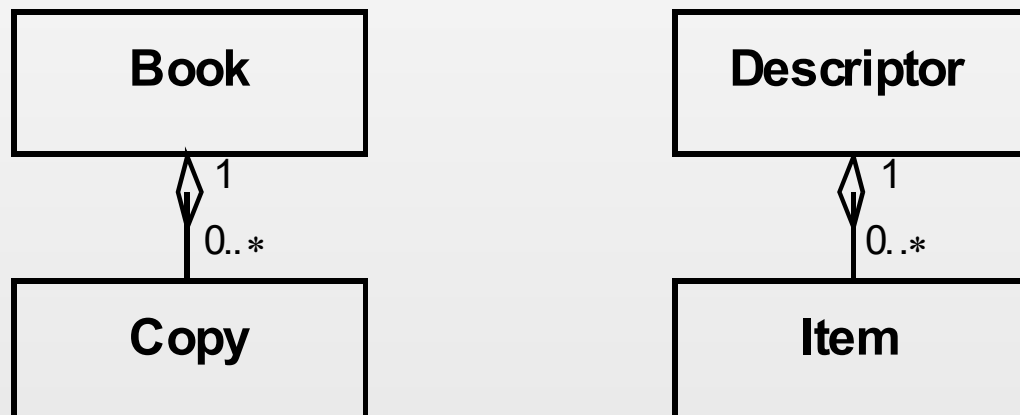
Explore Patterns: Hierarchy

- Objects from different classes form a hierarchy

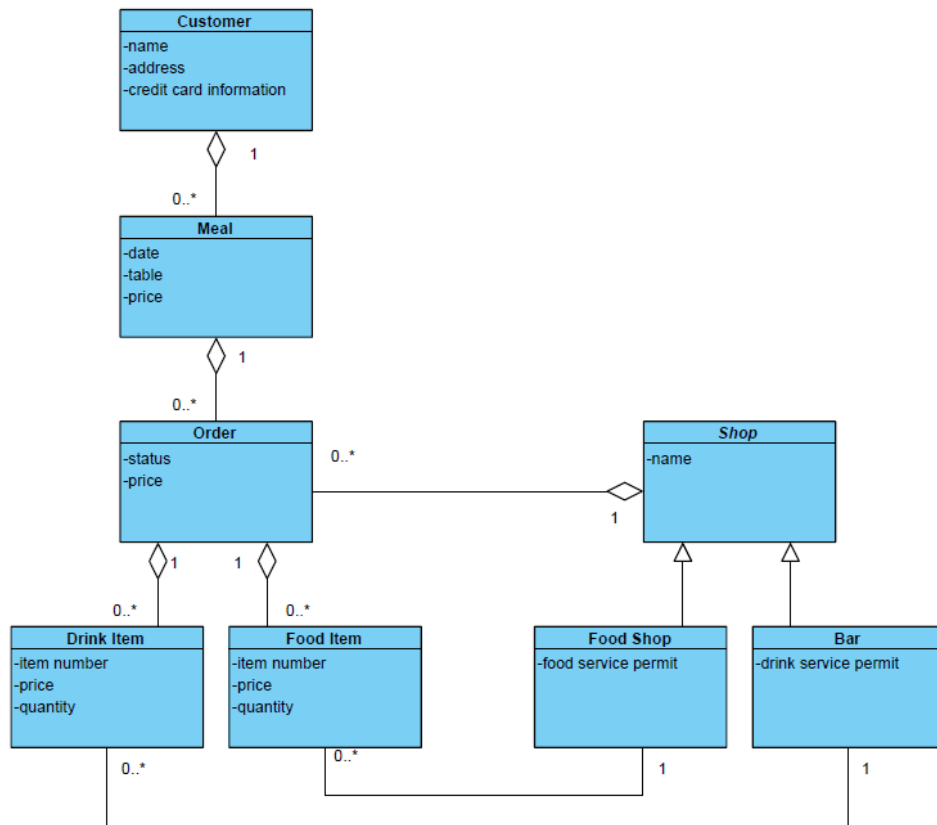


Explore Patterns: Item-Descriptor

- ▶ Properties of objects from one class (items) are described in an object from another class (descriptor)



Identify Patterns



► Which patterns are used in this class diagram?

Pattern	Classes connected
Hierarchy	Customer – Meal – Order – Drink Item/Food Item
Role	Shop – Food Shop – Bar
Relation	Order – Drink Item - Bar
Relation	Order – Food Item – Food Shop
Relation	Meal – Order - Shop
Hierarchy (not so obvious)	Shop – Order – Food Item/Drink Item

Contents

- ▶ Hand-in assignment
- ▶ Summary of last lecture
- ▶ The Structure activity
- ▶ Example: street food
- ▶ Explore patterns
- ▶ Challenges in this activity

Challenges

- ▶ Selecting the right structure is difficult
 - Try them out one by one for each pair of classes
 - Use the criteria to select the most correct structure
- ▶ It is very easy to include too many structures
 - Try to simulate functions and see if you can get to the relevant objects

Work in Exercises for this Activity

- ▶ Find candidates for structural relations between the classes in your event table – do this pair by pair
- ▶ Explore if any of the patterns are relevant to your case – do this for each class
- ▶ Make a class diagram
- ▶ Look for opportunities to simplify and extend the class diagram