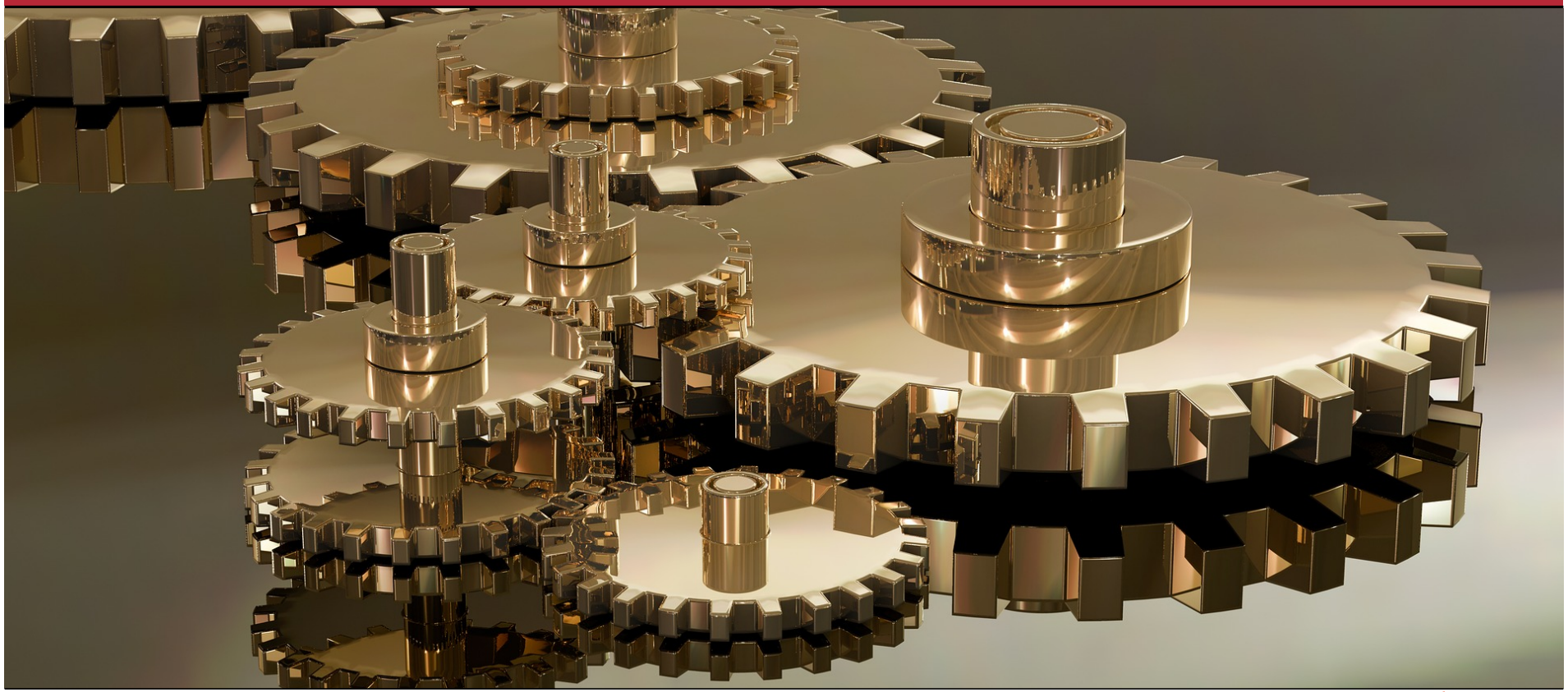
Software Engineering

TECHNISCHE UNIVERSITÄT DARMSTADT

Domain Modelling

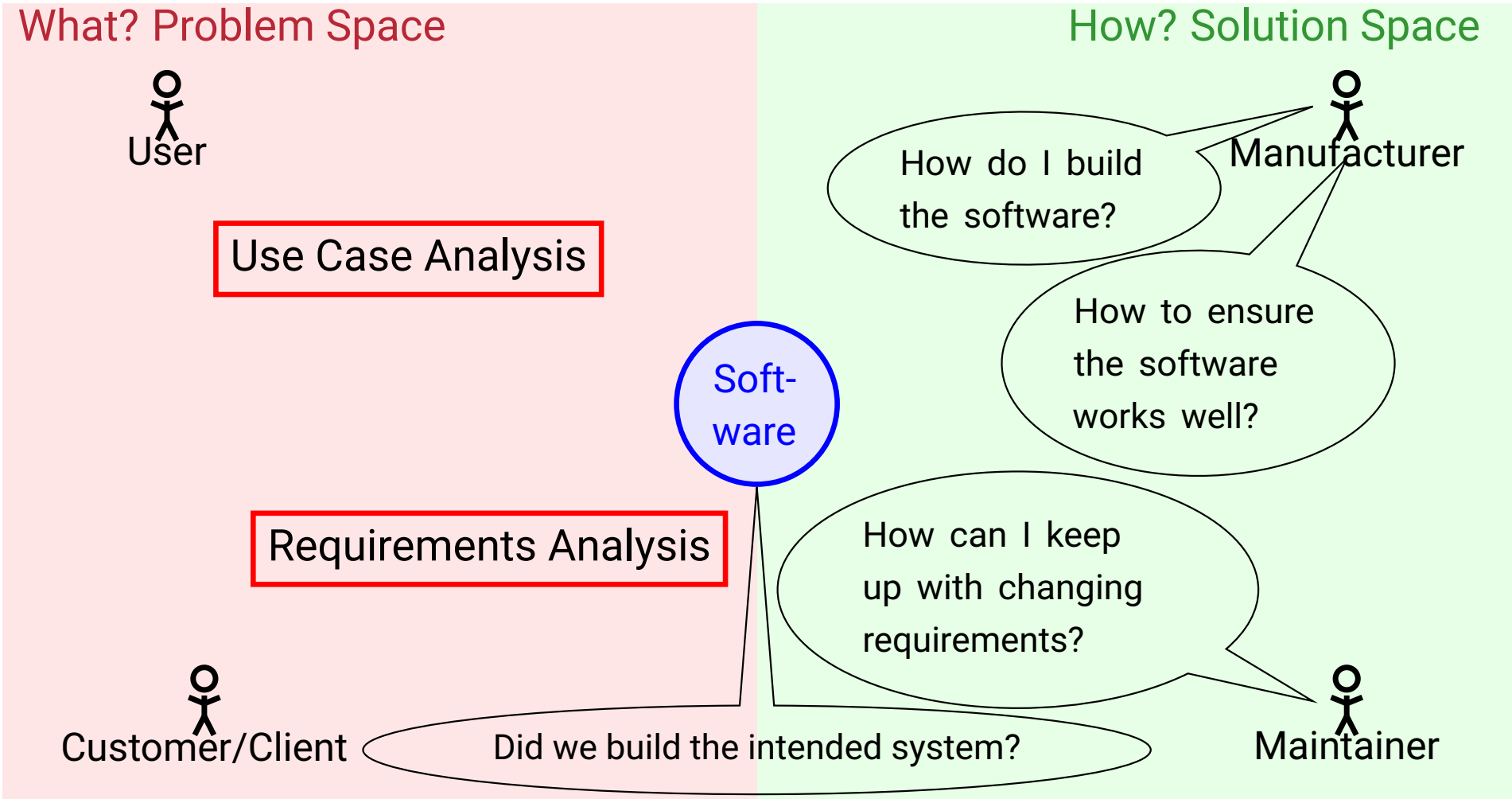
Prof. Dr. Reiner Hähnle
Fachgebiet Software Engineering



Software Engineering Group

Course Navigation

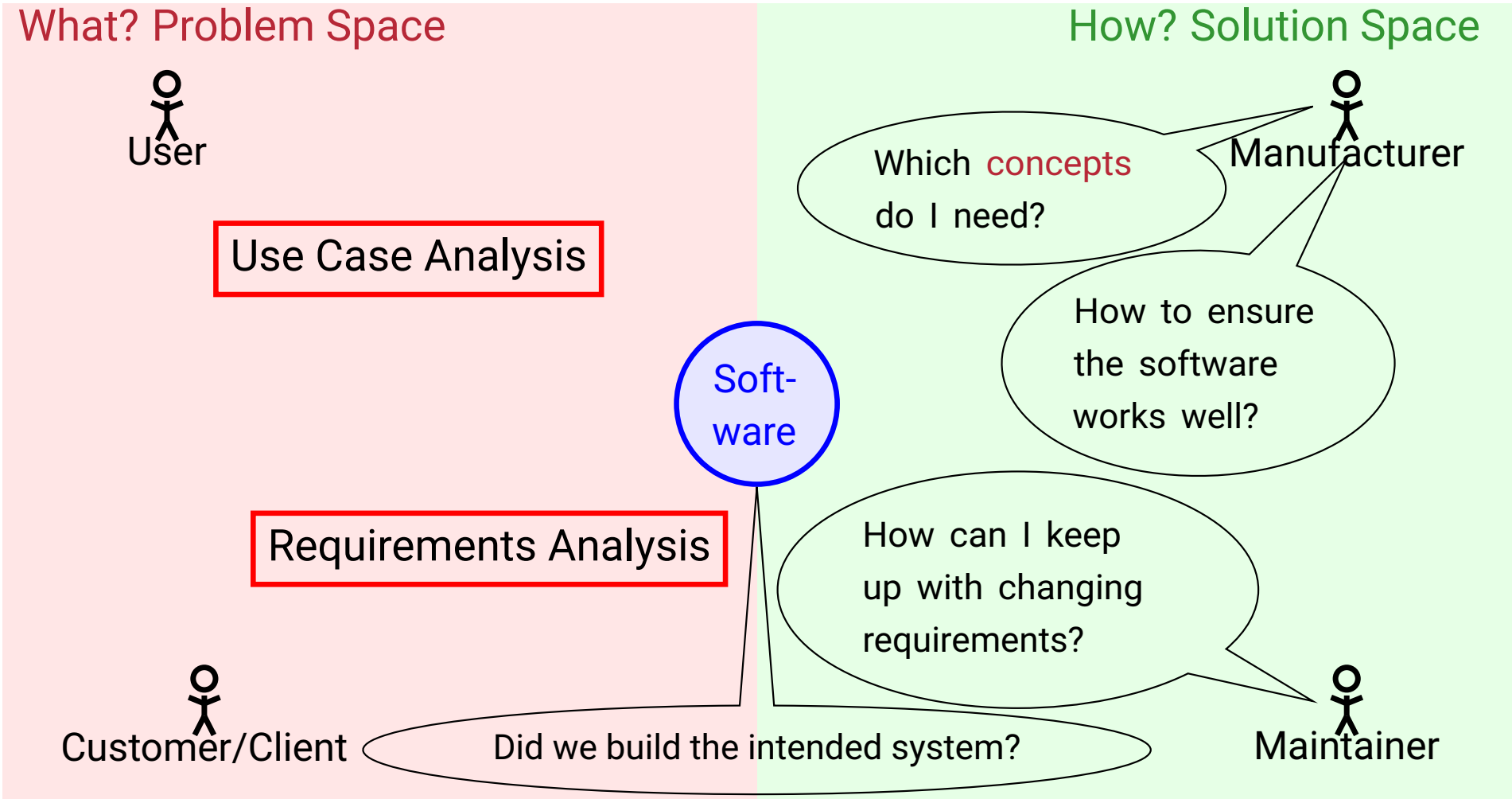
TECHNISCHE UNIVERSITÄT DARMSTADT



Software Engineering Group

Course Navigation

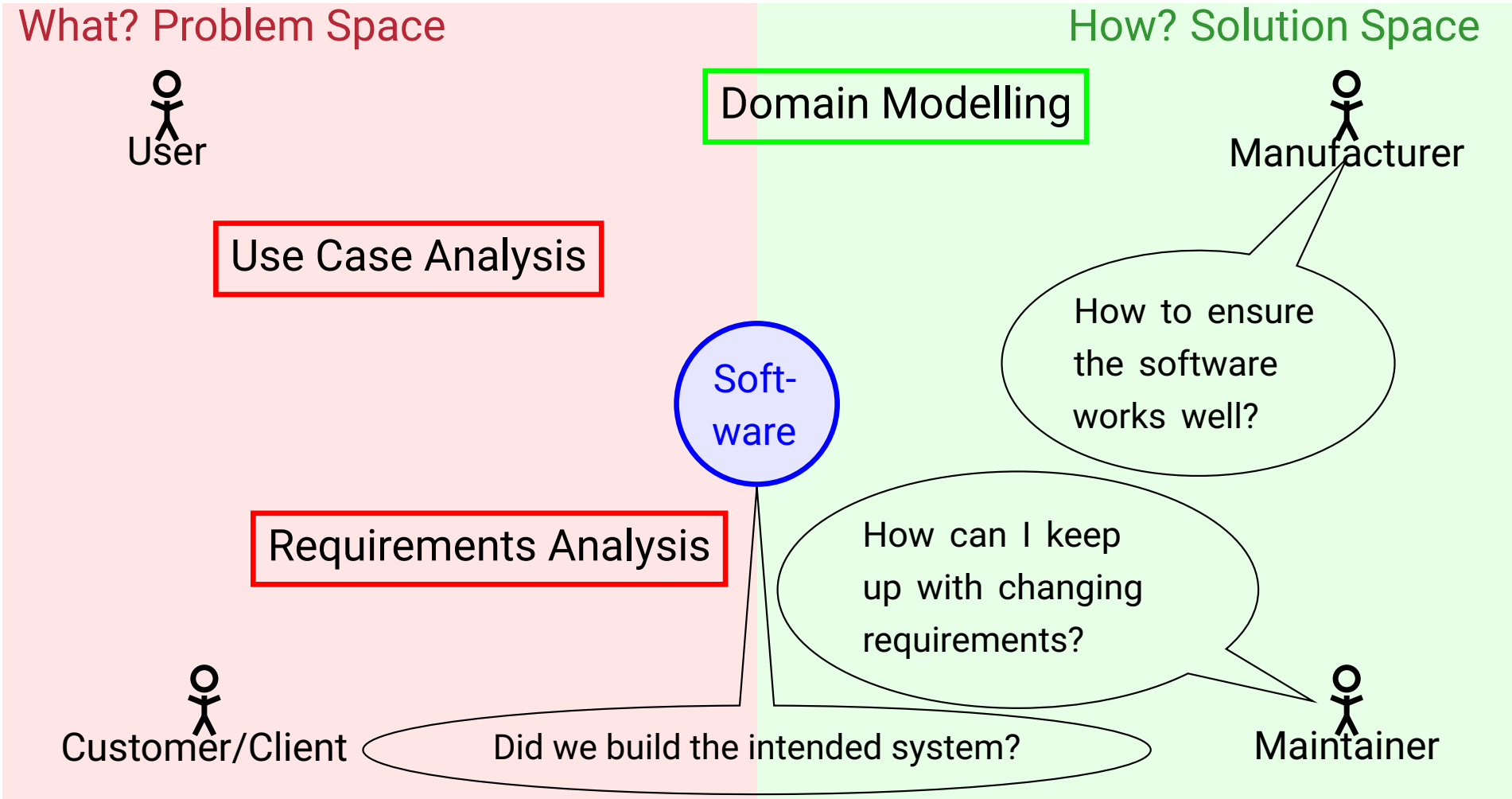
TECHNISCHE UNIVERSITÄT DARMSTADT



Software Engineering Group

Course Navigation

TECHNISCHE UNIVERSITÄT DARMSTADT



Software Engineering Group

General Approach: Object-oriented Analysis & Design

TECHNISCHE UNIVERSITÄT DARMSTADT

In the solution space we use Object-oriented Analysis & Design (OOAD)

Software Engineering Group

General Approach: Object-oriented Analysis & Design

TECHNISCHE UNIVERSITÄT DARMSTADT

In the solution space we use Object-oriented Analysis & Design (OOAD)

Motivation

- Considered state-of-art
- Widely-used in industrial practice, many software processes
- Numerous training resources available
- Compatible with wide-spread visual modeling notation UML
- Compatible with C++, C#, JAVA, KOTLIN, SCALA, ...

Domain Modelling

TECHNISCHE UNIVERSITÄT DARMSTADT

Domain Modelling

Fixing the terminology and fundamental activities of the solution space

Why: Helps to identify the relevant concepts and tasks of a domain

Prerequisite: Requirements analysis and use case analysis are done

Context: Part of object-oriented analysis (OOA)

Heuristics: Only performed for the tasks at hand

Curtis' law: "...good designs require deep application knowledge."

— A. Endres & D. Rombach, A Handbook of Software and Systems Engineering

Domain Modelling

TECHNISCHE UNIVERSITÄT DARMSTADT

Domain Modelling

Fixing the terminology and fundamental activities of the solution space

Why: Helps to identify the relevant concepts and tasks of a domain

Prerequisite: Requirements analysis and use case analysis are done

Context: Part of object-oriented analysis (OOA)

Heuristics: Only performed for the tasks at hand

Curtis' law: "...good designs require deep application knowledge."

— A. Endres & D. Rombach, A Handbook of Software and Systems Engineering

Domain model must be validated with Client

Domain Model

TECHNISCHE UNIVERSITÄT DARMSTADT

Goal of an Object-Oriented Domain Model

Decompose the domain into concepts or objects that represent the real world (as captured in the requirements specification)

How to create an Object-Oriented Domain Model

- Identify the set of conceptual classes and fundamental actions

Iteratively completed and forms the basis for the software design

Domain Model

TECHNISCHE UNIVERSITÄT DARMSTADT

Goal of an Object-Oriented Domain Model

Decompose the domain into concepts or objects that represent the real world (as captured in the requirements specification)

How to create an Object-Oriented Domain Model

- Identify the set of conceptual classes and fundamental actions

Iteratively completed and forms the basis for the software design

Synonyms for Domain Model

- Conceptual model, domain object model, analysis object model
  “Konzeptmodell”, “Analysemodell”

Conceptual Classes

TECHNISCHE UNIVERSITÄT DARMSTADT

Definition (Conceptual class “Konzeptklasse”)

Conceptual classes represent ideas, things or objects in the domain.
A conceptual class has

- a name or symbol representing the class,
- an intention (“Bedeutung”),
- an extension (“Ausprägung”, “Extension”).

The extension contains the domain elements the conceptual class represents.

Software Engineering Group

Conceptual Classes

TECHNISCHE UNIVERSITÄT DARMSTADT

Definition (Conceptual class “Konzeptklasse”)

Conceptual classes represent ideas, things or objects in the domain.
A conceptual class has

- a name or symbol representing the class,
- an intention (“Bedeutung”),
- an extension (“Ausprägung”, “Extension”).

The extension contains the domain elements the conceptual class represents.

Domain concepts/conceptual classes are not software objects –
They derive from the problem space

Software Engineering Group

Part II

UML Class Diagrams as Conceptual Models

Visualization of Domain Models

TECHNISCHE UNIVERSITÄT DARMSTADT

Domain models are visualized using UML class diagrams

But with suitable restrictions to emphasize domain modelling:

- Only domain objects and conceptual classes
- Only associations, no aggregation, no composition
- Classes may have attributes (used sparingly), but no operations

UML class diagram of domain model is called conceptual perspective model
"Modell der Konzeptperspektive"

Example: CaSh Booking System

TECHNISCHE UNIVERSITÄT DARMSTADT

- Bookings are made by customers
- Customers must have an authentication to make a booking
- Each booking is for exactly one car
- The monthly bill comprises all completed and unpaid bookings
- Each booking is for a duration of between 30 minutes and four weeks
- A customer cannot have more than three active bookings
- Each car needs to have a regular service carried out by a service person

Software Engineering Group

Example: CaSh Booking System

TECHNISCHE UNIVERSITÄT DARMSTADT

Booking

Customer

- Bookings are made by customers
- Customers must have an authentication to make a booking
- Each booking is for exactly one car
- The monthly bill comprises all completed and unpaid bookings
- Each booking is for a duration of between 30 minutes and four weeks
- A customer cannot have more than three active bookings
- Each car needs to have a regular service carried out by a service person

Example: CaSh Booking System

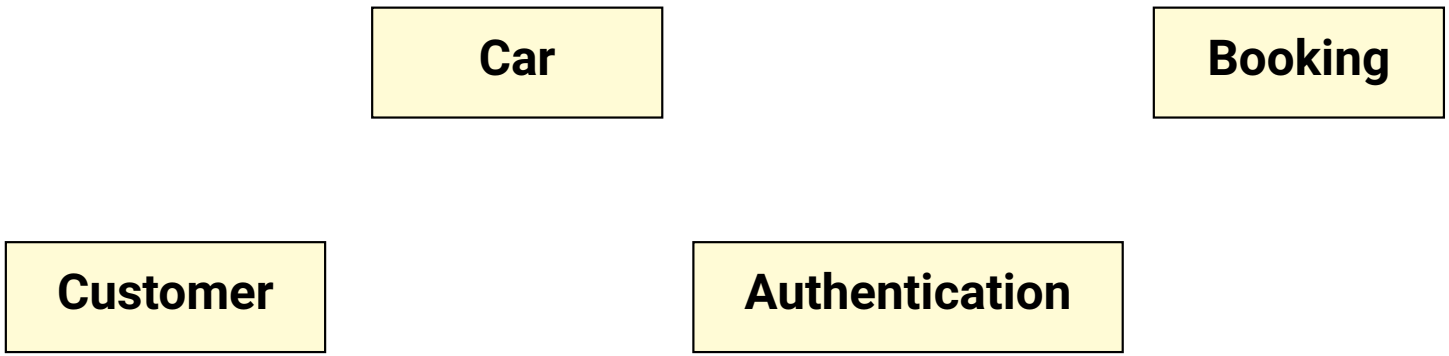
TECHNISCHE UNIVERSITÄT DARMSTADT

- Bookings are made by customers
- Customers must have an authentication to make a booking
- Each booking is for exactly one car
- The monthly bill comprises all completed and unpaid bookings
- Each booking is for a duration of between 30 minutes and four weeks
- A customer cannot have more than three active bookings
- Each car needs to have a regular service carried out by a service person

Software Engineering Group

Example: CaSh Booking System

TECHNISCHE UNIVERSITÄT DARMSTADT

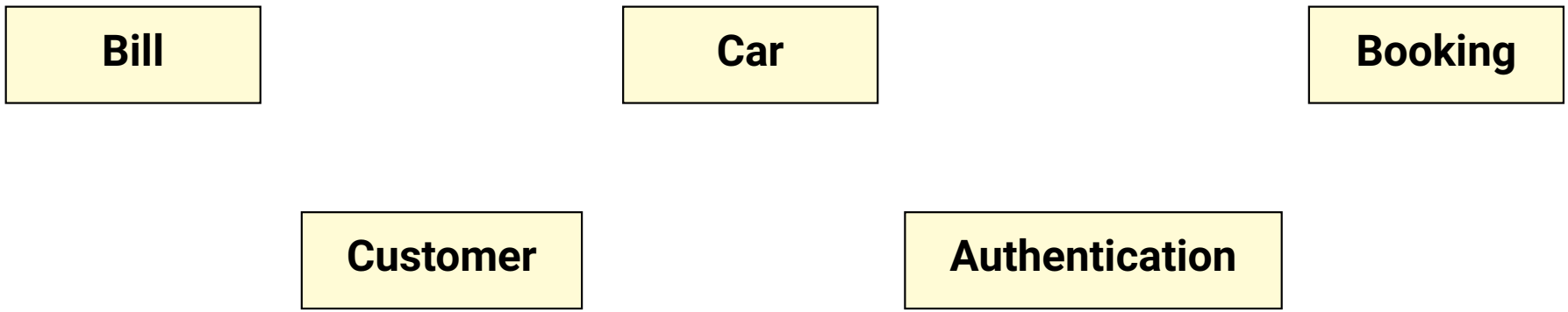


- Bookings are made by customers
- Customers must have an authentication to make a booking
- Each booking is for exactly one car
- The monthly bill comprises all completed and unpaid bookings
- Each booking is for a duration of between 30 minutes and four weeks
- A customer cannot have more than three active bookings
- Each car needs to have a regular service carried out by a service person

Software Engineering Group

Example: CaSh Booking System

TECHNISCHE UNIVERSITÄT DARMSTADT

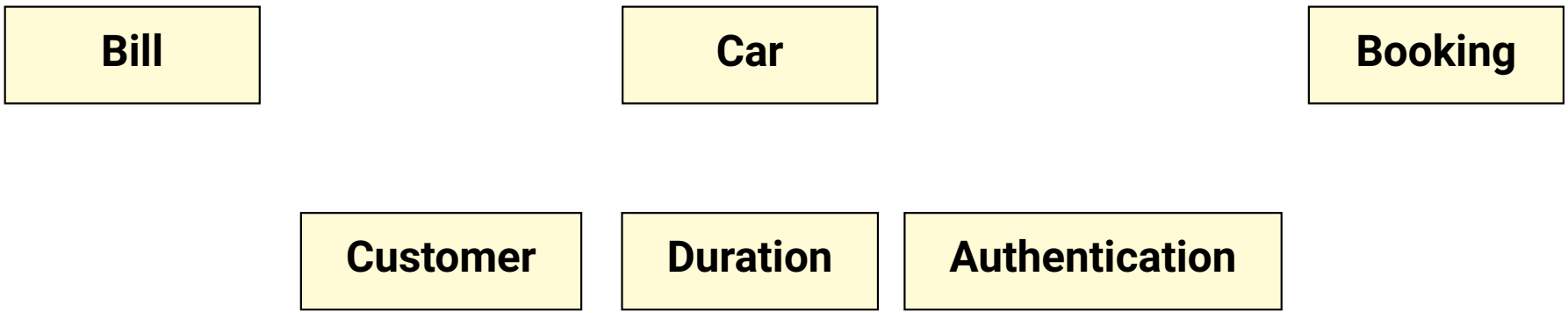


- Bookings are made by customers
- Customers must have an authentication to make a booking
- Each booking is for exactly one car
- The monthly bill comprises all completed and unpaid bookings
- Each booking is for a duration of between 30 minutes and four weeks
- A customer cannot have more than three active bookings
- Each car needs to have a regular service carried out by a service person

Software Engineering Group

Example: CaSh Booking System

TECHNISCHE UNIVERSITÄT DARMSTADT

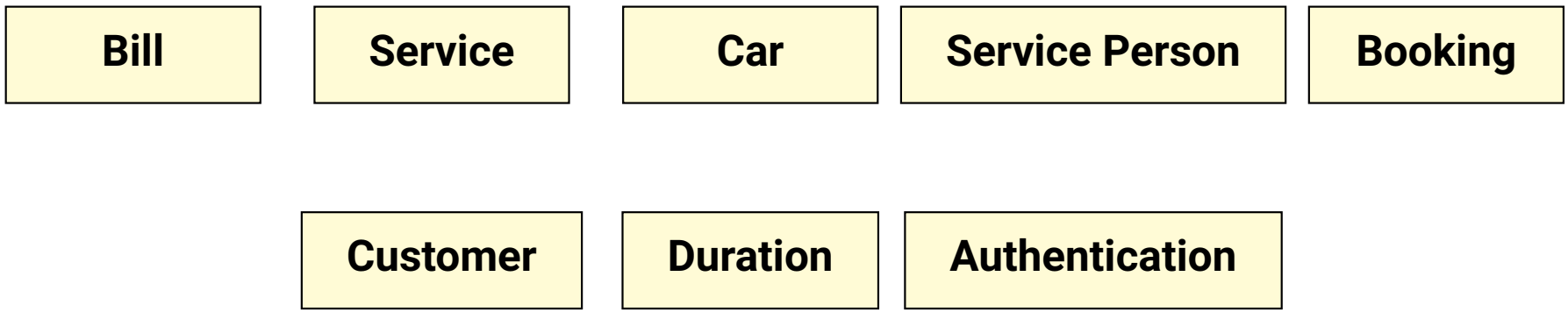


- Bookings are made by customers
- Customers must have an authentication to make a booking
- Each booking is for exactly one car
- The monthly bill comprises all completed and unpaid bookings
- Each booking is for a duration of between 30 minutes and four weeks
- A customer cannot have more than three active bookings
- Each car needs to have a regular service carried out by a service person

Software Engineering Group

Example: CaSh Booking System

TECHNISCHE UNIVERSITÄT DARMSTADT



- Bookings are made by customers
- Customers must have an authentication to make a booking
- Each booking is for exactly one car
- The monthly bill comprises all completed and unpaid bookings
- Each booking is for a duration of between 30 minutes and four weeks
- A customer cannot have more than three active bookings
- Each car needs to have a regular service carried out by a service person

Software Engineering Group

Extracting a First Class Model

TECHNISCHE UNIVERSITÄT DARMSTADT

Bill | **Service** | **Car** | **Service Person** | **Booking**

Customer | **Duration** | **Authentication**

Classes

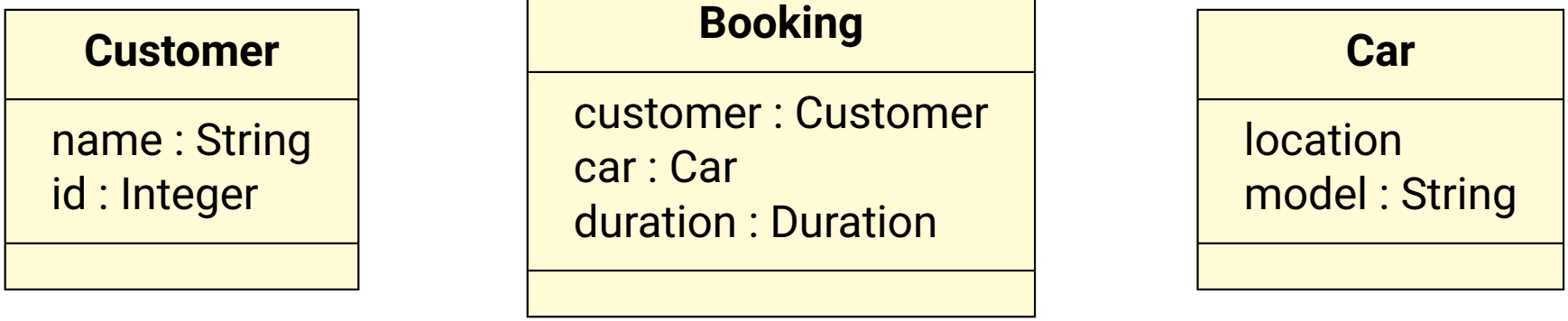
A (conceptual) class describes a set of (domain) objects

An object is an individual thing with a state and relations to other objects

Software Engineering Group

Properties of Conceptual Classes: Attributes

TECHNISCHE UNIVERSITÄT DARMSTADT



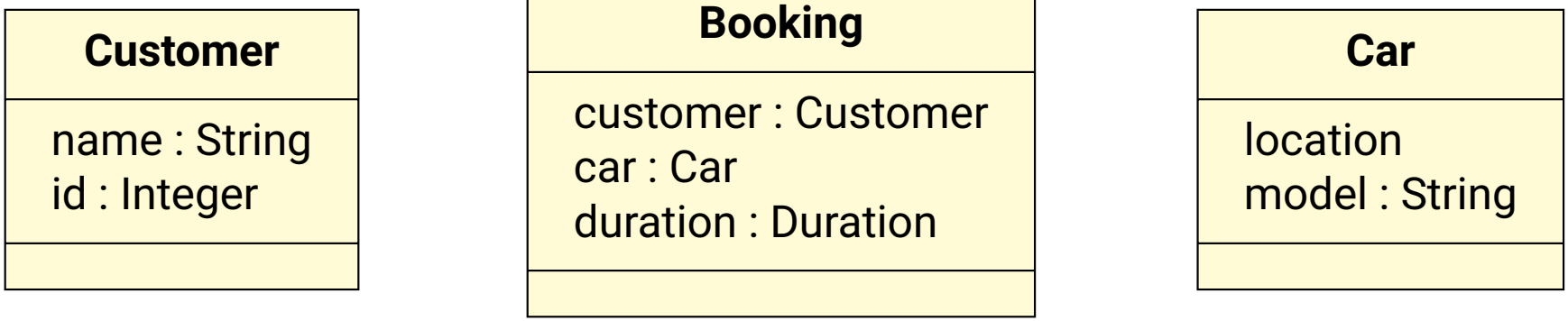
Attributes:

- Logical data values of an object (class instance, domain element)

Software Engineering Group

Properties of Conceptual Classes: Attributes

TECHNISCHE UNIVERSITÄT DARMSTADT



model(aCar)=“Opel Corsa”

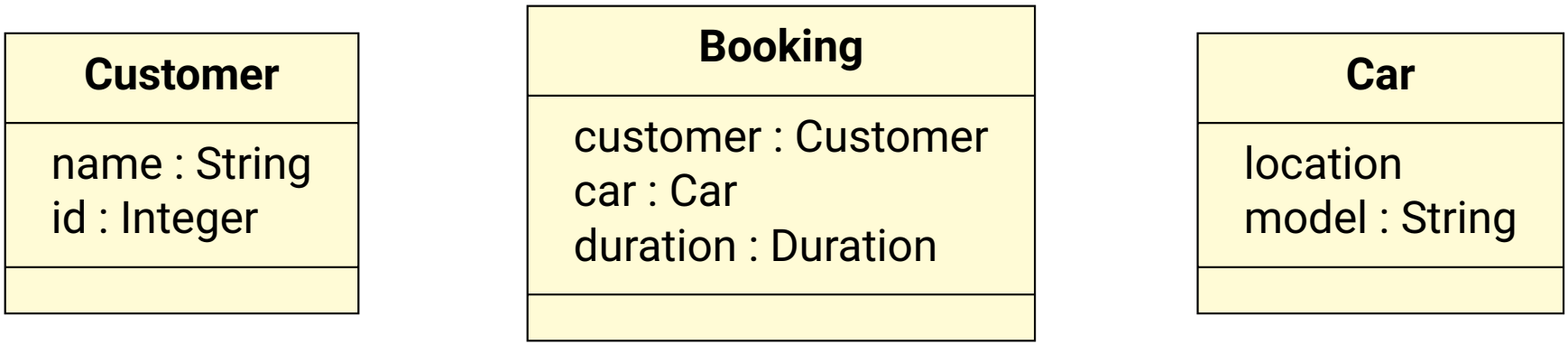
Attributes:

- Logical data values of an object (class instance, domain element)
- Map objects of the containing class to objects of their target type

Software Engineering Group

Properties of Conceptual Classes: Attributes

TECHNISCHE UNIVERSITÄT DARMSTADT



Attributes:

- Logical data values of an object (class instance, domain element)
- Map objects of the containing class to objects of their target type

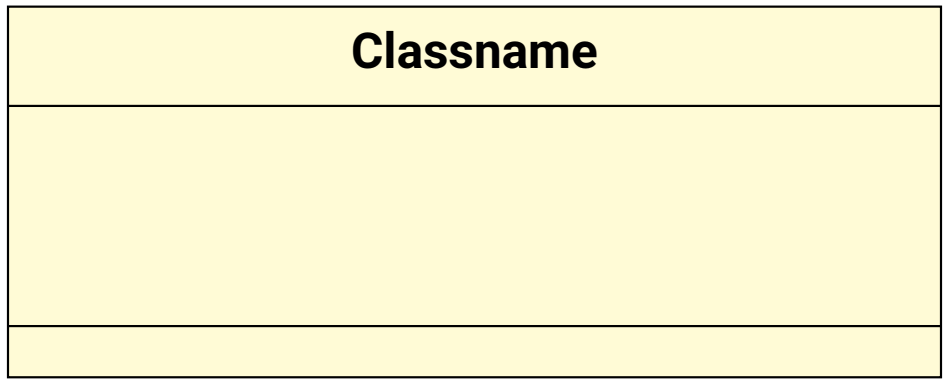
During domain modelling:

- Identify attributes of conceptual classes needed to satisfy information requirements
- Limit number of attributes to scope of treated scenarios

Software Engineering Group

UML Model Element Class: Simplified Syntax for Domain Modelling

TECHNISCHE UNIVERSITÄT DARMSTADT



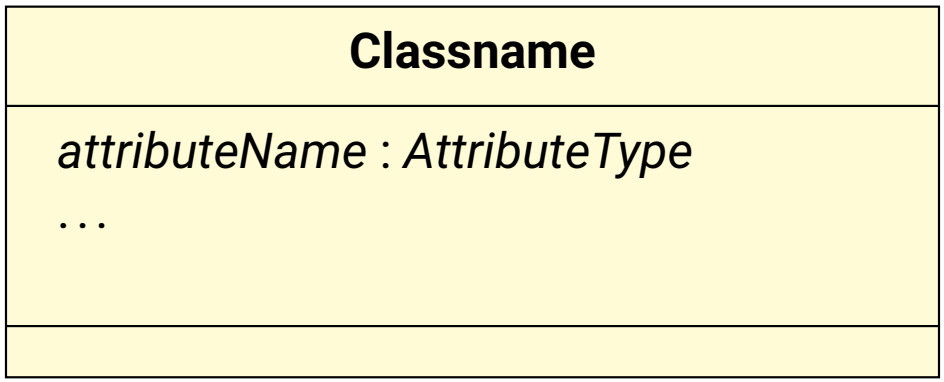
UML Class Elements and Conventions

Class Name Class names start always with an upper case letter

Software Engineering Group

UML Model Element Class: Simplified Syntax for Domain Modelling

TECHNISCHE UNIVERSITÄT DARMSTADT



UML Class Elements and Conventions

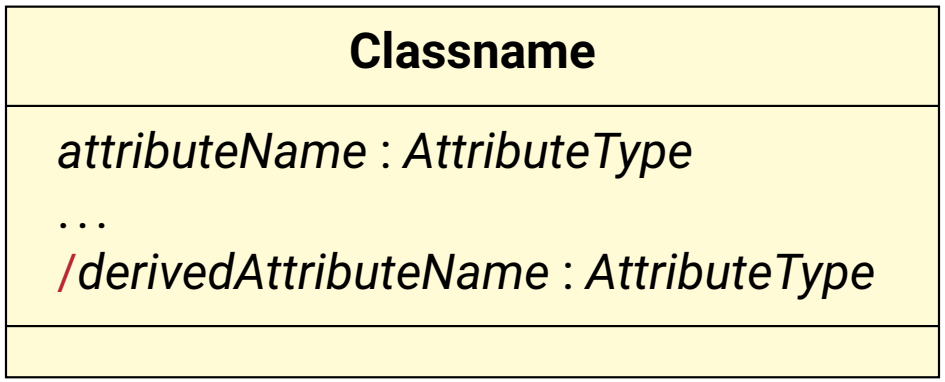
Class Name Class names start always with an upper case letter

Attribute Attribute name: starts with a lower case letter
Attribute type: pre-defined type or other domain model class
(type can be omitted in domain modelling)

Software Engineering Group

UML Model Element Class: Simplified Syntax for Domain Modelling

TECHNISCHE UNIVERSITÄT DARMSTADT



UML Class Elements and Conventions

Class Name Class names start always with an upper case letter

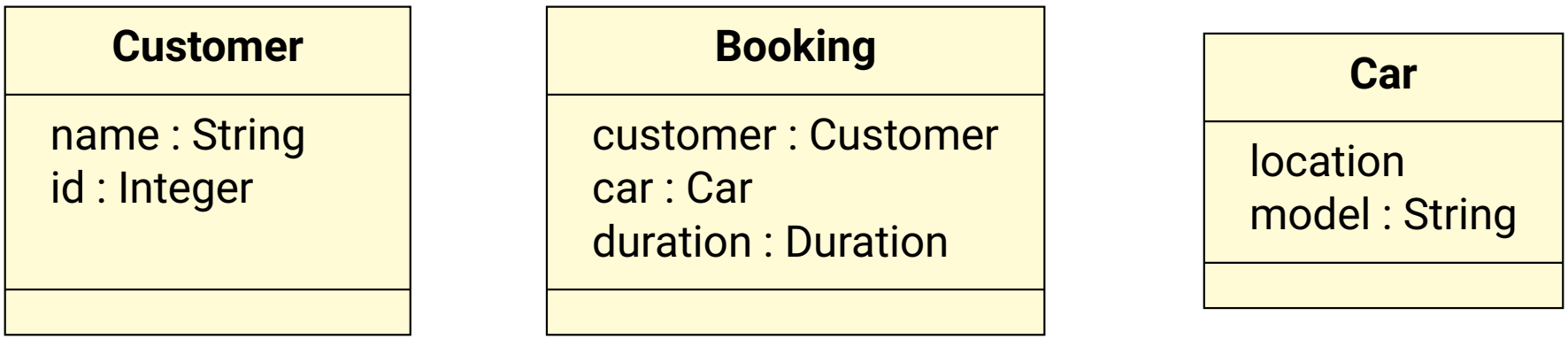
Attribute Attribute name: starts with a lower case letter
Attribute type: pre-defined type or other domain model class (type can be omitted in domain modelling)

Derived Attribute: name prefixed by a slash (’/’)
Value computable from existing information

Software Engineering Group

Example: Derived Attribute

TECHNISCHE UNIVERSITÄT DARMSTADT



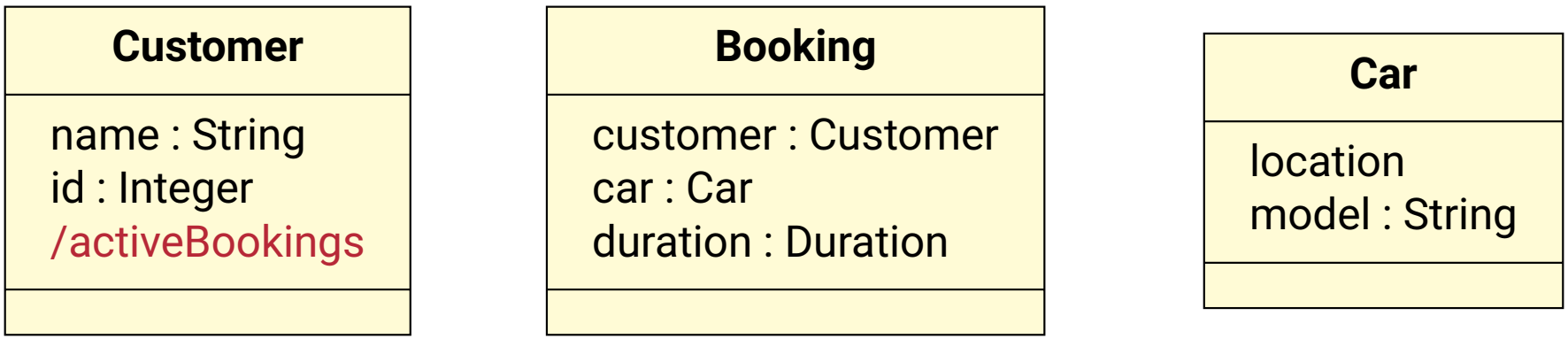
Example (Active Bookings)

- The active bookings of a customer are all not yet completed bookings

Software Engineering Group

Example: Derived Attribute

TECHNISCHE UNIVERSITÄT DARMSTADT



Example (Active Bookings)

- The active bookings of a customer are all not yet completed bookings

The active bookings for a customer can be computed:
Collect all bookings of the customer whose end time is not past the current time
Derived attributes are usually not assigned values independently

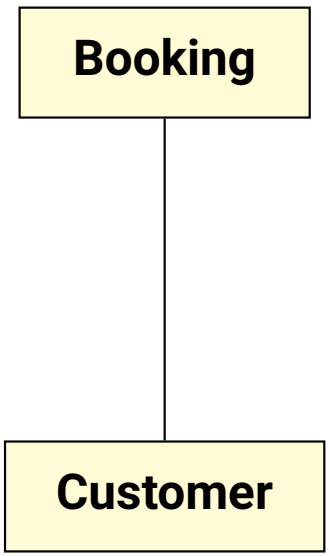
Software Engineering Group

CaSh Booking System: Associations (Relations)

TECHNISCHE UNIVERSITÄT DARMSTADT

- Bookings are made by customers

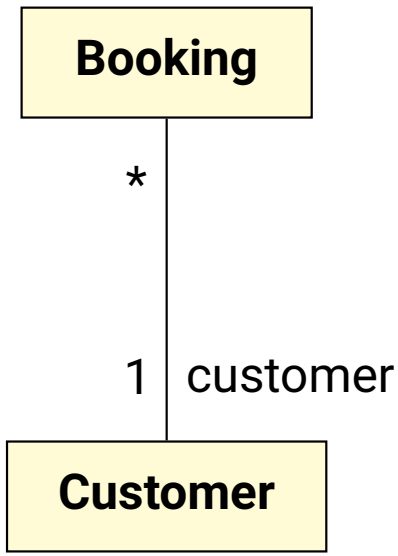
CaSh Booking System: Associations (Relations)



- Bookings are made by customers

CaSh Booking System: Associations (Relations)

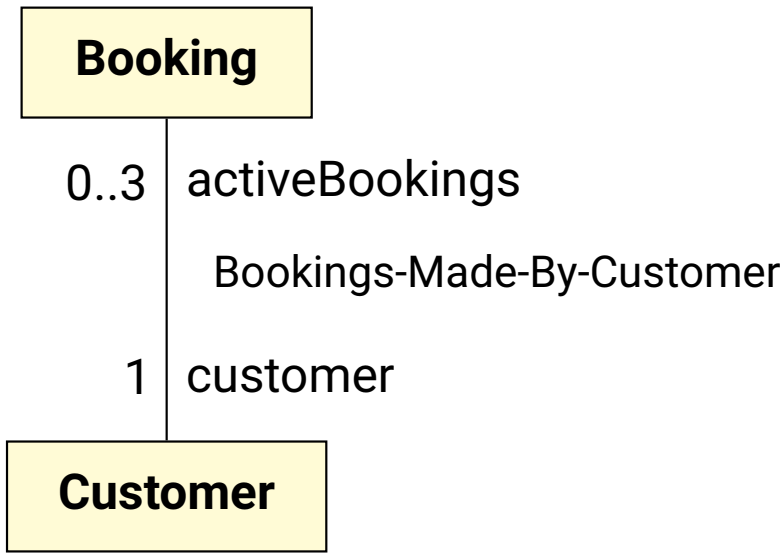
TECHNISCHE UNIVERSITÄT DARMSTADT



- Bookings are made by customers

Software Engineering Group

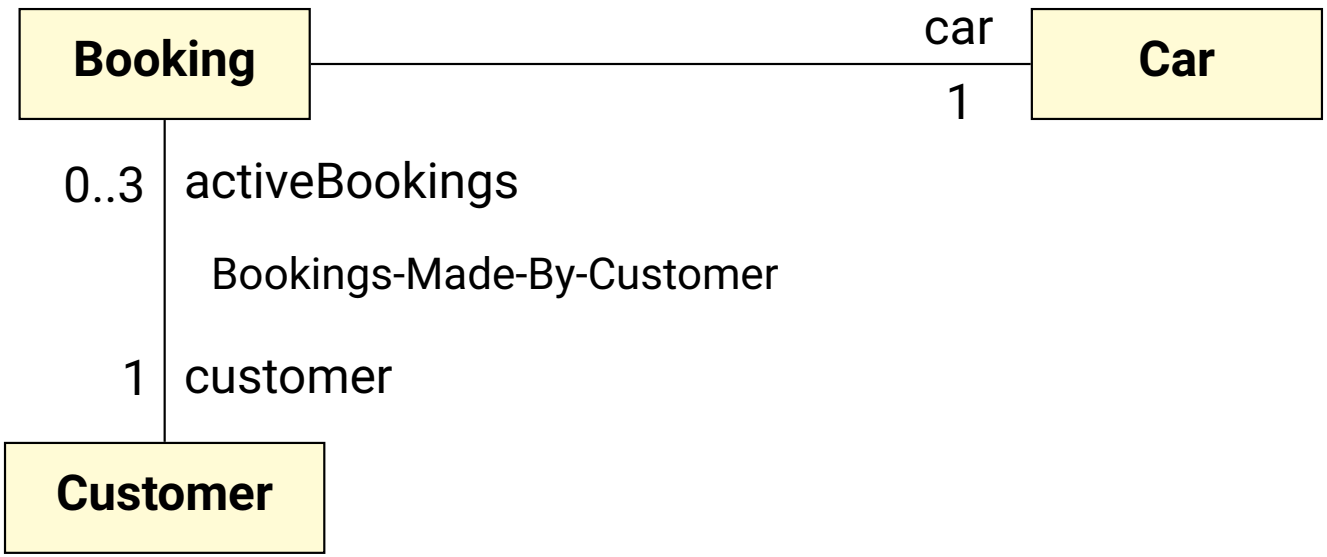
CaSh Booking System: Associations (Relations)



- Bookings are made by customers
- A customer cannot have more than three active bookings

CaSh Booking System: Associations (Relations)

TECHNISCHE UNIVERSITÄT DARMSTADT



- Bookings are made by customers
- A customer cannot have more than three active bookings
- Each booking is for exactly one car

Software Engineering Group

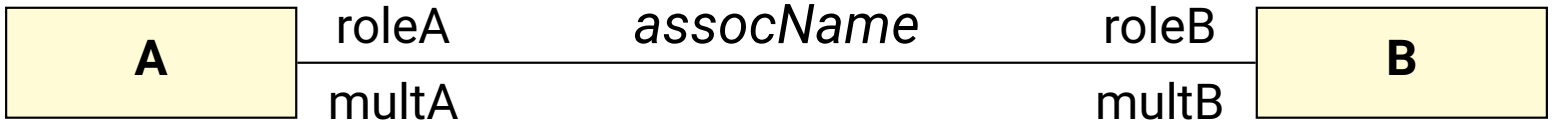
CaSh Booking System: Associations (Relations)

TECHNISCHE UNIVERSITÄT DARMSTADT

- Bookings are made by customers
- A customer cannot have more than three active bookings
- Each booking is for exactly one car
- A car may appear in an arbitrary number of bookings

Software Engineering Group

Syntax of UML Association



An association is a relation among classes with the elements:

Syntax of UML Association

An association is a relation among classes with the elements:

- Name (optional)

Syntax of UML Association

An association is a relation among classes with the elements:

- Name (optional)
- Two association ends called roles, each with:
  - A role name (defaults to class name starting with lower case letter)

Syntax of UML Association

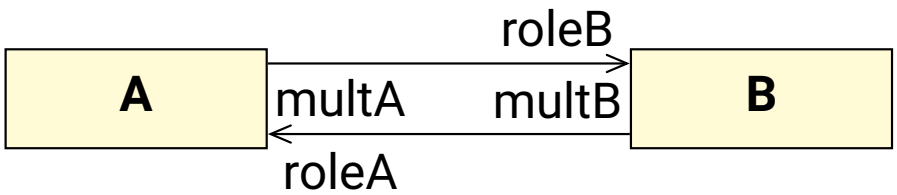
An association is a relation among classes with the elements:

- Name (optional)
- Two association ends called roles, each with:
  - A role name (defaults to class name starting with lower case letter)
  - A multiplicity (defaults to multiplicity 1)
    Possible multiplicities: $*$ or $a..b$
    a, b are value specifications (numbers expressions); upper bound incl. $*$

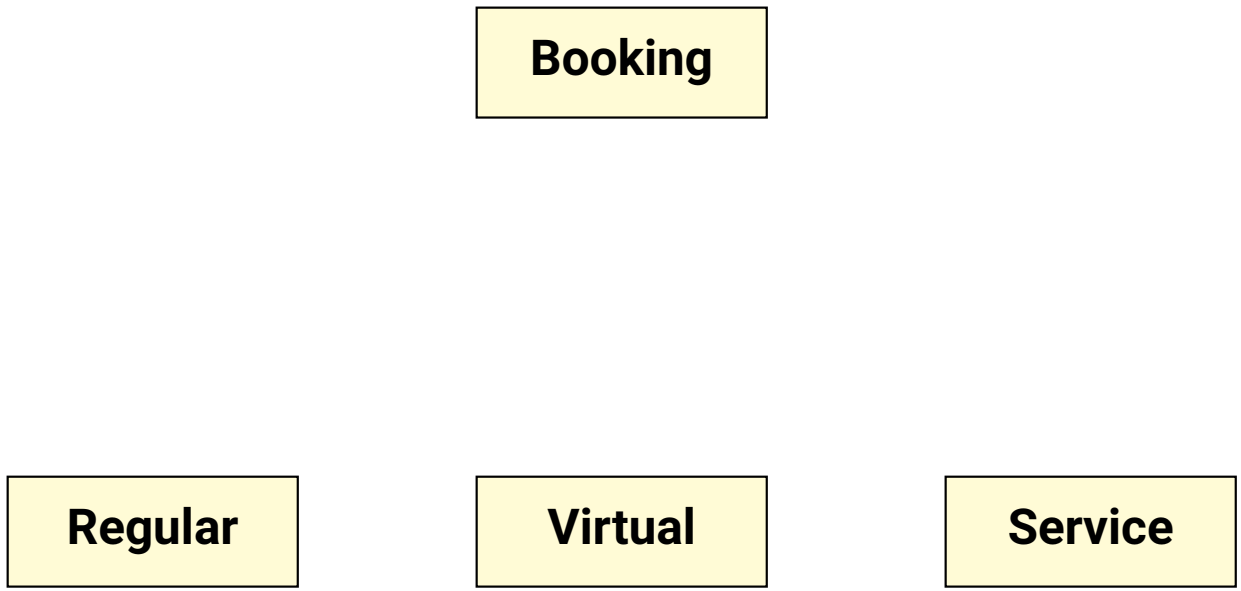
Syntax of UML Association

An association is a relation among classes with the elements:

- Name (optional)
- Two association ends called roles, each with:
  - A role name (defaults to class name starting with lower case letter)
  - A multiplicity (defaults to multiplicity 1)
    Possible multiplicities: $*$ or *a*..*b*
    a, *b* are value specifications (numbers expressions); upper bound incl. $*$
  - A navigability (defaults to bi-directional, not used for conceptual classes)

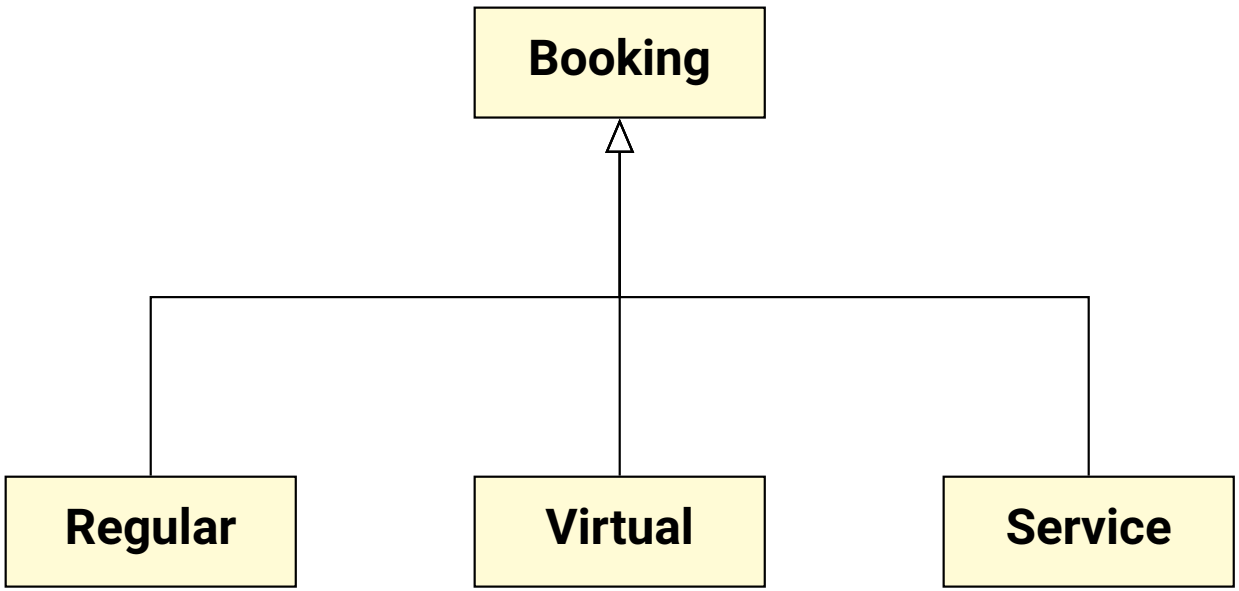


Generalizations of Concepts



- There are different types of bookings: regular, virtual , service

Generalizations of Concepts



- There are different types of bookings: regular, virtual , service

Elicitation of Domain Model for a New Domain: Workflow

TECHNISCHE UNIVERSITÄT DARMSTADT

1. Find the conceptual classes

   Possible Strategies:

   - Re-use or modify an existing model
   - Use a category list
   - Identify noun phrases
2. Draw elicited concepts as classes in a UML class diagram
3. Add attributes
4. Add associations

Use the domain vocabulary:
The CaSh model should use names like Customer instead of Person

Elicitation of Domain Model for a New Domain: Workflow

TECHNISCHE UNIVERSITÄT DARMSTADT

1. Find the conceptual classes

   Possible Strategies:

   - Re-use or modify an existing model
   - Use a category list
   - Identify noun phrases
2. Draw elicited concepts as classes in a UML class diagram
3. Add attributes
4. Add associations

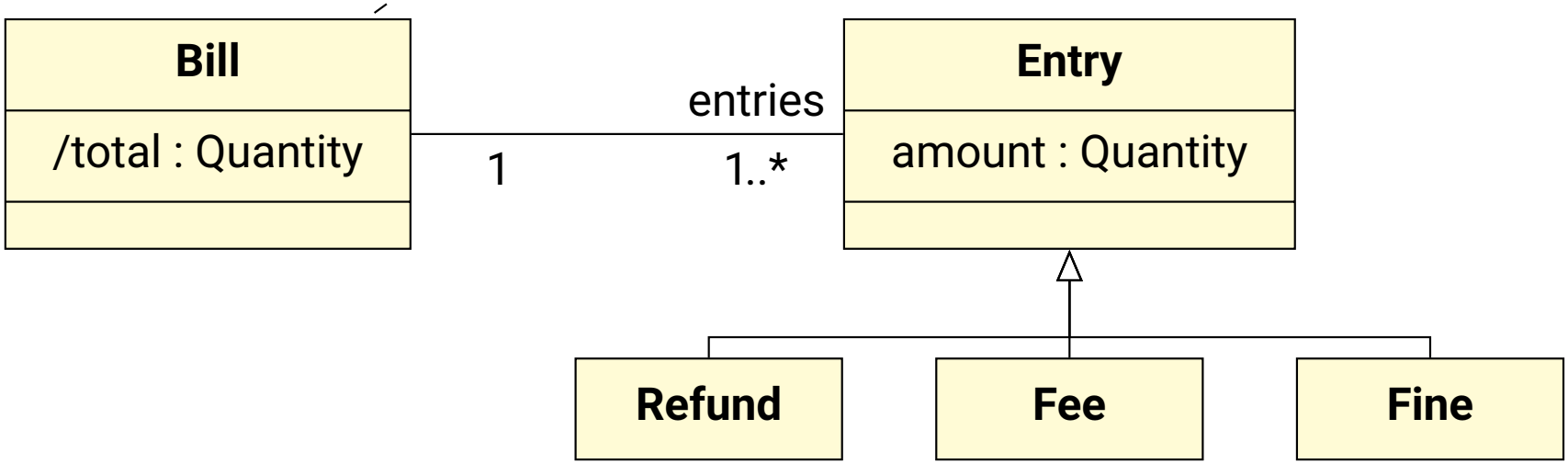
Use the domain vocabulary:
The CaSh model should use names like Customer instead of Person

Re-using an Existing Model

TECHNISCHE UNIVERSITÄT DARMSTADT

for any object self of type **Bill**: $\text{self.total} = \sum_{s \in \text{self.entries}} s.\text{amount}$

[[in OCL: self.total = self.entries -> collect(s | s.amount) -> sum()]]



(For example: M. Fowler. Analysis Patterns—Reusable Object Models, Addison-Wesley, 1997)

Software Engineering Group

Conceptual Class Discovery Technique: List of Conceptual Class Categories

TECHNISCHE UNIVERSITÄT DARMSTADT

Some Categories:

- Physical or tangible objects
- Specifications or description of things
- Locations
- Events
- Transactions
- Transaction items
- Organizations
- Roles (of people, organizations, etc.)
- Containers
- . . .

Using a Category: Example

TECHNISCHE UNIVERSITÄT DARMSTADT

Conceptual Class Category

Conceptual Classes (in CaSh)

Software Engineering Group

Using a Category: Example

TECHNISCHE UNIVERSITÄT DARMSTADT

Conceptual Class Category	Conceptual Classes (in CaSh)
Business transaction	Bill, Payment
Transaction line item	Entry
Product or service related to a transaction or to a transaction line item	Refund, Rent, Fine
Place where transaction is recorded	Registry
Roles of people or organizations related to a transaction (actors in use cases)	Customer, Accountant
Location where transaction executed	Website
Noteworthy events, with a time or place that needs to be remembered	Bill, Booking

Software Engineering Group

Conceptual Class Discovery Technique: Noun Identification

TECHNISCHE UNIVERSITÄT DARMSTADT

Workflow

- Identify nouns and noun phrases in textual descriptions of domain
- Consider them as a candidate for a conceptual class or an attribute

Software Engineering Group

Conceptual Class Discovery Technique: Noun Identification

TECHNISCHE UNIVERSITÄT DARMSTADT

Workflow

- Identify nouns and noun phrases in textual descriptions of domain
- Consider them as a candidate for a conceptual class or an attribute

Can it be automated?

Software Engineering Group

Conceptual Class Discovery Technique: Noun Identification

TECHNISCHE UNIVERSITÄT DARMSTADT

Workflow

- Identify nouns and noun phrases in textual descriptions of domain
- Consider them as a candidate for a conceptual class or an attribute

Can it be automated?

Only partially: Words in natural language texts are ambiguous

- Same noun can mean multiple things
- Different nouns can mean the same thing

LLMs might be helpful, but need to validate

Conceptual Class Discovery Technique: Noun Identification

TECHNISCHE UNIVERSITÄT DARMSTADT

Workflow

- Identify nouns and noun phrases in textual descriptions of domain
- Consider them as a candidate for a conceptual class or an attribute

Can it be automated?

Only partially: Words in natural language texts are ambiguous

- Same noun can mean multiple things
- Different nouns can mean the same thing

LLMs might be helpful, but need to validate

Use Cases (stories) are a canonical source for identifying conceptual classes

Example:
Noun Identification

TECHNISCHE UNIVERSITÄT DARMSTADT

Text Story: Change Existing Booking

The customer selects the booking to change. The system displays the booking details, including adjacent reservations. The customer is offered to change the reserved duration. When a new duration is entered, the system checks availability and records the change. In case of no availability, nothing happens and an information message is displayed. Before any change happens, a confirmation action is requested. Afterwards, a confirmation message is sent to the customer's preferred contact.

Example:
Noun Identification

TECHNISCHE UNIVERSITÄT DARMSTADT

Text Story: Change Existing **Booking**

The customer selects the booking to change. The system displays the booking details, including adjacent reservations. The customer is offered to change the reserved duration. When a new duration is entered, the system checks availability and records the change. In case of no availability, nothing happens and an information message is displayed. Before any change happens, a confirmation action is requested. Afterwards, a confirmation message is sent to the customer's preferred contact.

Software Engineering Group

Example: Noun Identification

TECHNISCHE UNIVERSITÄT DARMSTADT

Text Story: Change Existing Booking

The customer selects the booking to change. The system displays the booking details, including adjacent reservations. The customer is offered to change the reserved duration. When a new duration is entered, the system checks availability and records the change. In case of no availability, nothing happens and an information message is displayed. Before any change happens, a confirmation action is requested. Afterwards, a confirmation message is sent to the customer's preferred contact.

Candidate Conceptual Classes

Customer, Booking, Detail, Reservation, Duration, System, Availability, Change, InformationMessage, ConfirmationAction, ConfirmationMessage, Contact

Selection of Suitable Conceptual Classes

Which nouns should become conceptual classes in domain model?

Example: Should ConfirmationMessage be a conceptual class?

Selection of Suitable Conceptual Classes

TECHNISCHE UNIVERSITÄT DARMSTADT

Which nouns should become conceptual classes in domain model?

Example: Should ConfirmationMessage be a conceptual class?

Criteria to include a candidate conceptual class:

- Must carry information not available/computable from other sources
  Avoid candidates that add only
  - redundant information (available elsewhere)
  - derived information
- Must have specific semantics in relation to the business

Selection of Suitable Conceptual Classes

TECHNISCHE UNIVERSITÄT DARMSTADT

Which nouns should become conceptual classes in domain model?

Example: Should ConfirmationMessage be a conceptual class?

Criteria to include a candidate conceptual class:

- Must carry information not available/computable from other sources
  Avoid candidates that add only
  - redundant information (available elsewhere)
  - derived information
- Must have specific semantics in relation to the business

What does that mean for ConfirmationMessage?

Selection of Suitable Conceptual Classes

TECHNISCHE UNIVERSITÄT DARMSTADT

Which nouns should become conceptual classes in domain model?

Example: Should ConfirmationMessage be a conceptual class?

Criteria to include a candidate conceptual class:

- Must carry information not available/computable from other sources
  Avoid candidates that add only
  - redundant information (available elsewhere)
  - derived information
- Must have specific semantics in relation to the business

What does that mean for ConfirmationMessage?

- A confirmation message repeats content from changed booking

Selection of Suitable Conceptual Classes

TECHNISCHE UNIVERSITÄT DARMSTADT

Which nouns should become conceptual classes in domain model?

Example: Should ConfirmationMessage be a conceptual class?

Criteria to include a candidate conceptual class:

- Must carry information not available/computable from other sources
  Avoid candidates that add only
  - redundant information (available elsewhere)
  - derived information
- Must have specific semantics in relation to the business

What does that mean for ConfirmationMessage?

- A confirmation message repeats content from changed booking

Conclusion: Confirmation Message is report about a booking

Boundary of Domain Model

TECHNISCHE UNIVERSITÄT DARMSTADT

Report: Part of the domain model? — It depends!

Boundary of Domain Model

TECHNISCHE UNIVERSITÄT DARMSTADT

Report: Part of the domain model? — It depends!

- Considering current scenario only:
  ⇒ Confirmation message not part of domain model (only a report)

Boundary of Domain Model

TECHNISCHE UNIVERSITÄT DARMSTADT

Report: Part of the domain model? — It depends!

- Considering current scenario only:
  ⇒ Confirmation message not part of domain model (only a report)
- Considering, for example, handling of disputes:
  ⇒ Confirmation message represents an important concept on its own

Boundary of Domain Model

TECHNISCHE UNIVERSITÄT DARMSTADT

Report: Part of the domain model? — It depends!

- Considering current scenario only:
  ⇒ Confirmation message not part of domain model (only a report)
- Considering, for example, handling of disputes:
  ⇒ Confirmation message represents an important concept on its own

In general:
Regarding legal aspects . . . ? ⟵ Non-functional domain requirement (lecture 2)

More In-/Exclusion Criteria for Conceptual Classes

TECHNISCHE UNIVERSITÄT DARMSTADT

Candidate Conceptual Classes

Customer, Booking, Detail, Reservation, Duration, System, Availability, Change, InformationMessage, ConfirmationMessage, ConfirmationAction, Contact

More In-/Exclusion Criteria for Conceptual Classes

TECHNISCHE UNIVERSITÄT DARMSTADT

Candidate Conceptual Classes

Customer, Booking, Detail, Reservation, Duration, System, Availability, Change, InformationMessage, ConfirmationMessage, ConfirmationAction, Contact

1. Already used, discussed

More In-/Exclusion Criteria for Conceptual Classes

TECHNISCHE UNIVERSITÄT DARMSTADT

Candidate Conceptual Classes

Customer, Booking, Detail, Reservation, Duration, System, Availability, Change, InformationMessage, ConfirmationMessage, ConfirmationAction, Contact

1. Already used, discussed
2. Summary expression actually referring to Car, Duration, ...

More In-/Exclusion Criteria for Conceptual Classes

TECHNISCHE UNIVERSITÄT DARMSTADT

Candidate Conceptual Classes

Customer, Booking, Detail, Reservation, Duration, System, Availability, Change, InformationMessage, ConfirmationMessage, ConfirmationAction, Contact

1. Already used, discussed
2. Summary expression actually referring to Car, Duration, ...
3. Synonym of Booking

More In-/Exclusion Criteria for Conceptual Classes

TECHNISCHE UNIVERSITÄT DARMSTADT

Candidate Conceptual Classes

Customer, Booking, Detail, Reservation, Duration, System, Availability, Change, InformationMessage, ConfirmationMessage, ConfirmationAction, Contact

1. Already used, discussed
2. Summary expression actually referring to Car, Duration, ...
3. Synonym of Booking
4. Root class, not a specific concept

More In-/Exclusion Criteria for Conceptual Classes

TECHNISCHE UNIVERSITÄT DARMSTADT

Candidate Conceptual Classes

Customer, Booking, Detail, Reservation, Duration, System, Availability, Change, InformationMessage, ConfirmationMessage, ConfirmationAction, Contact

1. Already used, discussed
2. Summary expression actually referring to Car, Duration, ...
3. Synonym of Booking
4. Root class, not a specific concept
5. Derivable from Duration and bookings

More In-/Exclusion Criteria for Conceptual Classes

TECHNISCHE UNIVERSITÄT DARMSTADT

Candidate Conceptual Classes

Customer, Booking, Detail, Reservation, Duration, System, Availability, Change, InformationMessage, ConfirmationMessage, ConfirmationAction, Contact

1. Already used, discussed
2. Summary expression actually referring to Car, Duration, ...
3. Synonym of Booking
4. Root class, not a specific concept
5. Derivable from Duration and bookings
6. Synonym of changed Booking

⋮

Description Classes

TECHNISCHE UNIVERSITÄT DARMSTADT

A description class contains information that describes an entity

Software Engineering Group

Description Classes

TECHNISCHE UNIVERSITÄT DARMSTADT

A description class contains information that describes an entity

Example

- A rental location description has information on:
  Address, accessibility, closest public transport, etc.

Description Classes

TECHNISCHE UNIVERSITÄT DARMSTADT

A description class contains information that describes an entity

Example

- A rental location description has information on:
  Address, accessibility, closest public transport, etc.

A description class should be added to the domain model in these cases:

1. Information about an item or service is required,
   regardless of whether any instances of those items or services exist

Description Classes

TECHNISCHE UNIVERSITÄT DARMSTADT

A description class contains information that describes an entity

Example

- A rental location description has information on:
  Address, accessibility, closest public transport, etc.

A description class should be added to the domain model in these cases:

1. Information about an item or service is required,
   regardless of whether any instances of those items or services exist
2. Deleting instances of described entities results in loss of information

Description Classes

TECHNISCHE UNIVERSITÄT DARMSTADT

A description class contains information that describes an entity

Example

- A rental location description has information on:
  Address, accessibility, closest public transport, etc.

A description class should be added to the domain model in these cases:

1. Information about an item or service is required,
   regardless of whether any instances of those items or services exist
2. Deleting instances of described entities results in loss of information
3. Redundant or duplicated information is reduced

Model a Notion as Class or Attribute?

TECHNISCHE UNIVERSITÄT DARMSTADT

Heuristics: Class or Attribute

If we do not think of a notion *C* as a number, text or data of the real world, then *C* is probably a conceptual class, not an attribute

Software Engineering Group

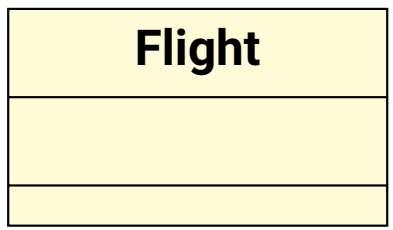
Model a Notion as Class or Attribute?

TECHNISCHE UNIVERSITÄT DARMSTADT

Heuristics: Class or Attribute

If we do not think of a notion *C* as a number, text or data of the real world, then *C* is probably a conceptual class, not an attribute

Example (Airport)



Software Engineering Group

Model a Notion as Class or Attribute?

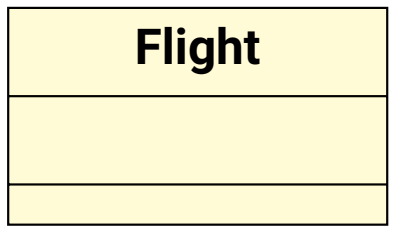
TECHNISCHE UNIVERSITÄT DARMSTADT

Heuristics: Class or Attribute

If we do not think of a notion *C* as a number, text or data of the real world, then *C* is probably a conceptual class, not an attribute

Example (Airport)

- Should destination be an attribute of flight, or a conceptual class airport?



Software Engineering Group

Model a Notion as Class or Attribute?

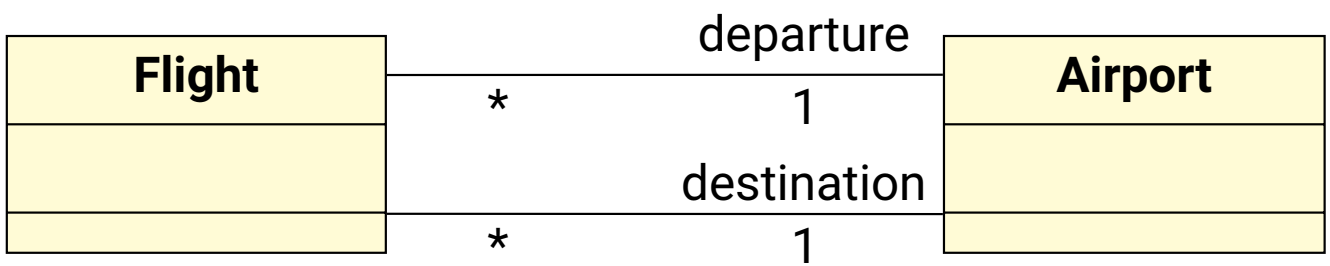
TECHNISCHE UNIVERSITÄT DARMSTADT

Heuristics: Class or Attribute

If we do not think of a notion *C* as a number, text or data of the real world, then *C* is probably a conceptual class, not an attribute

Example (Airport)

- Should destination be an attribute of flight, or a conceptual class airport?
- Name of airport?



Software Engineering Group

Model a Notion as Class or Attribute?

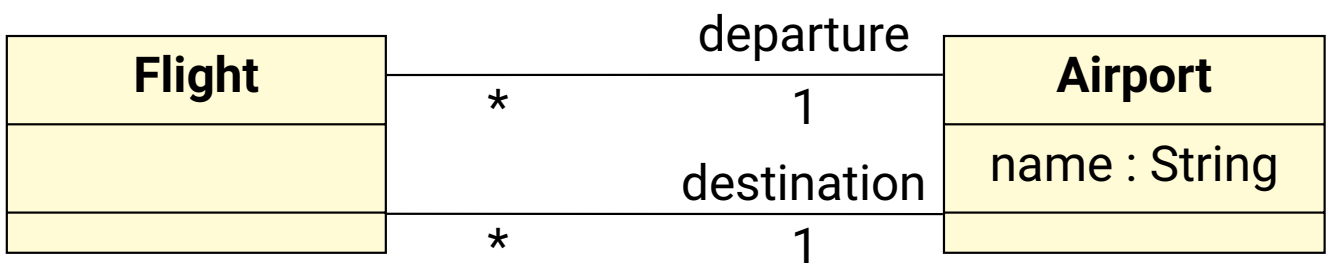
TECHNISCHE UNIVERSITÄT DARMSTADT

Heuristics: Class or Attribute

If we do not think of a notion *C* as a number, text or data of the real world, then *C* is probably a conceptual class, not an attribute

Example (Airport)

- Should destination be an attribute of flight, or a conceptual class airport?
- Name of airport?



Software Engineering Group

Use of Associations in Domain Modelling

TECHNISCHE UNIVERSITÄT DARMSTADT

Heuristics

Include associations in the domain model when:
Knowledge about the relation needs to be preserved for some time

Software Engineering Group

Use of Associations in Domain Modelling

TECHNISCHE UNIVERSITÄT DARMSTADT

Heuristics

Include associations in the domain model when:
Knowledge about the relation needs to be preserved for some time

Example

The relation between a bill and its entries needs to be remembered

But it is unnecessary to store the relation
between a customer and his or her most recent search for available cars

On the other hand, a browser might well record the search history

Use of Associations in Domain Modelling

TECHNISCHE UNIVERSITÄT DARMSTADT

Heuristics

Include associations in the domain model when:
Knowledge about the relation needs to be preserved for some time

List of common association categories:

- *A* is a transaction related to another transaction *B*
- *A* is a product or a service for a transaction *B*
- *A* is a role related to a transaction *B*
- *A* is a physical or logical part of *B*
  For example, *A* is a line item of a transaction *B*

Association Name

TECHNISCHE UNIVERSITÄT DARMSTADT

Name an association based on a Class name-Verb phrase-Class name format

The verb phrase creates a sequence that is readable and meaningful

Software Engineering Group

Association Name

TECHNISCHE UNIVERSITÄT DARMSTADT

Name an association based on a Class name-Verb phrase-Class name format

The verb phrase creates a sequence that is readable and meaningful

Good examples:

- Player-Stands-on-Square
- Sale-Paid-by-CashPayment

Association Name

TECHNISCHE UNIVERSITÄT DARMSTADT

Name an association based on a Class name-Verb phrase-Class name format

The verb phrase creates a sequence that is readable and meaningful

Good examples:

- Player-Stands-on-Square
- Sale-Paid-by-CashPayment

Bad examples:

- Player-Has-Square (“Has” is generic, doesn’t tell about relation)
- Sale-Uses-CashPayment (“Uses” dito)

Attribute or Association?

TECHNISCHE UNIVERSITÄT DARMSTADT

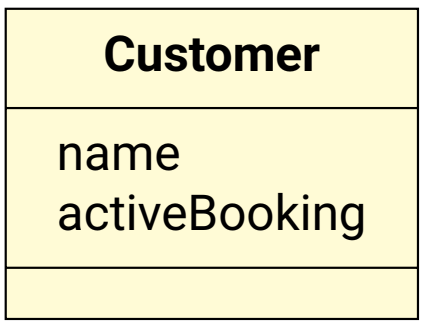
A property can be modelled as an association or an attribute

The notations are very different! — When to use what?

Software Engineering Group

Attribute or Association?

TECHNISCHE UNIVERSITÄT DARMSTADT



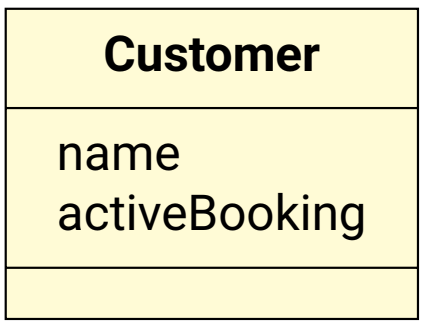
?

Software Engineering Group

Attribute or Association?

TECHNISCHE UNIVERSITÄT DARMSTADT

- Attributes should always be used for primitive datatypes
  - Boolean, Integer, Character, String
  - Date, Address, Color, Phone Number, . . .



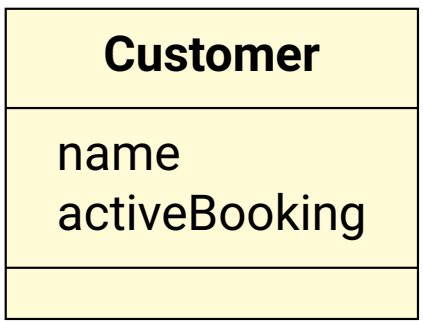
?

Software Engineering Group

Attribute or Association?

TECHNISCHE UNIVERSITÄT DARMSTADT

- Attributes should always be used for primitive datatypes
  - Boolean, Integer, Character, String
  - Date, Address, Color, Phone Number, . . .
- Quantities may be modelled as classes to attach units
  - currency (EUR, USD, SEK, . . .)
  - weight (kilogram, cm, . . .)



?

Software Engineering Group

Attribute or Association?

TECHNISCHE UNIVERSITÄT DARMSTADT

- Attributes should always be used for primitive datatypes
  - Boolean, Integer, Character, String
  - Date, Address, Color, Phone Number, . . .
- Quantities may be modelled as classes to attach units
  - currency (EUR, USD, SEK, . . .)
  - weight (kilogram, cm, . . .)
- Always use associations to model relations between conceptual classes

Software Engineering Group

Refining a Model: Replace String Type

TECHNISCHE UNIVERSITÄT DARMSTADT

It is convenient to type attributes initially with String:

Generic type that avoids premature decision

Later, consider refinement to description class

Car
model : String

Software Engineering Group

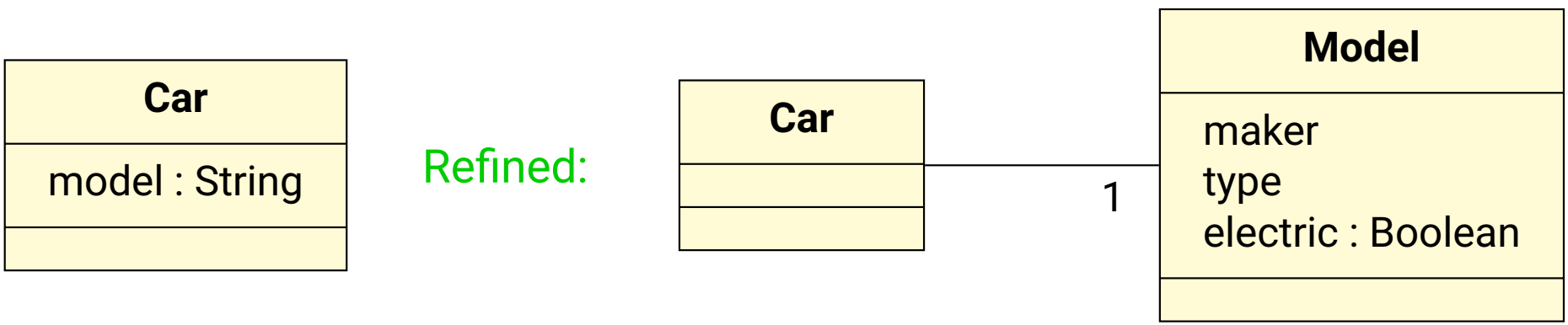
Refining a Model: Replace String Type

TECHNISCHE UNIVERSITÄT DARMSTADT

It is convenient to type attributes initially with String:

Generic type that avoids premature decision

Later, consider refinement to description class



Software Engineering Group

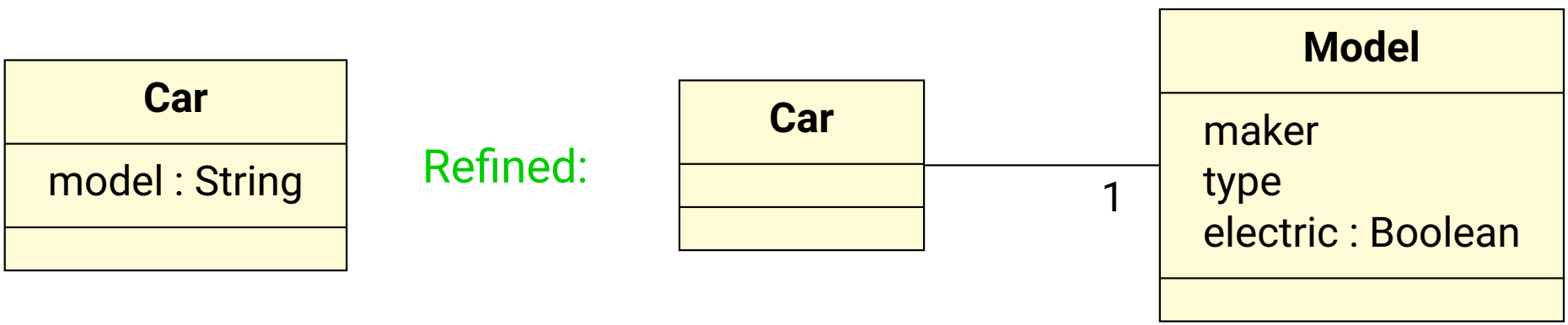
Refining a Model: Replace String Type

TECHNISCHE UNIVERSITÄT DARMSTADT

It is convenient to type attributes initially with String:

Generic type that avoids premature decision

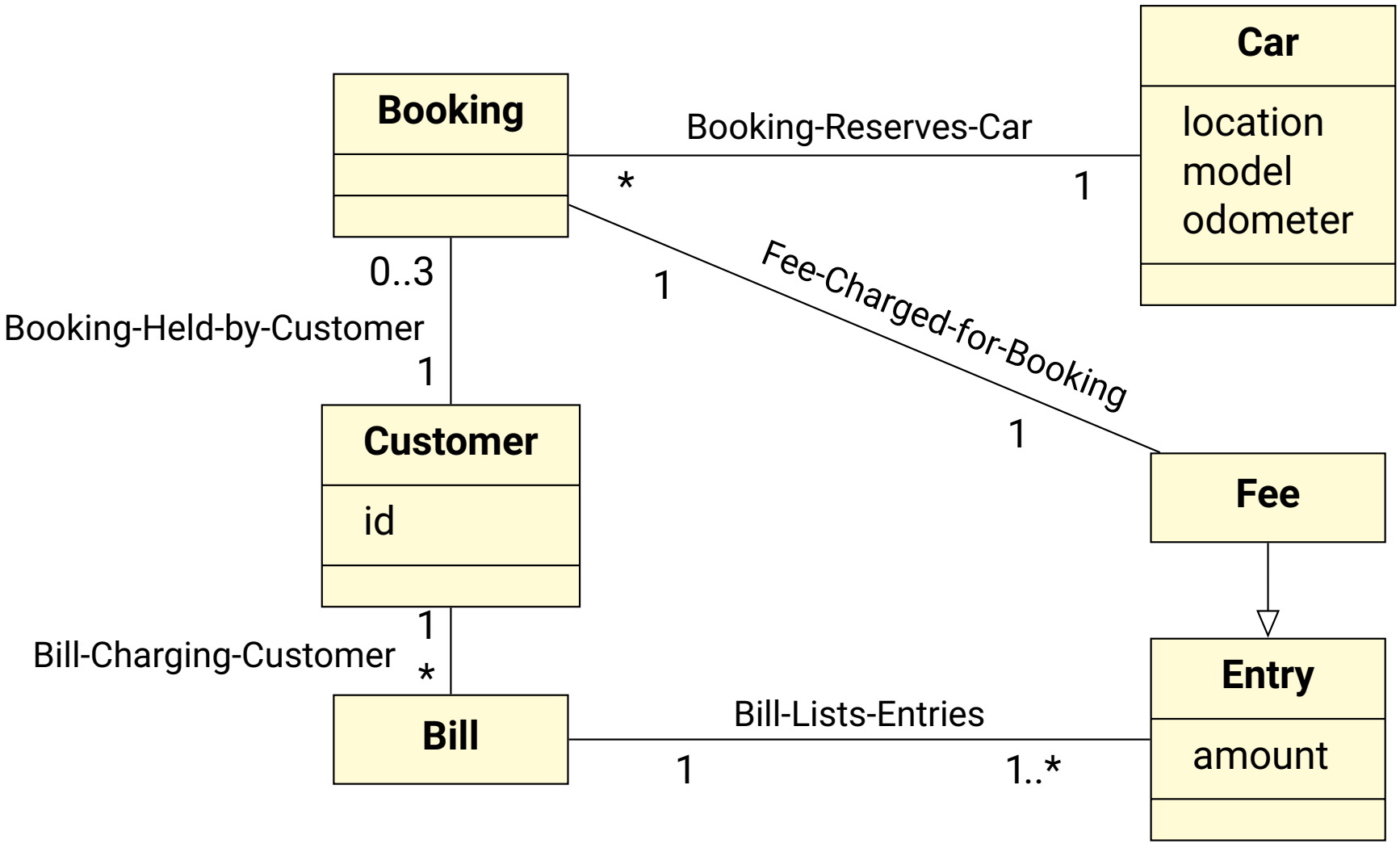
Later, consider refinement to description class



- When string has structure, i.e. composed of separate sections: Maker, type
- When other relations are associated with the string: social security nr
- When the string has other attributes: electric
- When the string is a quantity with a unit, for example, a currency

Software Engineering Group

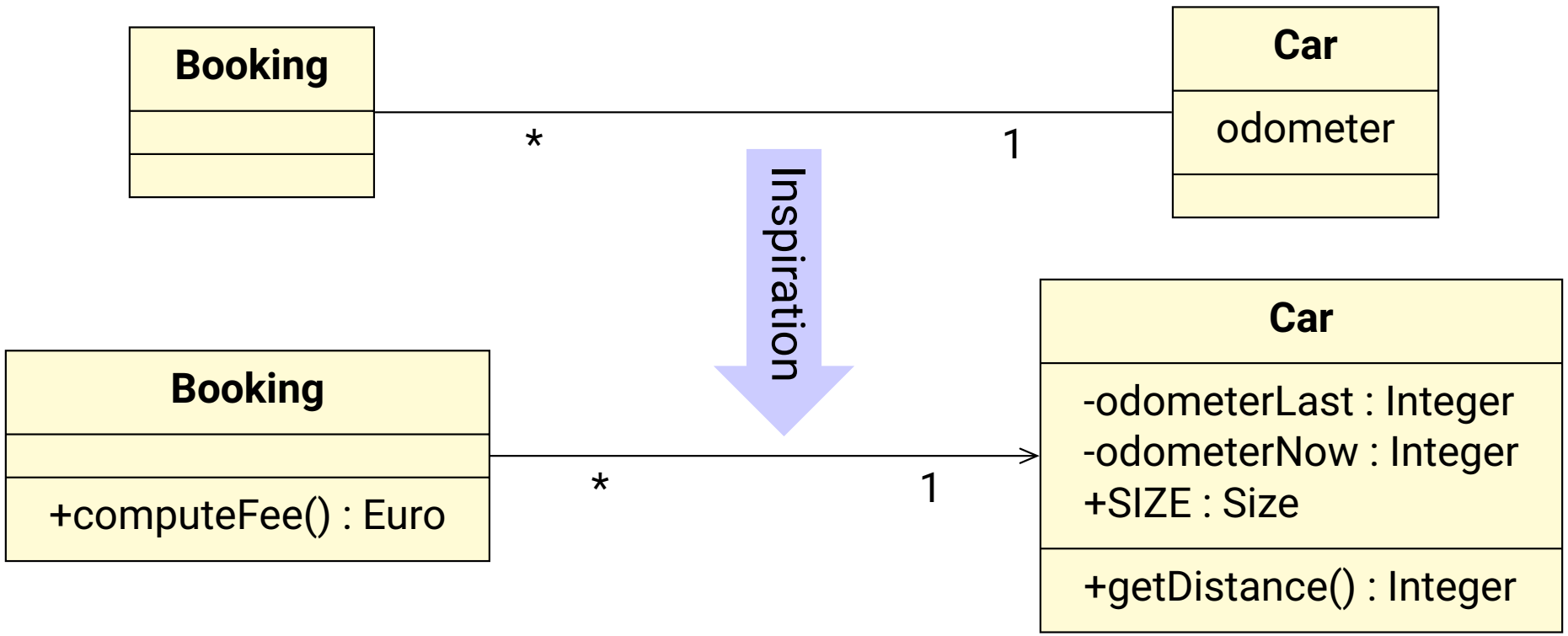
CaSh Booking System: Domain Model (Excerpt)



From Domain Model to Design Model

TECHNISCHE UNIVERSITÄT DARMSTADT

The domain model serves as a source of inspiration for the design model



Software Engineering Group

Domain Model is Not Static View on Code

TECHNISCHE UNIVERSITÄT DARMSTADT

The Two Faces of UML Class Diagrams

1. Conceptual perspective model in domain modelling (our usage)
2. Static view (classes, methods, fields) extracted from OO code

Don’t confuse them, for example, abuse the former as code stubs

Software Engineering Group

Part III

UML State Machine Diagrams as Behavioral Models

Behavioral Domain Modelling

TECHNISCHE UNIVERSITÄT DARMSTADT

Class diagrams as conceptual classes model static aspects of a domain

- Conceptual classes, domain elements
- Properties: attributes, associations
- (Functions)

Software Engineering Group

Behavioral Domain Modelling

TECHNISCHE UNIVERSITÄT DARMSTADT

Class diagrams as conceptual classes model static aspects of a domain

- Conceptual classes, domain elements
- Properties: attributes, associations
- (Functions)

What about behavior, such as scenarios?

- Sequences of actions and how they change the
  - state of a system
  - under which condition

UML State Machine Diagrams

TECHNISCHE UNIVERSITÄT DARMSTADT

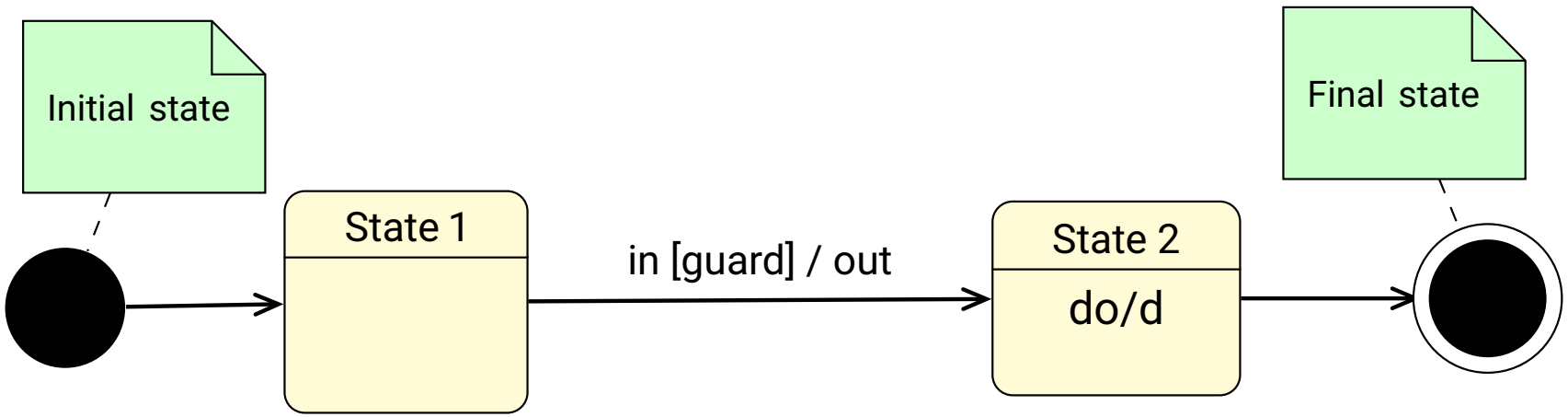
UML State Machine Diagrams are finite state machines
whose transitions are equipped with in-/output actions and guards

Software Engineering Group

UML State Machine Diagrams

TECHNISCHE UNIVERSITÄT DARMSTADT

UML State Machine Diagrams are finite state machines
whose transitions are equipped with in-/output actions and guards

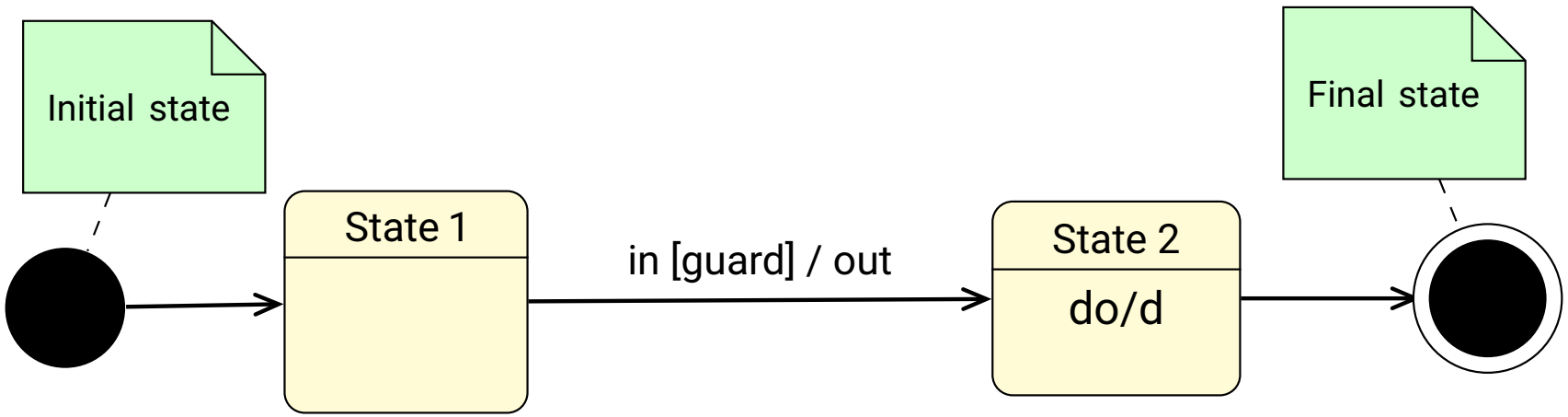


Software Engineering Group

UML State Machine Diagrams

TECHNISCHE UNIVERSITÄT DARMSTADT

UML State Machine Diagrams are finite state machines
whose transitions are equipped with in-/output actions and guards



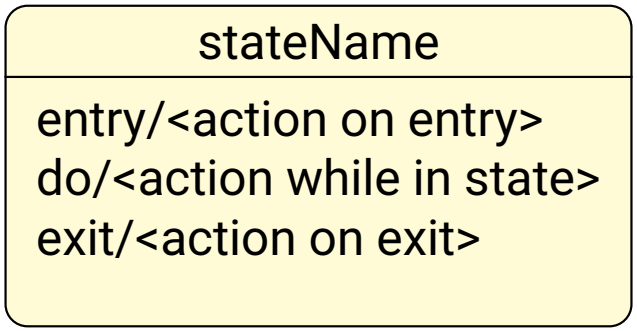
Semantics

When in State 1, if input in is observed and guard is true,
then output out happens and current state becomes State 2.
In State 2 perform (interruptible) action d.

Software Engineering Group

Basic States

TECHNISCHE UNIVERSITÄT DARMSTADT



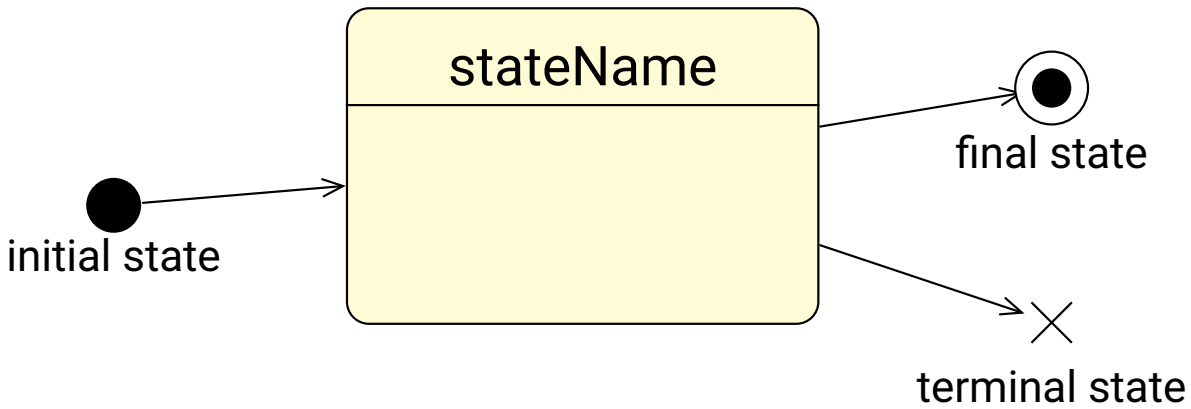
Basic States in Detail

- Basic states have a name and can define any combination of

entry action	action performed on state entry
do action	action performed while in state (until the action terminates or the state is left)
exit action	action performed on state exit

- Actions are executed operations

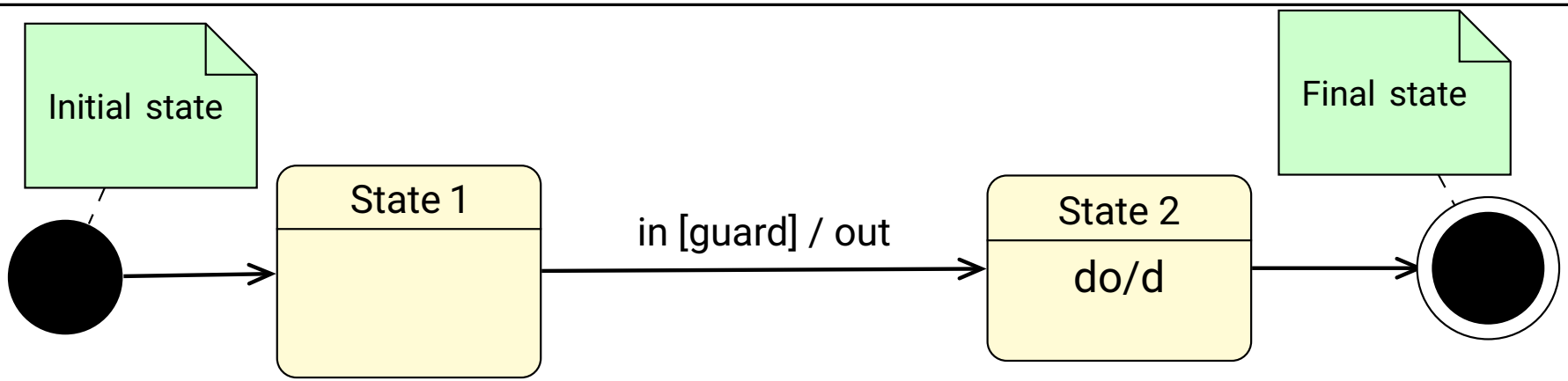
Basic States: Initial, Final & Terminal States



States with Special Meaning

Initial state	has single transition to first entered state (may be labeled by object creation event/message)
Final state	indicates completion of a scenario
Terminal state	completion and executing object destroyed

Transitions between States

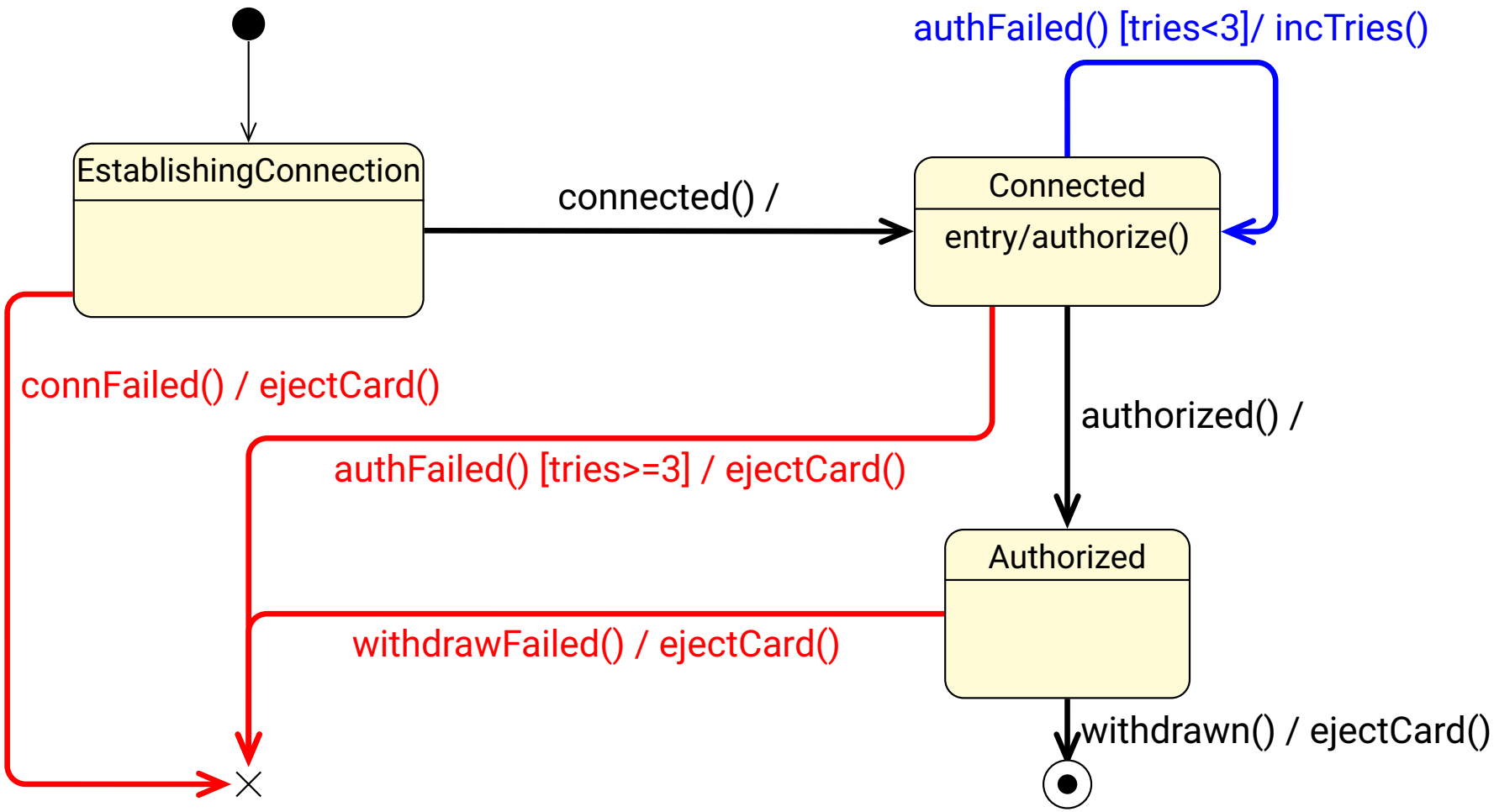


Transition Labels: *input?* (’[’ *guard* ’]’)*?* ’/’ *output?*

- All parts are optional (indicated by ?)
- Input (trigger) events are observations such as:
  - call event (start of operation)
  - time event (for example, time spent in a state)
  - change event (value of an attribute has changed)
- Guard is a Boolean expression
- Output (action) is an operation or domain-specific action expression

State Machine Diagram: Example

Credit Card Payment with PIN



State Machine Diagrams in Domain Modelling

TECHNISCHE UNIVERSITÄT DARMSTADT

During domain modelling: State machine diagrams do not relate to code

Uses of State Machine Diagrams in Domain Modeling

Capture the action sequence of a use case

- Can combine several, related use cases
- Clarify the states an object can be in
- Help to clarify protocols
- Help to validate the domain model
- Help to complete the domain model (with properties / actions)

State machine diagrams should only be used to model non-trivial behavior

Literature

TECHNISCHE UNIVERSITÄT DARMSTADT

- Ian Sommerville, Software Engineering, 10th edition, Pearson Education, 2015 (Chapter 5, in particular Sections 5.3, 5.4.2)

  TUDa ULB eBook (German edition)
- Craig Larman, Applying UML and patterns, 2nd edition, Prentice Hall, 2002

  (Available as hardcover in TUDa ULB, several copies)
- Martin Fowler, UML konzentriert, 3rd edition, Addison-Wesley, 2004

  (Available as paperback in TUDa ULB)
- Martin Fowler, Analysis patterns, Addison-Wesley, 1999

  (Available as hardcover in TUDa ULB)