

Bachelor of Science (BSc) in Informatik
Modul Software-Entwicklung 1 (SWEN1)

LE 04 - Domänenmodellierung

SWEN1/PM3 Team:
R. Ferri (feit), D. Liebhart (lieh), K. Bleisch (bles), G. Wyder (wydg)

Ausgabe: HS24

Um was geht es?

- Anforderungen können besser verstanden und umgesetzt werden, wenn man eine klare Vorstellung von der Fachdomäne hat.
- Die Erfahrung hat gezeigt, dass es eine gute Wahl ist, wenn die Software so strukturiert wird wie die Fachdomäne.
- Die statischen Aspekte einer Fachdomäne können mit einem vereinfachten Klassendiagramm modelliert werden.

Lernziele LE 04 – Domänenmodellierung

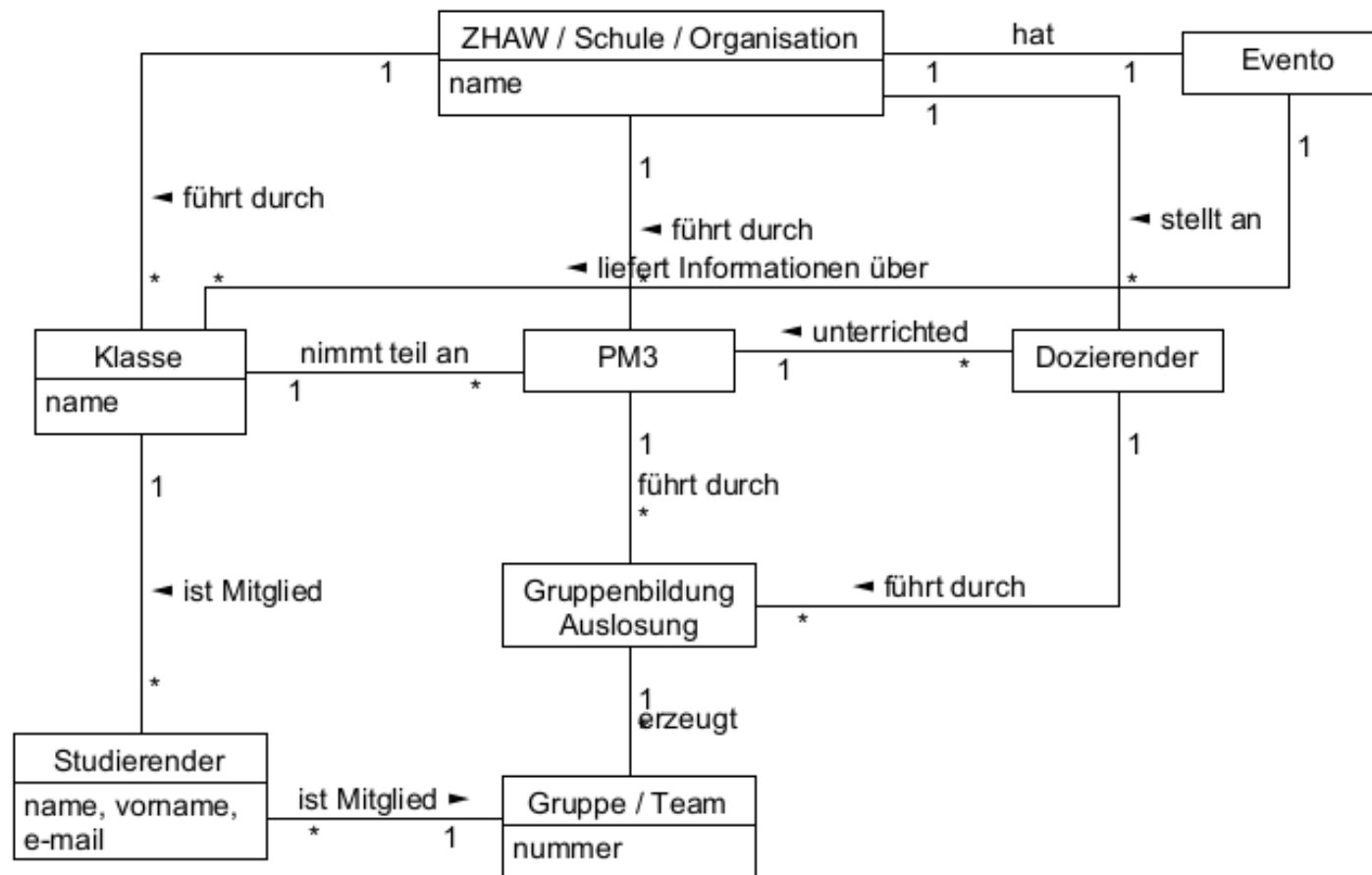
Sie sind in der Lage:

- Ein vereinfachtes **UML-Klassendiagramm** zu zeichnen,
- Ein Modell der **Fachdomäne** in Form eines UML-Klassendiagramms zu erstellen,
- **Konzepte** der Fachdomäne in Anforderungen zu identifizieren, in **Beziehung** zueinander zu setzen und mit sinnvollen **Attributen** zu versehen,
- Beschreibungskonzepte, Generalisierungen/Spezialisierungen, Kompositionen, Rollen und Assoziationsklassen zu identifizieren und korrekt in UML abzubilden.

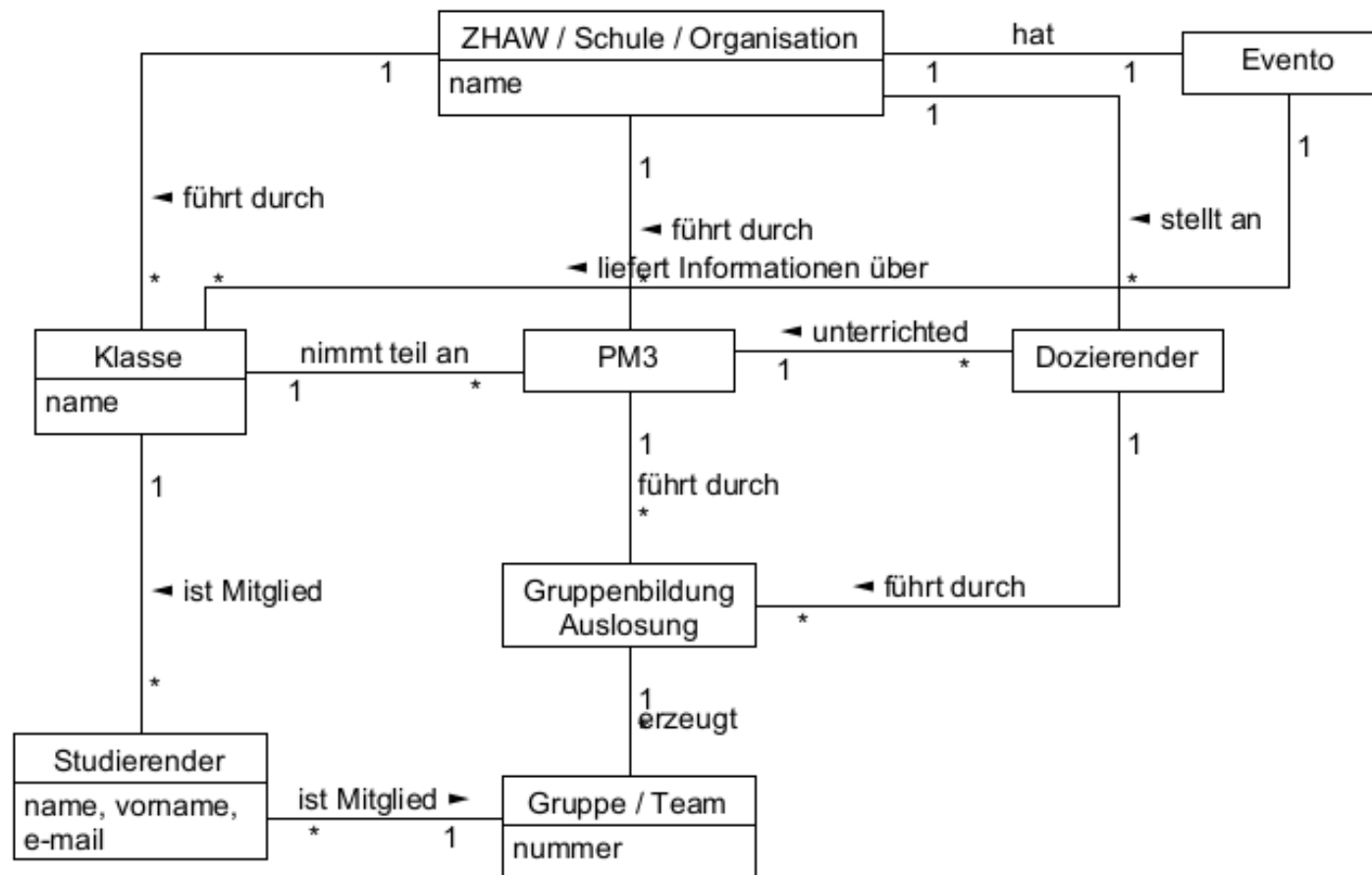
Agenda

1. Einleitung und Motivation
2. Grundlagen
3. Vorgehen
4. Analyse Muster
5. Wrap-up und Ausblick

Zur Erinnerung: Auslosungstool



Zur Erinnerung: Auslosungstool

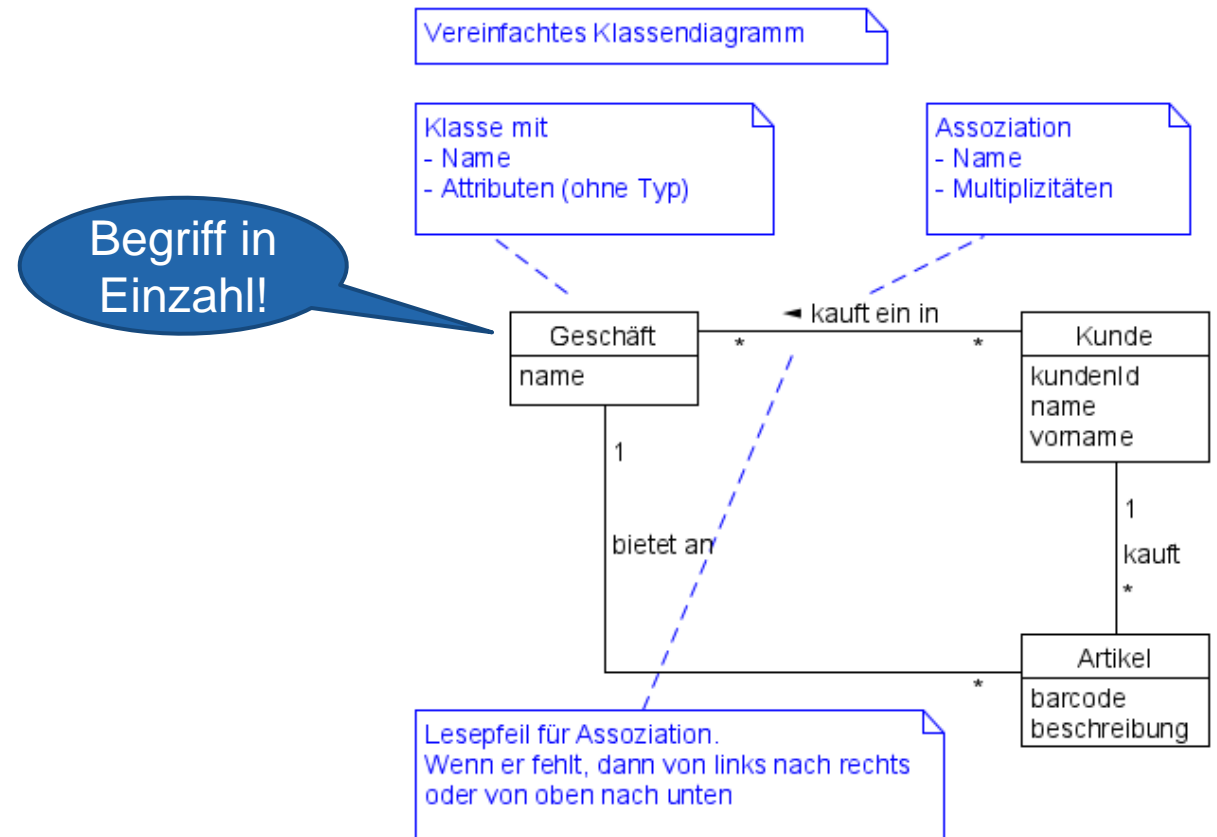


Agenda

1. Einleitung und Motivation
2. Grundlagen
3. Vorgehen
4. Analyse Muster
5. Wrap-up und Ausblick

Domänenmodell als vereinfachtes UML Klassendiagramm

- Das Domänenmodell wird als **UML Klassendiagramm** in einer vereinfachten Form gezeichnet.
- **Konzepte** werden als Klassen modelliert.
- Eigenschaften von Konzepten werden mit **Attributen** modelliert. Die Typangabe entfällt üblicherweise.
- **Assoziationen** werden verwendet, um Beziehungen zwischen Konzepten zu modellieren. Dabei beschreibt der Name der Assoziation die Beziehung und an beiden Enden werden **Multiplizitäten** angeschrieben.



Agenda

1. Einleitung und Motivation
2. Grundlagen
3. Vorgehen
4. Analyse Muster
5. Wrap-up und Ausblick

Vorgehen

- Zuerst werden die **Konzepte** identifiziert
 - Eigenes oder fremdes Fachwissen und Erfahrung verwenden
 - Substantive aus Anwendungsfällen herausziehen
 - Kategorienliste verwenden

Vorgehen

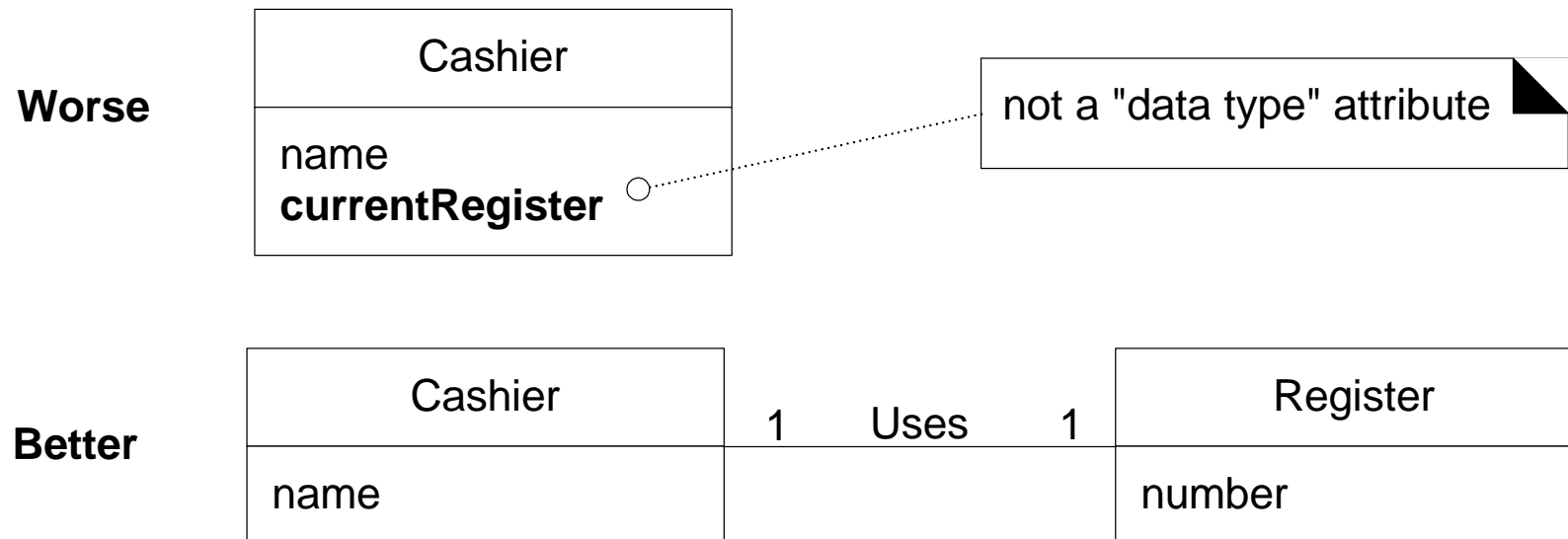
- Zuerst werden die **Konzepte** identifiziert
 - Eigenes oder fremdes Fachwissen und Erfahrung verwenden
 - Substantive aus Anwendungsfällen herausziehen
 - Kategorienliste verwenden
- Konzepte mit **Attributen** versehen
 - Fachwissen
- Konzepte in **Verbindung** zueinander setzen
 - Fachwissen
 - Kategorienliste verwenden
- Dabei Auftraggeber und/oder Fachexperten beiziehen
- Vorgehensweise eines Kartografen anwenden

Datentypen von Attributen

- Die meisten Attributtypen sind **einfach** («primitiv»).
- Integer, float, boolean
- Werden im DM **normalerweise nicht angegeben**
- Attributtypen können auch zusammengesetzte Typen sein
- Nur ihr Inhalt und nicht ihre Identität ist relevant.
- Die Java Typen `String` und `Instant` sind solche Typen.
 - Vergleich mit `equals (...)` und nicht mit `==`

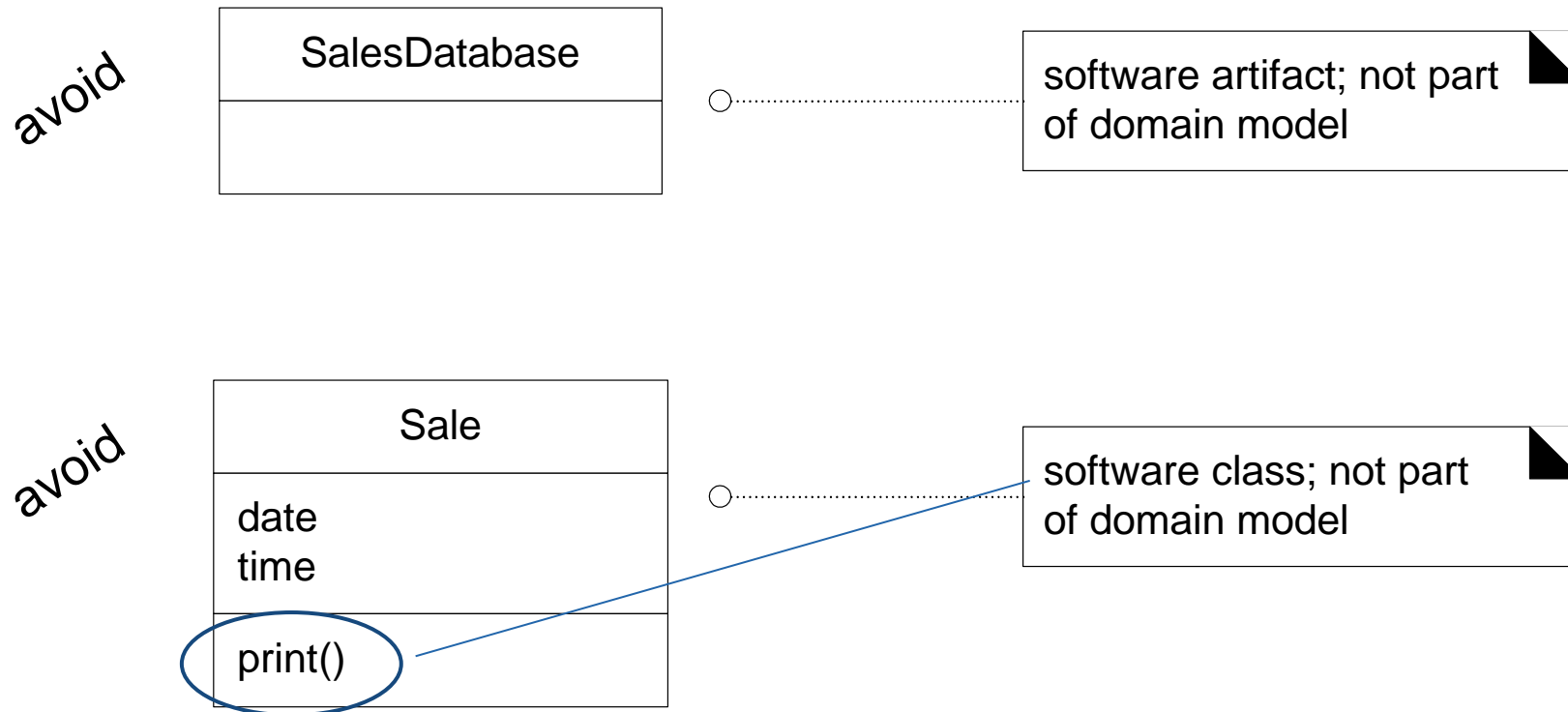
Anti-Pattern: Attribute an Stelle von Assoziationen

- Verwenden Sie **Assoziationen** und nicht Attribute, um Konzepte in Beziehung zueinander zu setzen.



Anti-Pattern : Software-Klassen

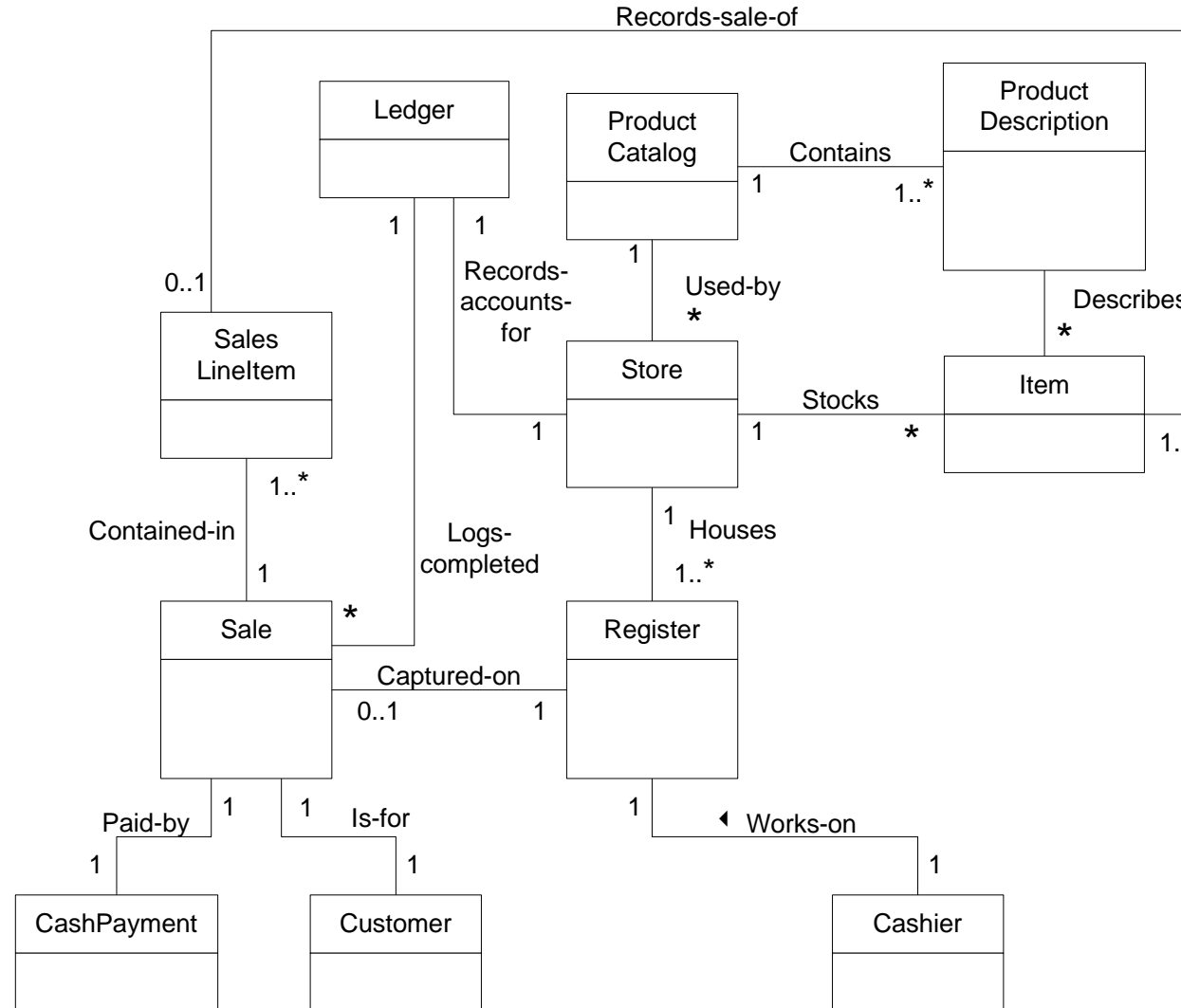
- Keine **Software** Klassen im Domänenmodell, die es so nicht in der Fachdomäne gibt.



Ein paar Bemerkungen zur Domänenmodellierung

- Das perfekte Domänenmodell gibt es so nicht.
- Es ist immer eine Annäherung an den Fachbereich.
- Werkzeug fürs
 - Verstehen der Fachdomäne
 - Kommunikation im Team und mit dem Auftraggeber

Domänenmodell für die elektronische Kasse



Agenda

1. Einleitung und Motivation
2. Grundlagen
3. Vorgehen
4. **Analysemuster**
5. Wrap-up und Ausblick

Analysemuster

- Beschreibungsklassen
- Generalisierung / Spezialisierung
- Komposition
- Zustände
- Rollen
- Assoziationsklasse
- Einheiten
- Zeitintervalle

Beschreibungsklassen

- Ein Artikel ist ein physischer Gegenstand oder eine Dienstleistung, die ein Kunde kaufen kann.
- Ein Geschäft hat typischerweise mehrere Artikel vom selben Typ in den Verkaufsregalen.
- Ein Artikel hat zumindest die Attribute Beschreibung, Preis, Serie Nummer und einen Code, der als Barcode auf der Verpackung aufgedruckt wird.

Item
description price serial number itemID

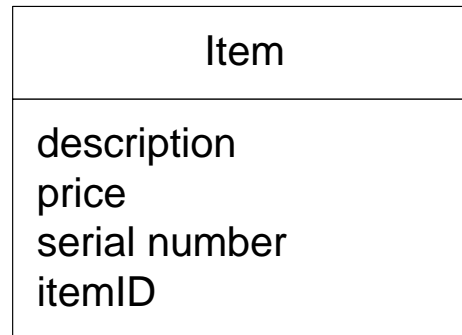
Aufgabe 4.4 (5')

Diskutieren Sie in Murmelgruppen folgende Fragen:

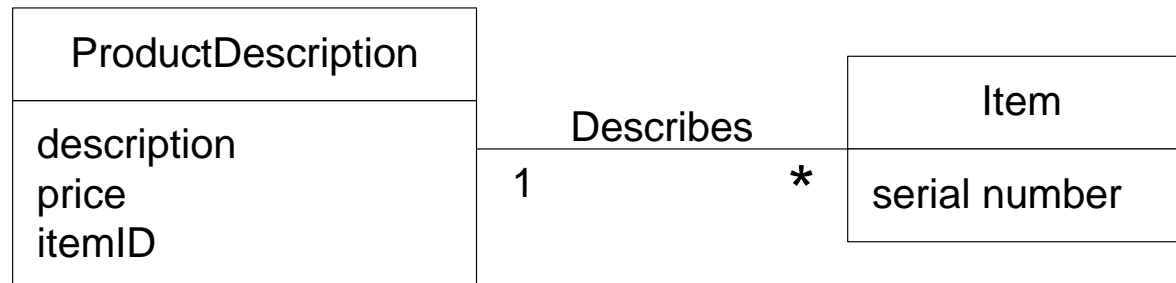
- Wenn dieses Modell so für die Software übernommen wird, wie steht es um die Redundanz?
- Was passiert, wenn alle Artikel von einem Typ verkauft sind?
- Wie könnte ein verbessertes Modell aussehen?

Beschreibungsklasse für Artikel

- Attribute, die für alle Artikel eines Typs gleich sind, werden in eine eigene Klasse herausgezogen.



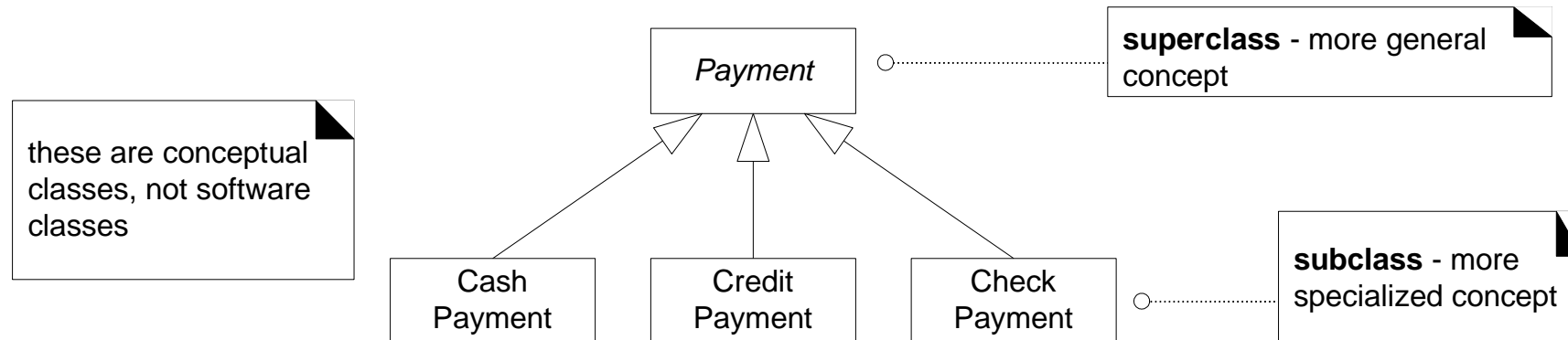
Worse



Better

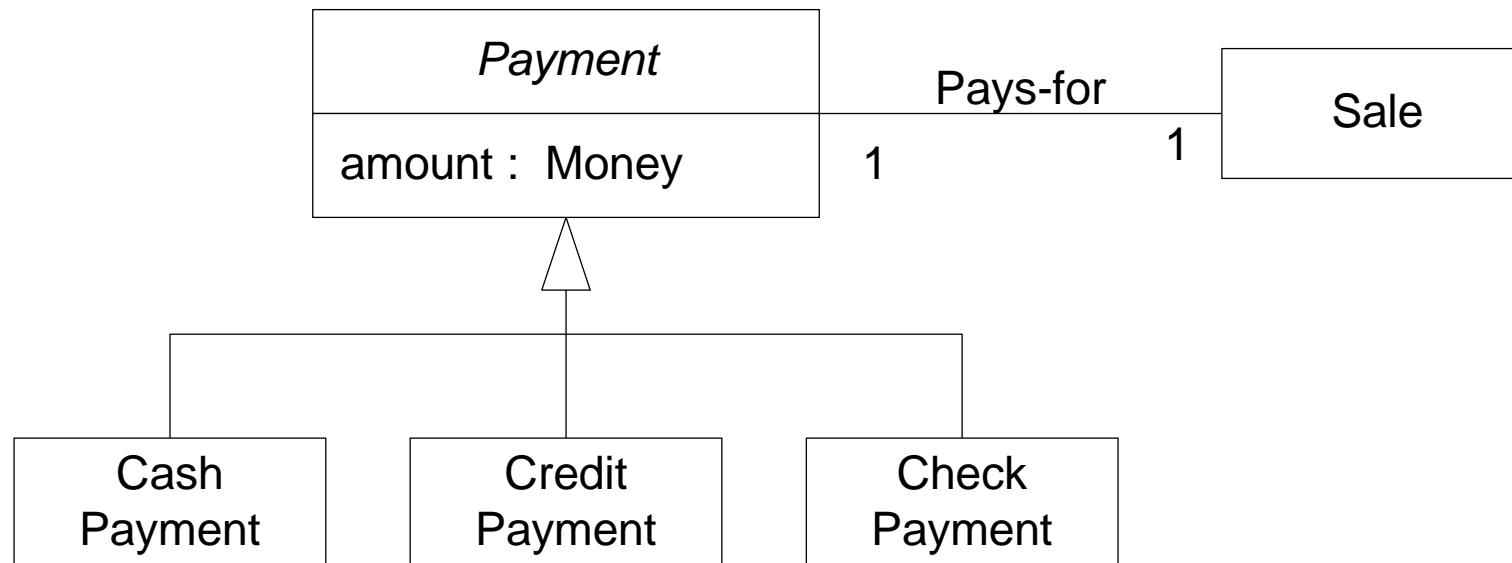
Generalisierung und Spezialisierung in der Fallstudie

- Es gibt verschiedene Zahlungsmöglichkeiten: Bar, Kreditkarte, Check



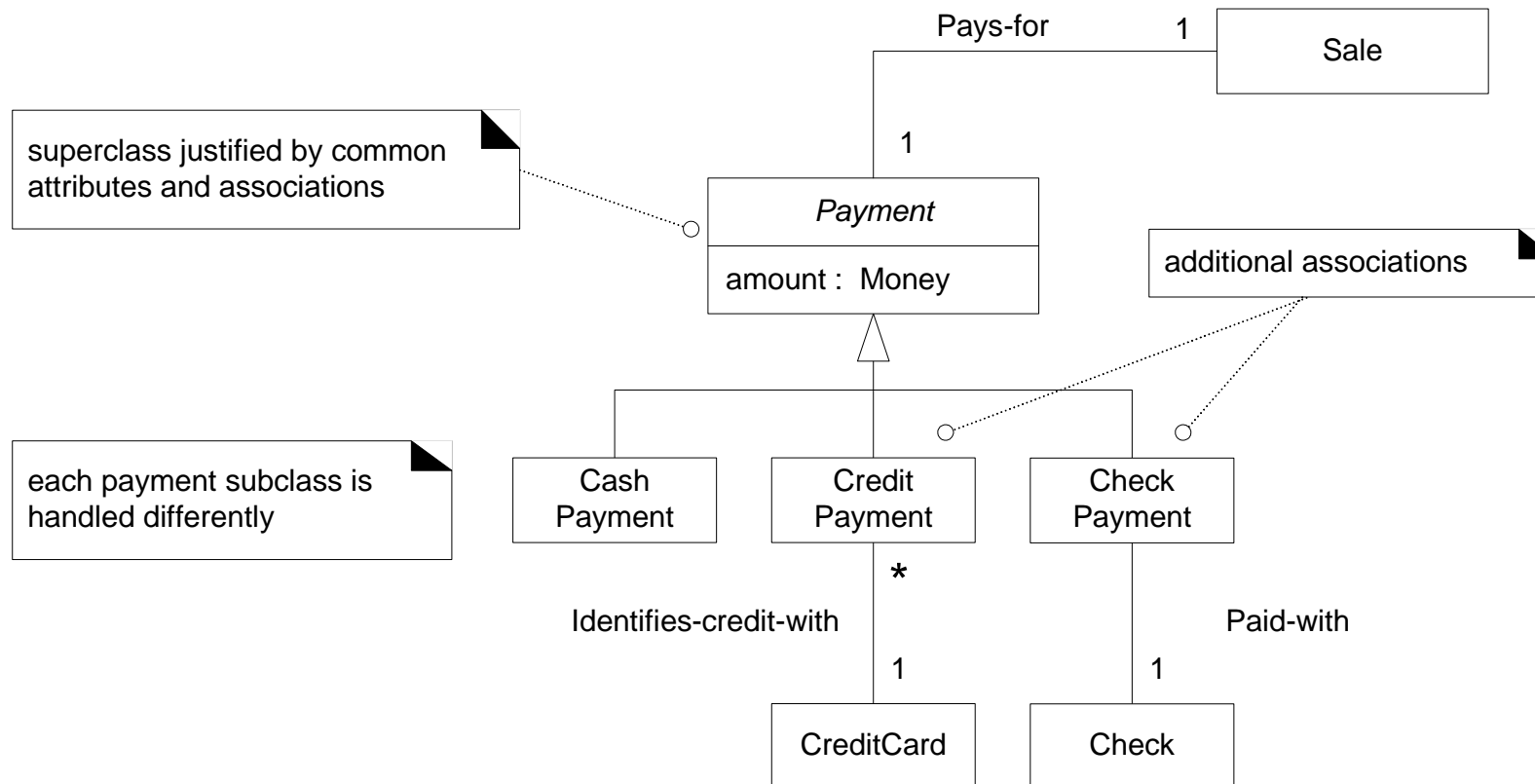
Generalisierung und Spezialisierung in der Fallstudie

- Assoziationen und Attribute der generalisierten Klasse werden an die spezialisierten Klassen weitergegeben.



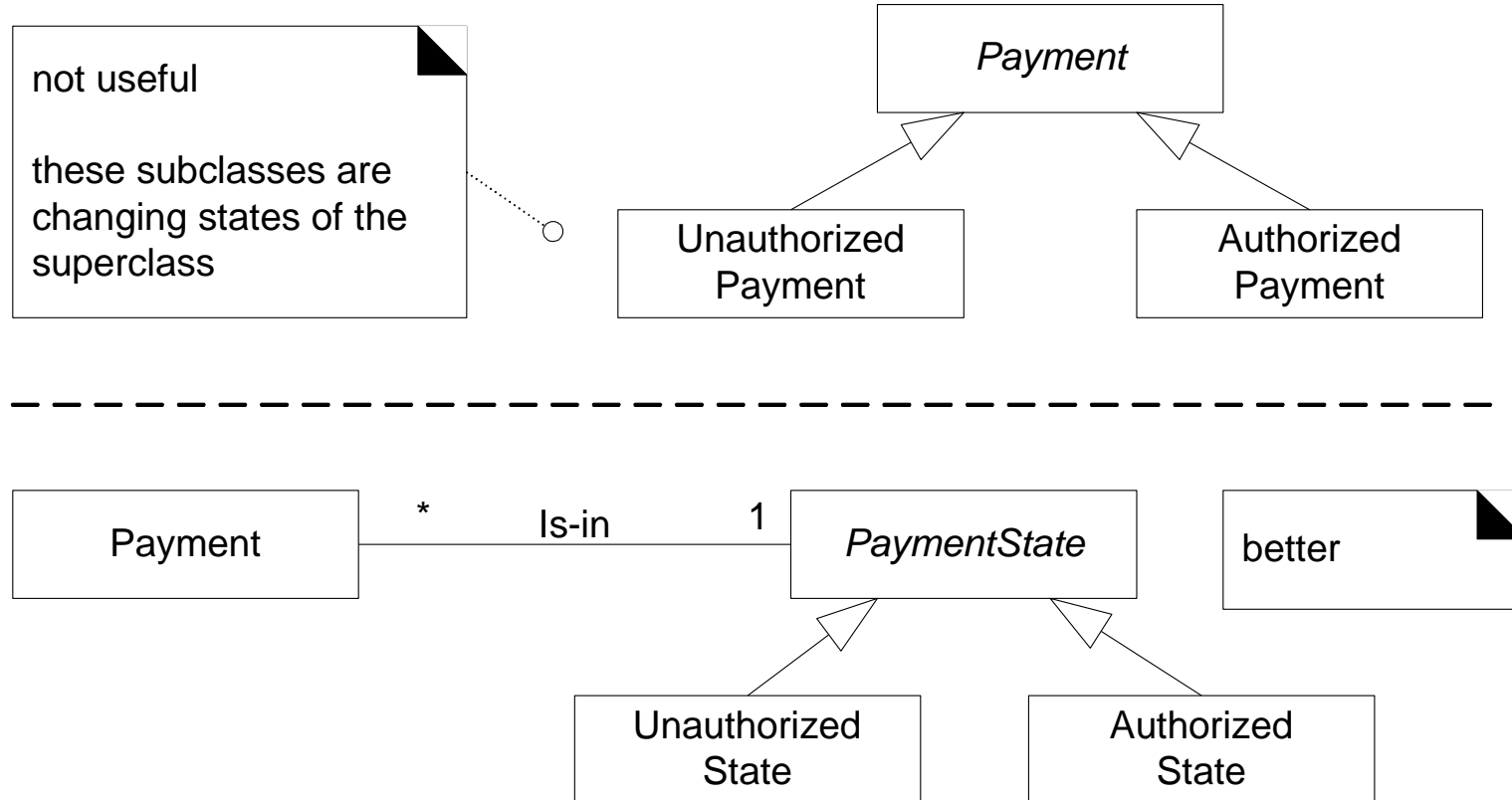
Generalisierung und Spezialisierung in der Fallstudie

- Assoziationen und Attribute dienen umgekehrt als Begründung für eine gemeinsame generalisierte Klasse.



Zustände im Domänenmodell

- Verschiedene konkrete und abstrakte Konzepte haben verschiedene **Zustände**, in denen sie sich befinden.
- Naheliegende Lösung
 - Zustände mittels Spezialisierung modellieren.
 - Das Problem: Wie können so Zustandsänderungen durchgeführt werden?
- Bessere Lösung: Eine **eigene Hierarchie** für die Zustände definieren.
 - Diese Lösung entspricht übrigens auch genau dem State-Pattern im SW-Design.



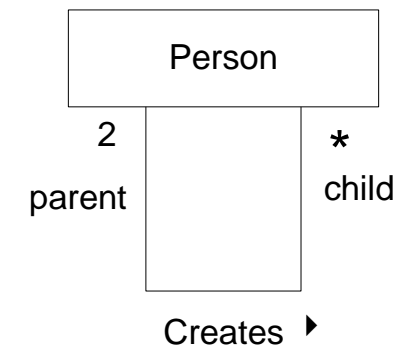
Rollen im DM

- Dasselbe Konzept (aber selten dieselbe Instanz) kann unterschiedliche Rollen einnehmen.
- Beispiel:
 - Je nach Stellenprofil hat ein Mitarbeiter andere Aufgaben, allenfalls noch Untergebene.
- Eine erste Möglichkeit zur Modellierung
 - Einsatz einer Assoziation, bei der dann das Ende mit einem Namen versehen wird (siehe nebenan).



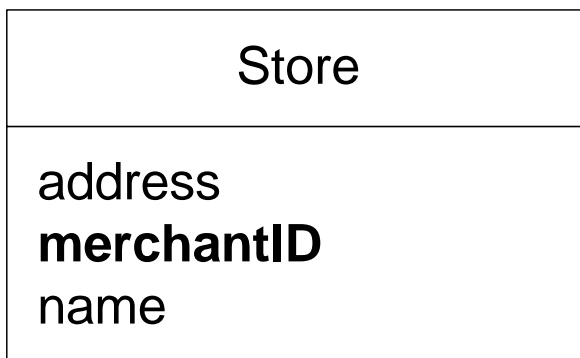
role name

describes the role of a city in the Flies-to association

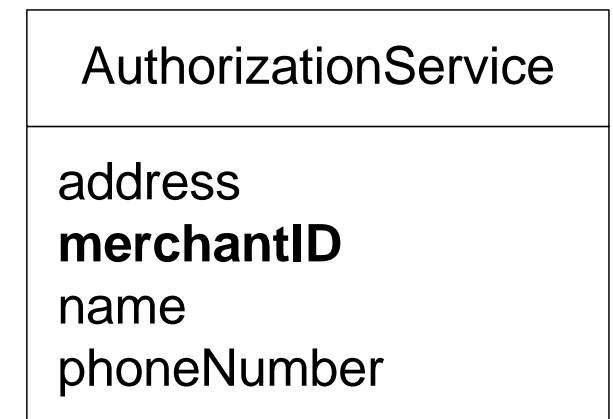


Assoziationsklassen

- Pro Kreditkartenherausgeber erhält das Geschäft eine eigene ID, und natürlich hat ein Kreditkartenherausgeber mehr als ein Geschäft als Kunde.
- Wo kommt nun diese merchantID hin?

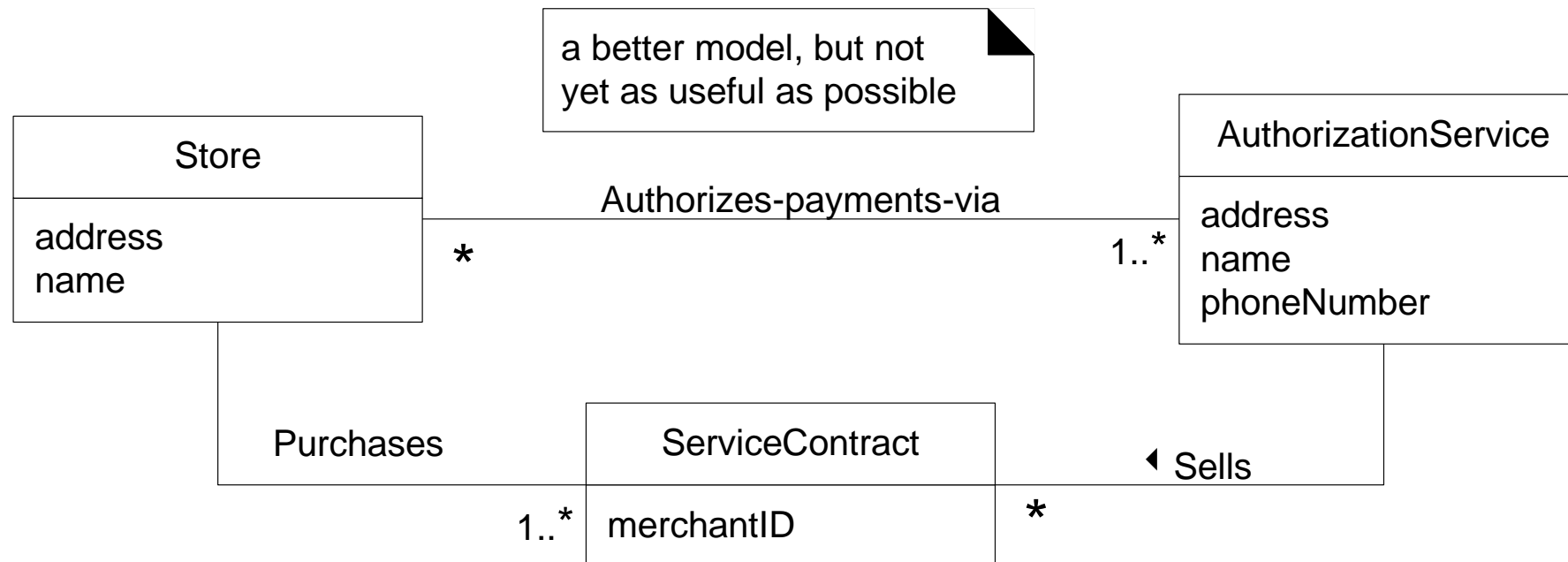


both placements of
merchantID are incorrect
because there may be more
than one merchantID

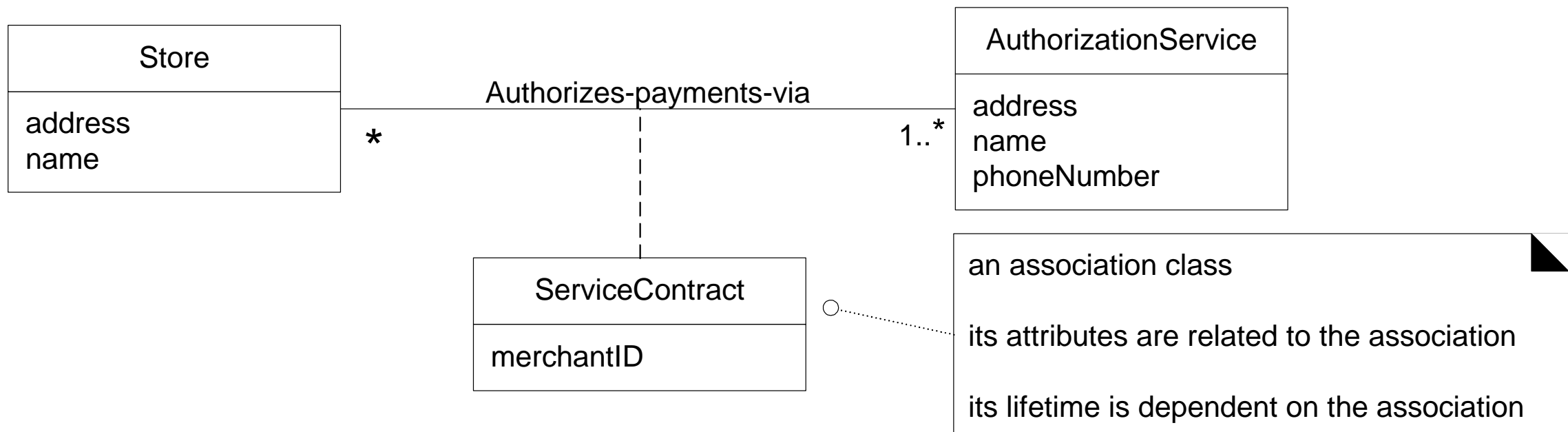


Assoziationsklassen

- Idee: Genauso, wie n:m Beziehungen mit einer weiteren Klasse zu 2x 1:n aufgebrochen werden, könnte auch hier so eine Klasse eingeführt werden.
- Aber eigentlich beschreibt ServiceContract ja die **Assoziation** zwischen Store und AuthorizationService genauer.



- Für dieses Problem kennt UML eine Lösung: **Assoziationsklassen**



Masseinheiten

- Gerade numerische Angaben sind oft mit einer **Masseinheit** verbunden.
 - Preis, Gewicht, Volumen, Geschwindigkeit
 - Ohne Masseinheit kann die angegebene Zahl nicht korrekt interpretiert werden
- Häufig macht es Sinn, diese Masseinheit im DM **explizit** als Konzept zu modellieren.
 - Money, Weight, Volume
- Eine entsprechende SW-Klasse kann später in der Umsetzung noch weitere hilfreiche Methoden aufnehmen
 - z.B. die Umrechnung von metrischen Werten in imperiale Einheiten.

Agenda

1. Einleitung und Motivation
2. Grundlagen
3. Vorgehen
4. Analysemuster
5. Wrap-up und Ausblick

Wrap-up

- Das **Domänenmodell** visualisiert den Fachbereich in Form eines vereinfachten UML Klassendiagramms.
- Das **Domänenmodell** hilft uns, den Fachbereich zu **verstehen** und dient als **Inspiration** für fachliche **SW-Klassen**.
- Entwickeln Sie das **Domänenmodell** nach denselben Prinzipien, die ein **Kartograf** einsetzt.
- Identifizieren Sie Konzepte, fügen Sie ihnen Attribute hinzu und setzen Sie die Konzepte zueinander in Beziehung.
- Wenden Sie bewährte **Analysemuster** an wie Beschreibungsklassen, Komposition, Generalisierung/Spezialisierung, Zustandsmodellierung und Einheiten als eigene Konzepte.

Ausblick

- In der nächsten Lerneinheit werden wir:
 - Den Begriff Software Architektur kennenlernen
 - Verschiedene Softwarearchitekturen genauer anschauen

Quellenverzeichnis

- [1] Larman, C.: UML 2 und Patterns angewendet, mitp Professional, 2005
- [2] Seidel, M. et al.: UML @ Classroom: Eine Einführung in die objektorientierte Modellierung, dpunkt.verlag, 2012