Zürcher Hochschule
für Angewandte Wissenschaften
School of Engineering
InIT Institut für angewandte Informationstechnologie

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.

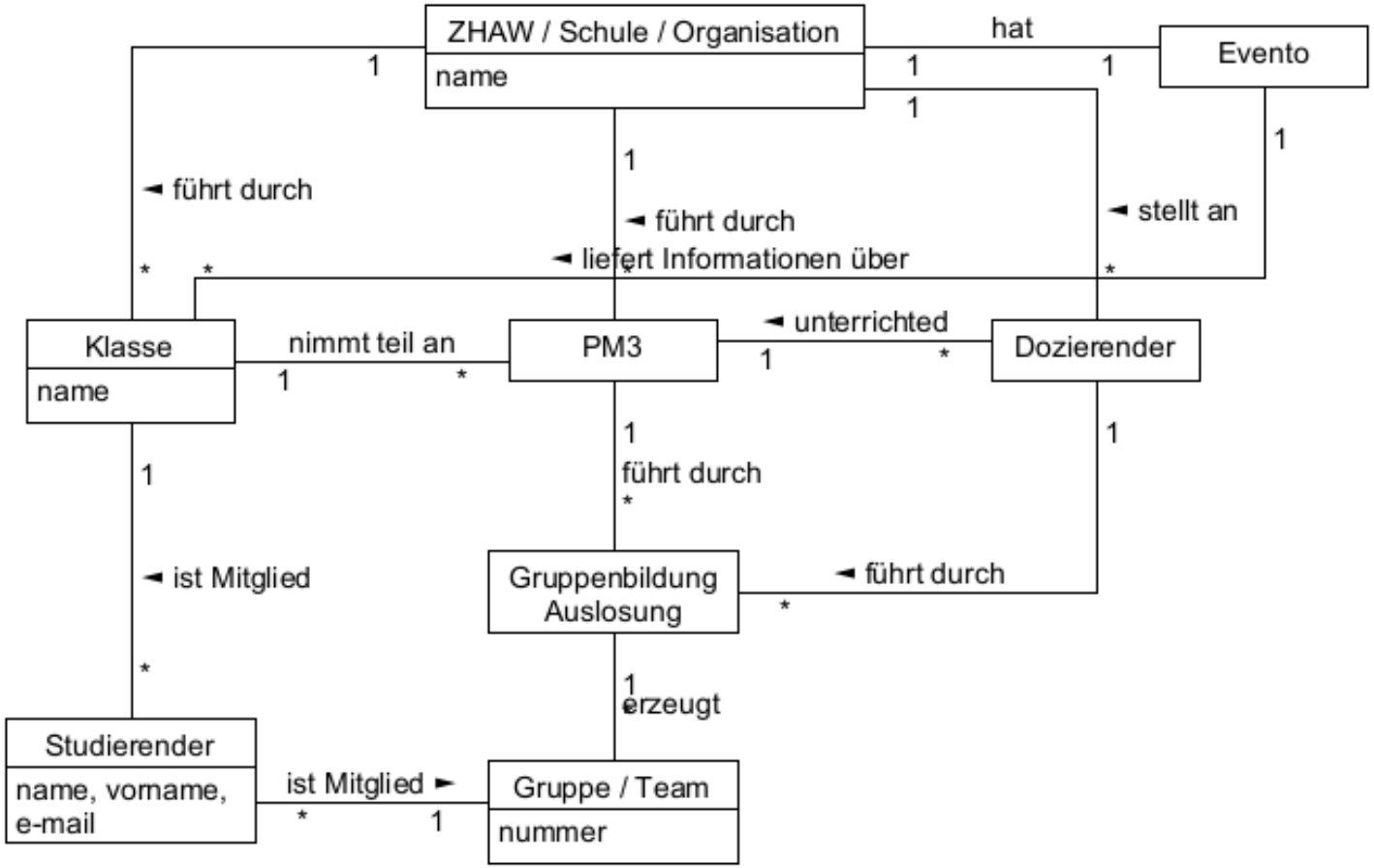
Agenda

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

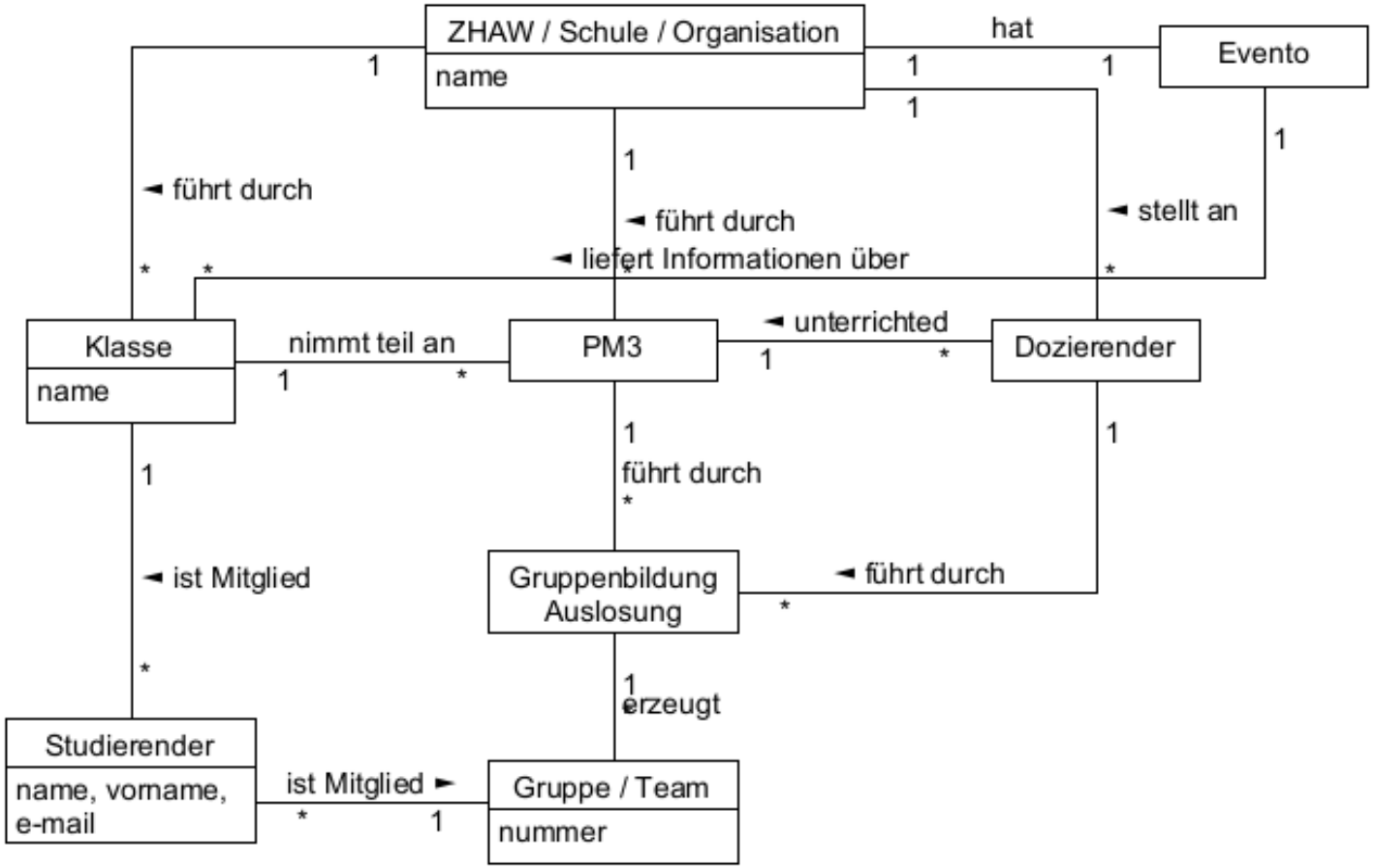
Einleitung und Motivation

- Ein Domänenmodell ist ein vereinfachtes UML Klassendiagramm.
- Es zeigt fachliche Begriffe mit ihren Attributen und setzt diese Begriffe zueinander in Beziehung.
- Es geht noch nicht um Software, sondern nur um die Problemstellung und das Fachgebiet.
- Gehen Sie dabei vor wie ein Kartograf, der bestehende Begriffe verwendet und nichts erfindet, was es nicht schon gibt.
- Verwenden Sie dabei bewährte Analyse-Muster.
- Das Domänenmodell hilft, die Anforderungen besser zu verstehen und ist im Design Inspirationsquelle für fachliche SW-Klassen.

Zur Erinnerung: Auslosungstool



Zur Erinnerung: Auslosungstool



Fallbeispiel: Anforderungen einer elektronischen Kasse

- Eine elektronische Kasse wird von einem Kassier bedient.
- Ein Kunde kommt mit seinen Artikeln zur Kasse und der Kassier hat nun die Aufgabe, diese Artikel zu erfassen.
- Nachdem alle Artikel erfasst wurden, muss das Total berechnet und die Bezahlung abgewickelt werden.
- Die Kasse steht in einem Geschäft.

Denkpause

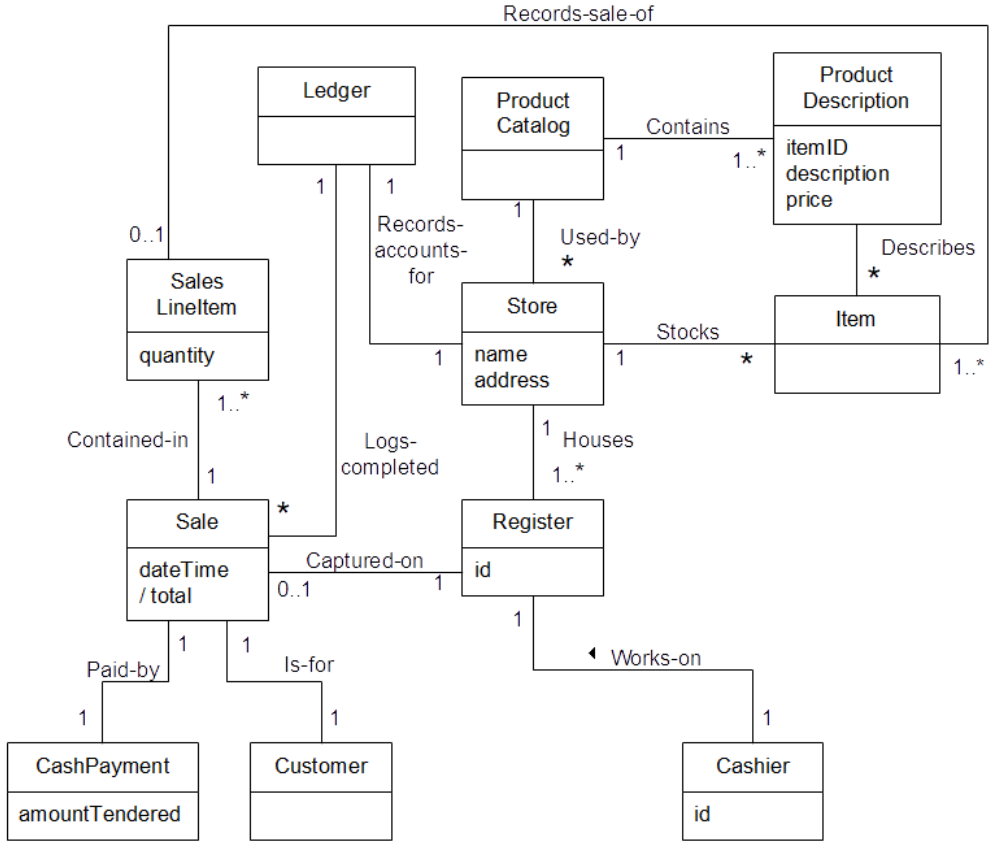
Aufgabe 4.1 (5’)

Diskutieren Sie in Murmelgruppen folgende Fragen:

- Welche Begriffe des Fachgebiets «elektronische Kasse» können Sie aus den Anforderungen identifizieren?
- Welche Eigenschaften haben diese Begriffe?
- Wie stehen diese Begriffe zueinander in Beziehung?

Denkpause

Aufgabe 1 – Musterlösung

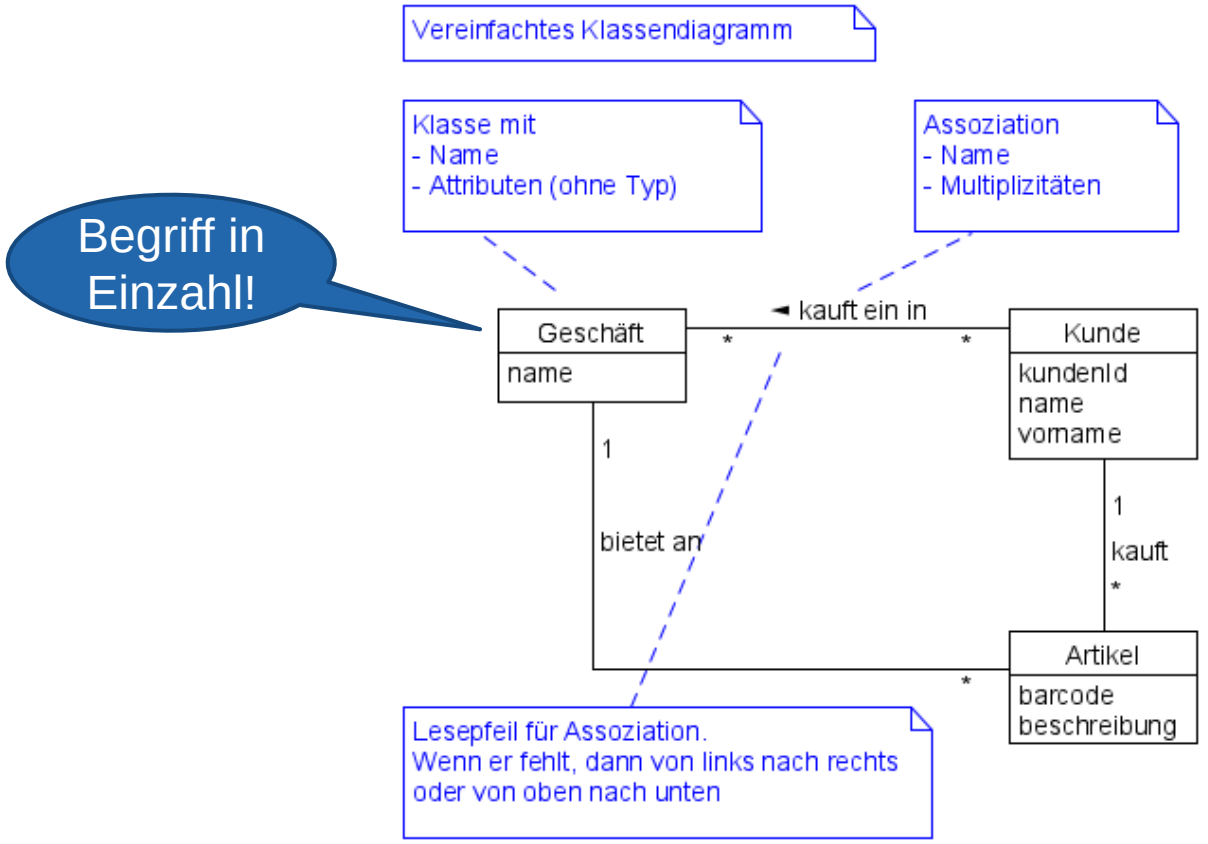


Agenda

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

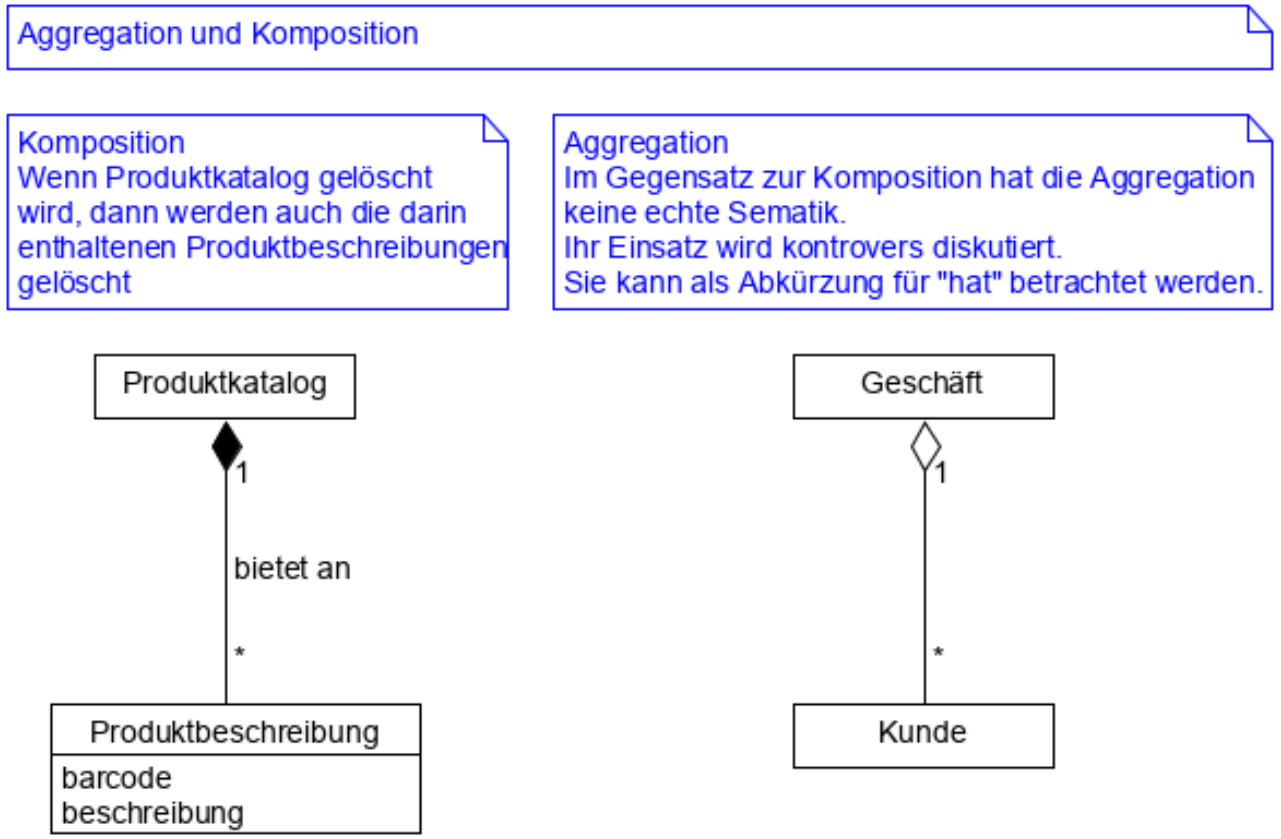
Domänenmodell als vereinfachtes UML Klassendiagramm

- 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.



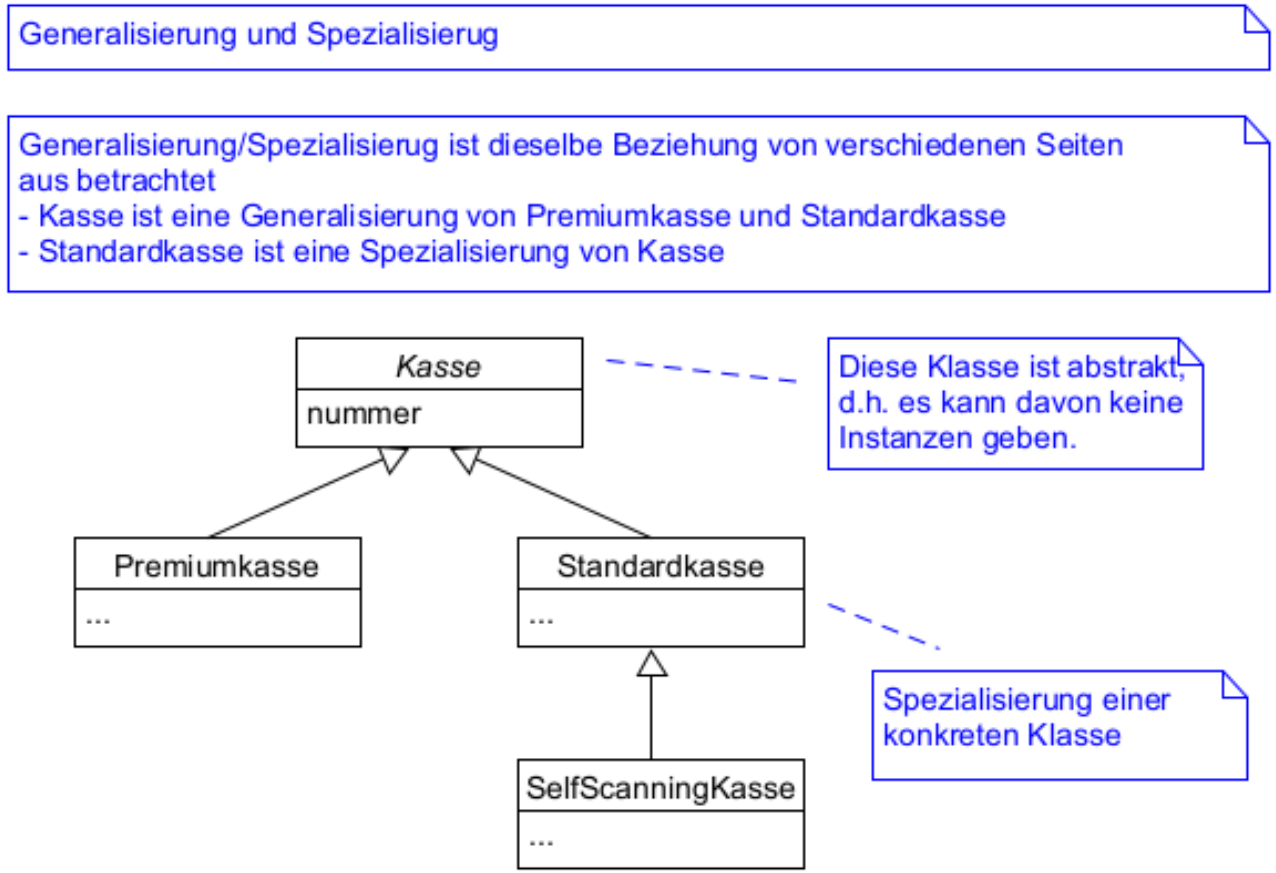
Vereinfachtes UML Klassendiagramm

- Mit Aggregation / Komposition kann die Assoziation noch genauer definiert werden.
- Es wird empfohlen, diese Assoziationsart nur dann im Domänenmodell zu verwenden, wenn es einen guten Grund dafür gibt. Im Zweifelsfall reicht eine einfache Assoziation aus.



Vereinfachtes UML Klassendiagramm

- Mit Generalisierung / Spezialisierung kann die Assoziation noch genauer definiert werden.



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

Substantive aus Anwendungsfällen herausziehen

- Schauen Sie die Anforderungen, insbesondere die Anwendungsfälle, an und überprüfen Sie jedes Substantiv, ob es ein relevantes Konzept des Fachgebiets beschreibt.
- Beachten Sie dabei die Mehrdeutigkeit der natürlichen Sprache.
- Beispiel «Handle Sale» Use Case :
  - Customer arrives at POS checkout with goods and/or services to purchase.
  - Cashier starts a new sale.
  - Cashier enters item identifier
- Nicht alle Substantive sind Konzepte, manche sind auch Attribute oder gehören nicht zum Fachgebiet.

Konzepte mit Kategorienlisten finden

- Versuchen Sie, anhand folgender Kategorienliste für Business-Applikationen, weitere Konzepte in Ihrem Fachgebiet zu finden.

Kategorie	Mögliche Konzepte für DM
Geschäftstransaktionen	
• Transaktionen als Ganzes	
• Transaktionsposition	
• Produkt, das damit verbunden ist	
• Wo wird Transaktion registriert?	
• Rollen von beteiligten Personen	
• Ort der Transaktion	
• Beschreibung von Dingen	
• Ereignisse mit Ort/Zeit	

Konzepte mit Kategorienlisten finden

- Versuchen Sie, anhand folgender Kategorienliste für Business-Applikationen, weitere Konzepte in Ihrem Fachgebiet zu finden.

Kategorie	Mögliche Konzepte für DM
Geschäftstransaktionen	
• Transaktionen als Ganzes	Sale
• Transaktionsposition	SalesLineItem
• Produkt, das damit verbunden ist	Item
• Wo wird Transaktion registriert?	Register
• Rollen von beteiligten Personen	Cashier
• Ort der Transaktion	Store
• Beschreibung von Dingen	ProductDescription
• Ereignisse mit Ort/Zeit	Sale

Konzepte mit Kategorienlisten finden

- Versuchen Sie, anhand folgender Kategorienliste für Business-Applikationen, weitere Konzepte in Ihrem Fachgebiet zu finden.

Kategorie	Mögliche Konzepte für DM
Physische Objekte	
Kataloge	
Container von Dingen	
Dinge in den Containern	
Andere beteiligte Systeme	
Rollen von beteiligten Personen	
Artefakte (Pläne, Finanzen, Arbeit, Verträge, ...)	
Zahlungsinstrumente	

Konzepte mit Kategorienlisten finden

- Versuchen Sie, anhand folgender Kategorienliste für Business-Applikationen, weitere Konzepte in Ihrem Fachgebiet zu finden.

Kategorie	Mögliche Konzepte für DM
Physische Objekte	Register
Kataloge	ProductCatalog
Container von Dingen	Store
Dinge in den Containern	Item
Andere beteiligte Systeme	CreditAuthorizationSystem
Rollen von beteiligten Personen	Cashier
Artefakte (Pläne, Finanzen, Arbeit, Verträge, ...)	Receipt
Zahlungsinstrumente	Cash, Credit Card

Assoziationen mit Kategorienlisten identifizieren

- Fragen Sie sich, ob es für die folgenden Kategorien Beziehungen zwischen Konzepten gibt.

Kategorie	Mögliche Assoziation für DM
Transaktion	
• Position	
• Produkt	
• Rolle	
Teil zum Ganzen	
Beschreibung zum Gegenstand	
Protokoll zum Gegenstand	
Verwendung	

Assoziationen mit Kategorienlisten identifizieren

- Fragen Sie sich, ob es für die folgenden Kategorien Beziehungen zwischen Konzepten gibt.

Kategorie	Mögliche Assoziation für DM
Transaktion	Payment - Sale
• Position	SalesLineItem - Sale
• Produkt	Item - SalesLineItem
• Rolle	Customer - Payment
Teil zum Ganzen	Register - Store
Beschreibung zum Gegenstand	ProductdDescription - Item
Protokoll zum Gegenstand	Sale - Register
Verwendung	Cashier - Register

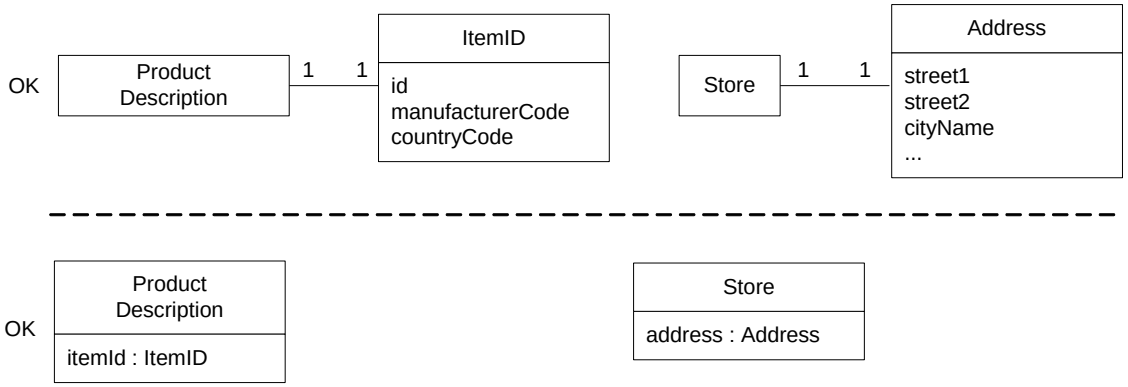
Datentypen von Attributen

- 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 `==`

Datentypen von Attributen

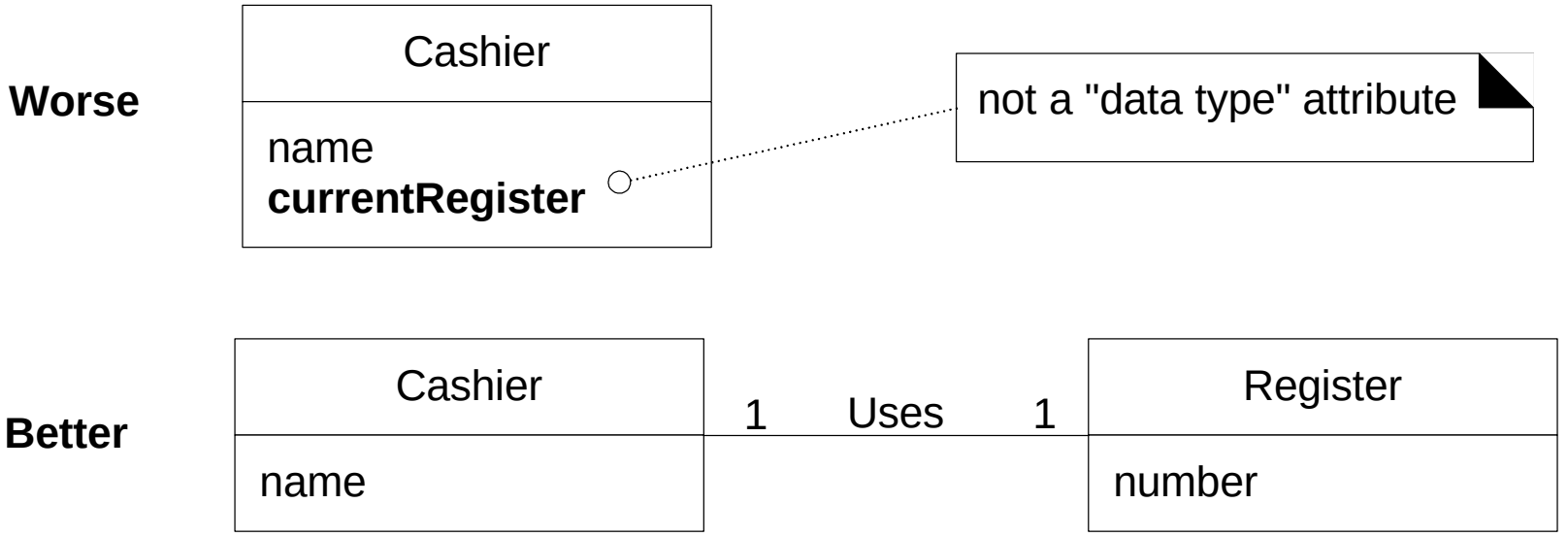
- Wenn nötig, werden im DM eigene Datentypen als Konzepte eingeführt.
- Eigene Datentypklassen dann definieren, wenn:
  - der Typ aus mehreren Abschnitten wie zum Beispiel die Telefonnummer besteht.
  - Operationen darauf möglich sind wie die Validierung einer Kreditkartennummer.
  - der Typ selber noch eigene Attribute hat wie zum Beispiel ein Verkaufspreis, der ein Anfangs- und Enddatum hat.
  - der Typ verknüpft ist mit einer Einheit, zum Beispiel ein Preis ist mit einer Währung verknüpft.

- Modellierung in UML
  - Verknüpfung über Assoziation
  - Direkt als Attributtyp angeben



Anti-Pattern: Attribute an Stelle von Assoziationen

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

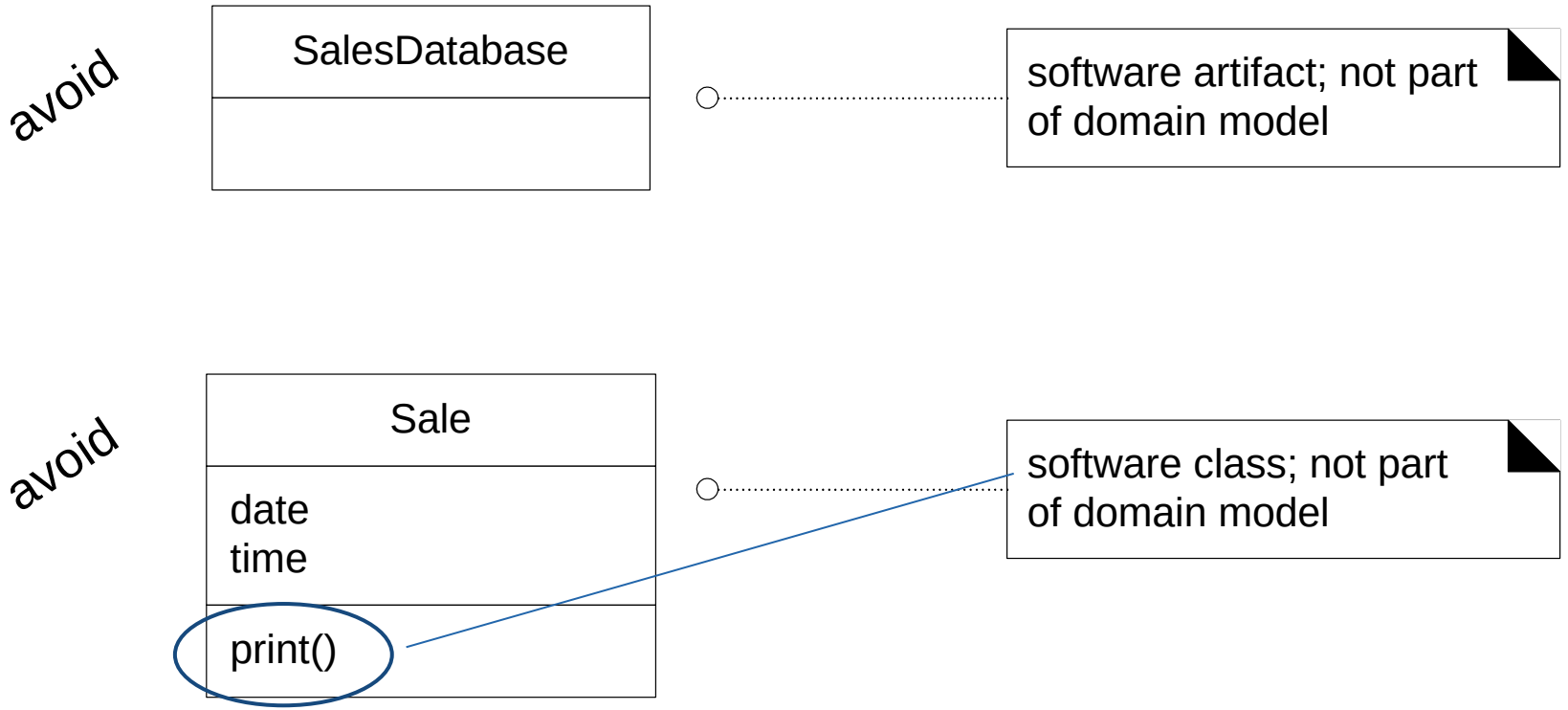


Vorgehensweise eines Kartografen

- Vorhandene Begriffe oder Wissen werden eingesetzt
  - Der Kartograf besucht das Gebiet, das er zeichnet
  - Er frägt die dortigen Bewohner (die „Experten“), wie die Ortschaften und Gewässer heissen
- Unwichtiges weglassen
  - Einzelne Bäume werden weggelassen, ausser sie sind von weitem sichtbar.
- Nichts hinzufügen, was es (noch) nicht gibt
  - Eigentlich selbstverständlich, oder?
  - Als Ausnahme darf das System, das entwickelt wird, aber so noch nicht existiert, auch eingetragen werden
  - Sicher keine Elemente der Software Lösung
- Nur analysieren, (noch) keine Lösungen entwerfen!

Anti-Pattern : Software-Klassen

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



Denkpause

Aufgabe 4.2 (5’)

Diskutieren Sie in Murmelgruppen folgende Fragen:

Ist «Database» verboten als Konzept eines Domänenmodells?

Machen wir den Gegentest. Können Sie sich ein Fachgebiet mit dem Konzept «Database» vorstellen?

Denkpause

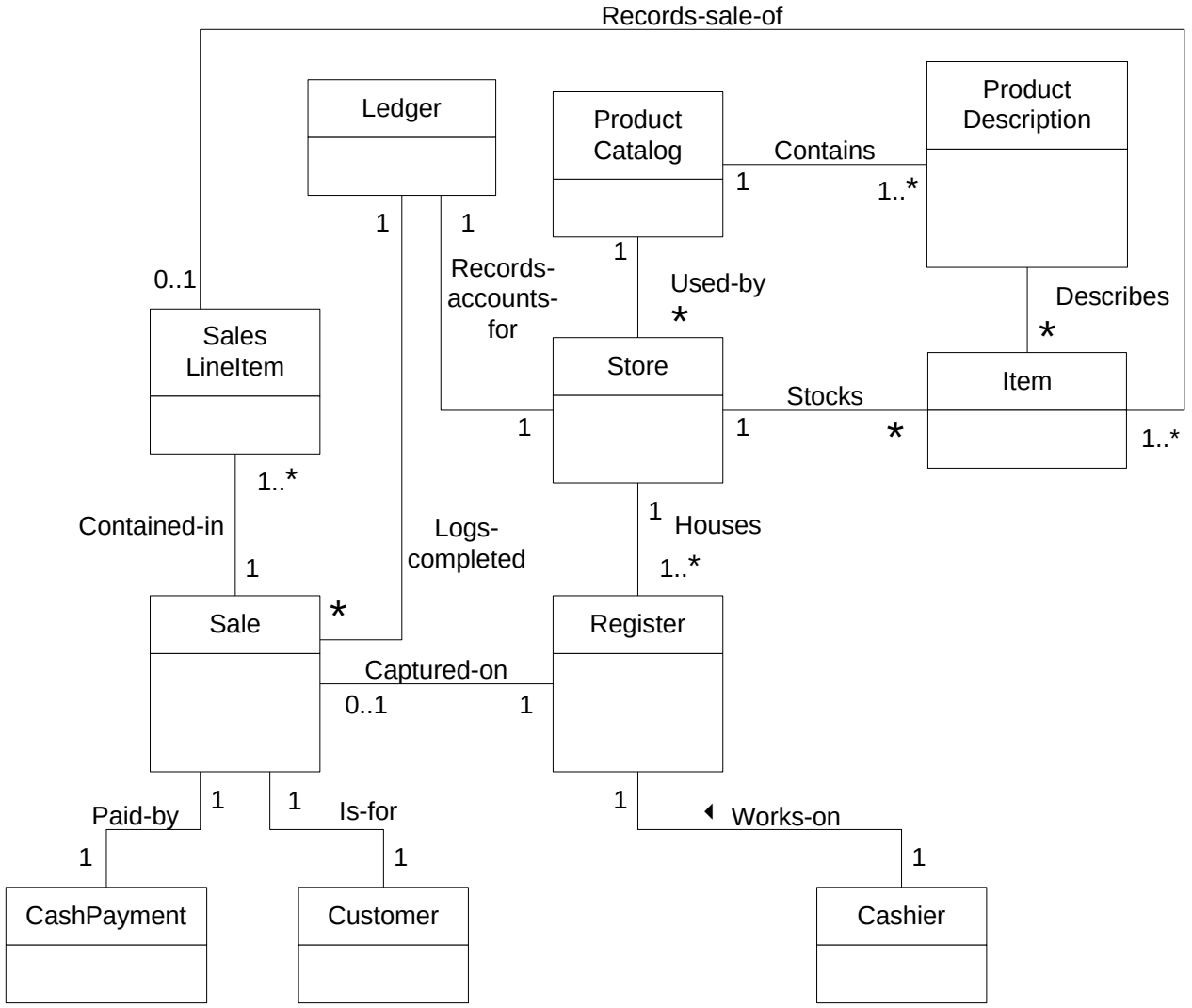
Aufgabe 4.2 – Musterlösung

Natürlich gibt es technische Anwendungen, wo eine Datenbank zur Fachdomäne gehört. Am offensichtlichsten ist es, eine Datenbank selber zu programmieren, aber auch z.B. Analyse-Werkzeuge von Datenbanken haben die Datenbank als Konzept im Domänenmodell.

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



Denkpause

Aufgabe 4.3 (5’)

Diskutieren Sie in Murmelgruppen folgende Frage:

Versuchen Sie Orte im aktuellen Domänenmodell zu finden, die vermutlich noch genauer modelliert werden müssen.

Denkpause

Aufgabe 4.3 – Musterlösung

Das Domänenmodell zeigt ja den Stand für die 1. Iteration. In den Iterationen 2 und 3 kommen noch folgende Aspekte dazu

- Verschiedene Bezahlarten wie Kreditkartebezahlung, aber auch Kreditkarte und Kreditkartenorganisation selber, analog dazu Check
- Zeitabhängige Preise, Rabatte für Kundengruppen und Produktgruppen, Behandlung von Fremdwährungen
- Buchhaltungssystem
- Rollen von Mitarbeitern

Agenda

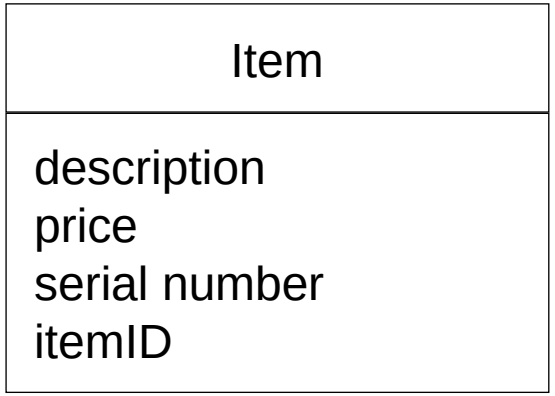
1. Einleitung und Motivation
2. Grundlagen
3. Vorgehen
4. Analysemuster
5. Wrap-up und Ausblick

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 typsicherweise 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.



Denkpause

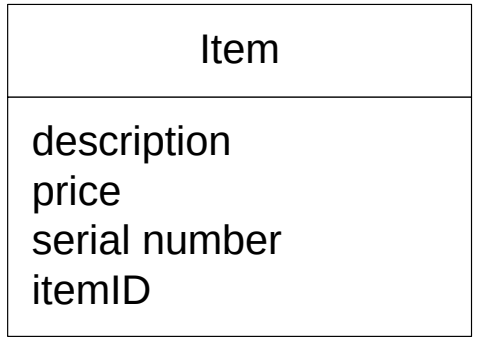
Aufgabe 4.4 (5’)

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

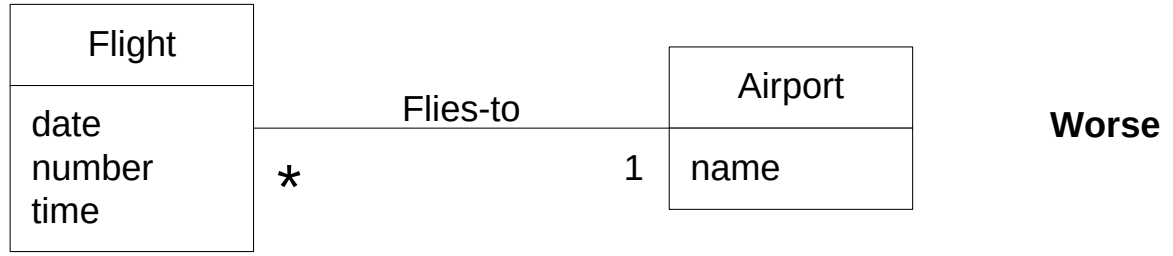
- Attribute, die für alle Artikel eines Typs gleich sind, werden in eine eigene Klasse herausgezogen.



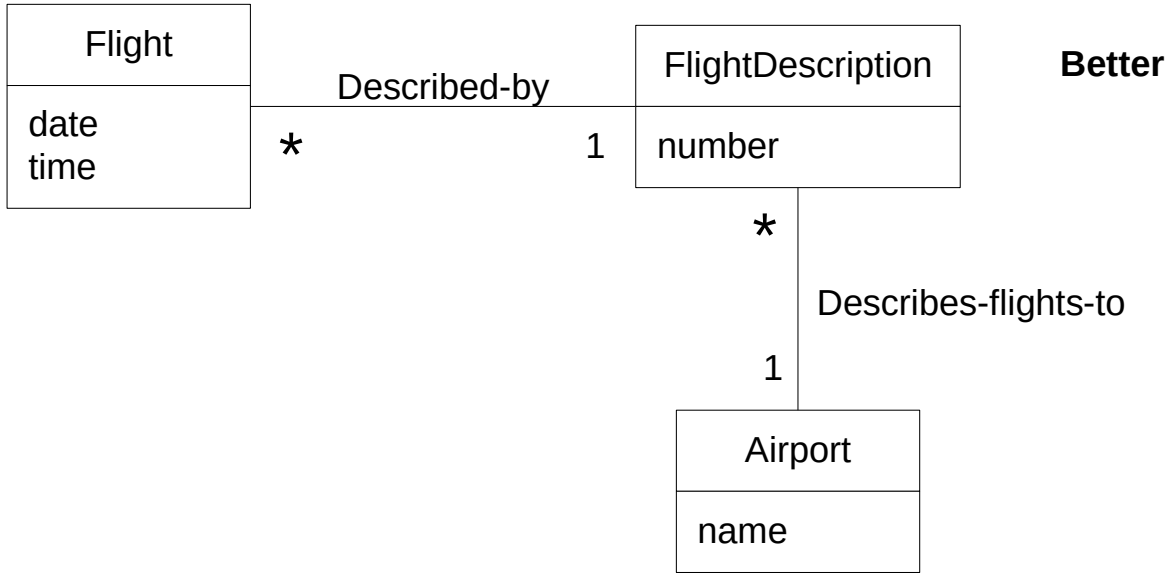
Worse

Better

Beschreibungsklasse für Flug



Worse



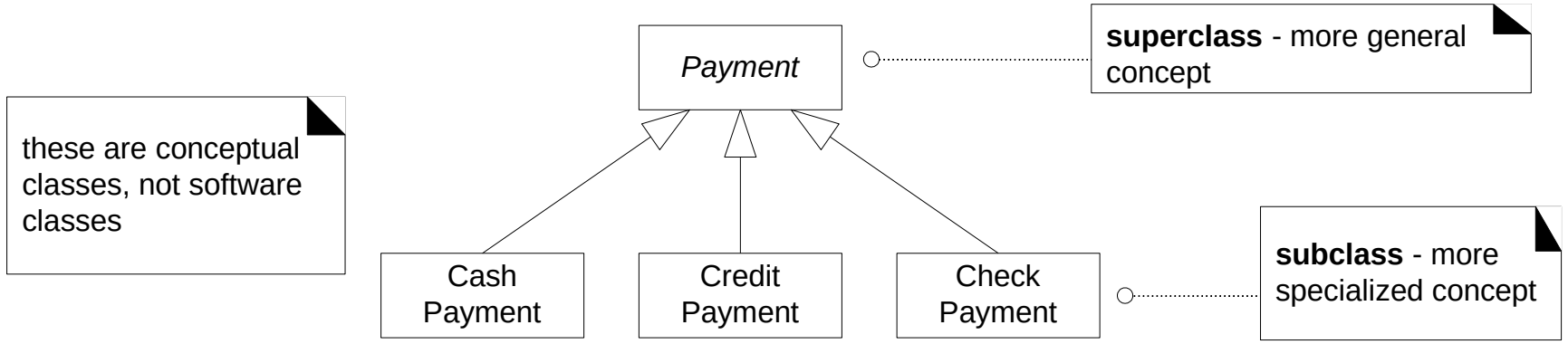
Better

Generalisierung und Spezialisierung

- Es gibt relativ häufig Konzepte, die als Spezialisierung eines allgemeineren Konzepts betrachtet werden können.
- Dieselbe Beziehung wird in umgekehrter Richtung als Generalisierung bezeichnet.
- 2 Regeln, die dabei beachtet werden müssen:
  - 100% Regel : Alle Instanzen eines spezialisierten Konzepts sind auch Instanzen des generalisierten Konzepts
  - «Ist ein» Regel : Spezialisiertes Konzept «is a» / ist ein generalisiertes Konzept
- Mit Augenmass einsetzen.
  - Immer überprüfen, ob es für die Anwendung schlussendlich relevant ist.
  - Werden spezialisierte Konzepte anders behandelt oder haben sie weitere, eigene Attribute und Assoziationen?

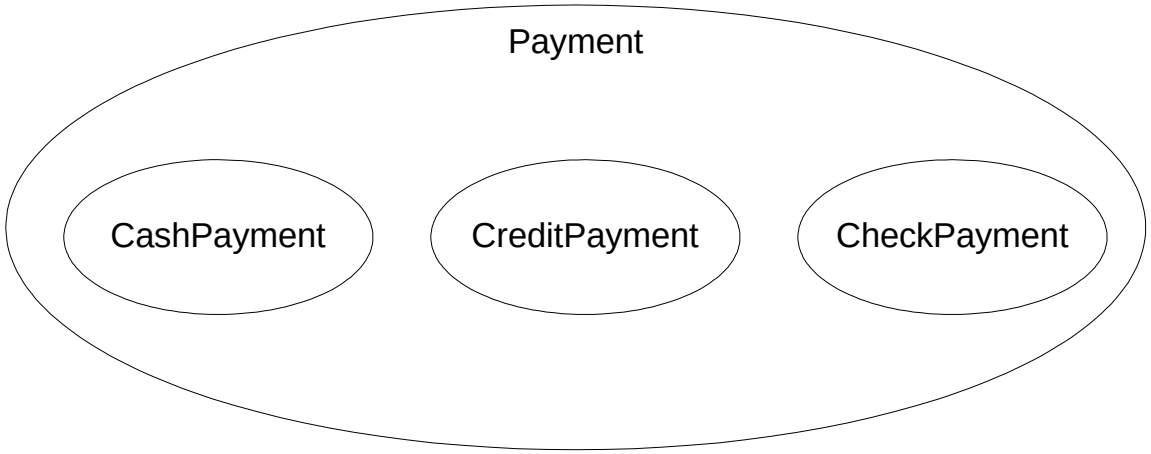
Generalisierung und Spezialisierung in der Fallstudie

- Es gibt verschiedene Zahlungsmöglichkeiten: Bar, Kreditkarte, Check



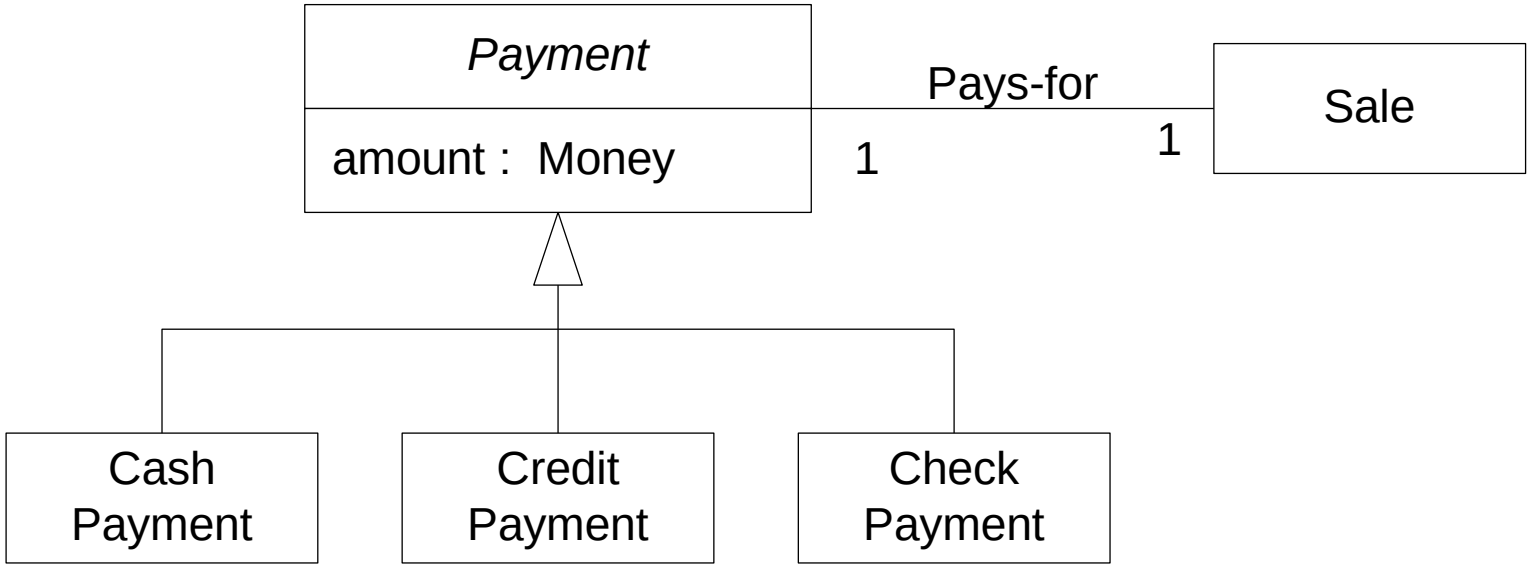
Generalisierung und Spezialisierung in der Fallstudie

- Diese Spezialisierungen erfüllen die 100% und «is a» Regel.



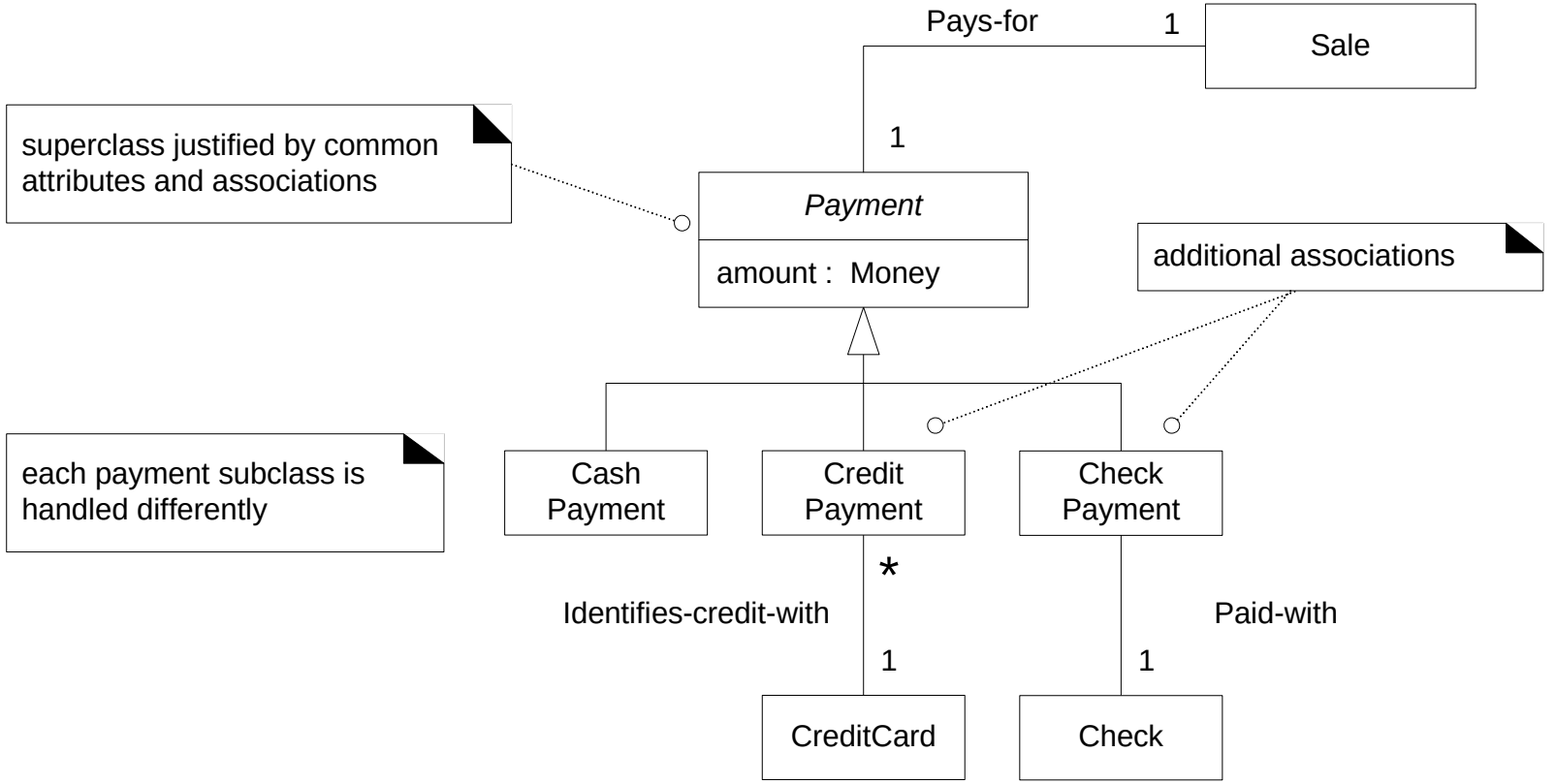
Generalisierung und Spezialisierung in der Fallstudie

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



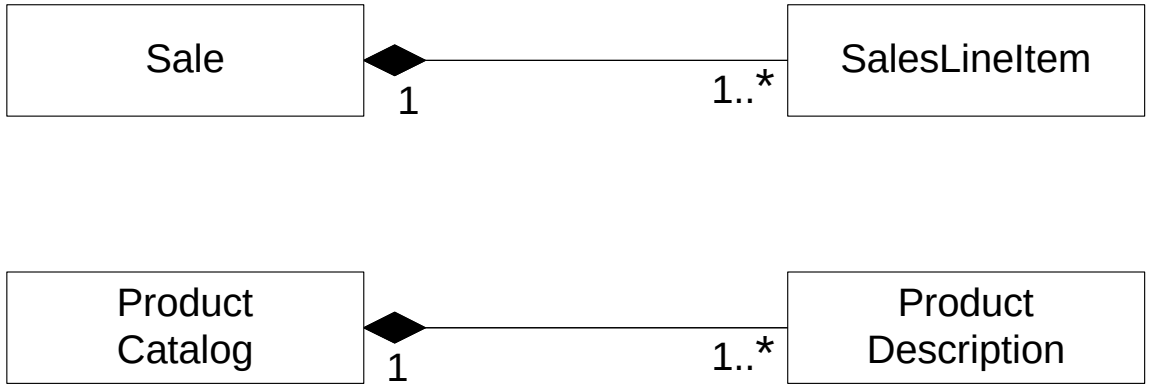
Generalisierung und Spezialisierung in der Fallstudie

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



Komposition

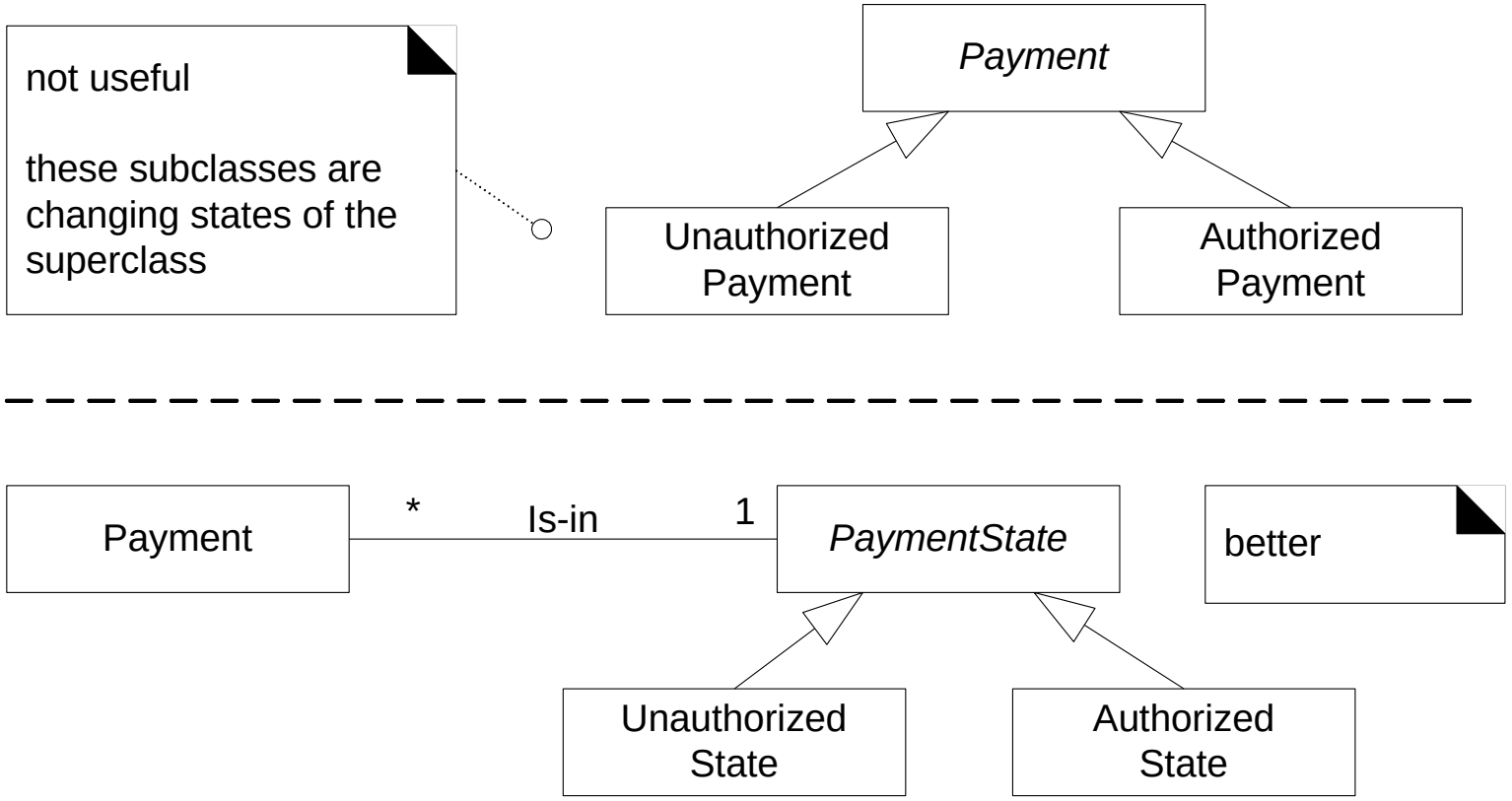
- Kompositionen erweitern den Informationsgehalt des Modells.
- Die Semantik der Komposition wurde am Anfang bei der Einführung des vereinfachten UML-Klassendiagramms erwähnt (vollst. Foliensatz).



Zustände im Domänenmodell

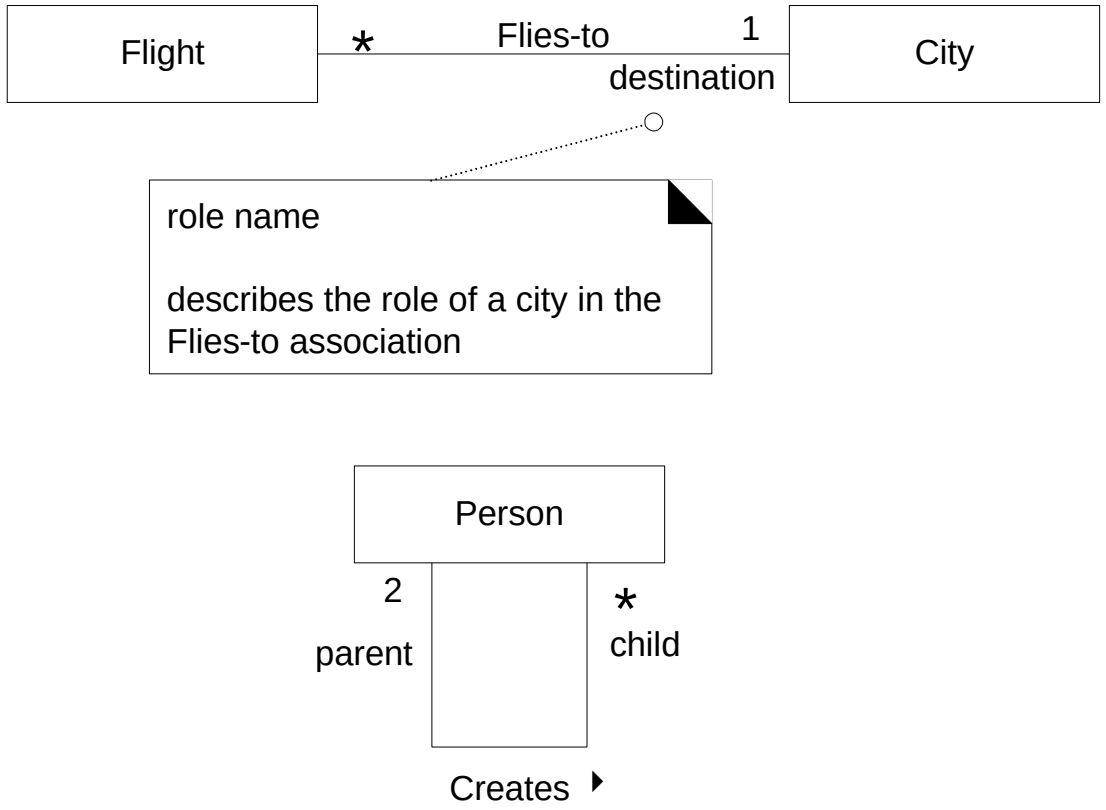
- Verschiedene konkrete und abstrakte Konzepte haben verschiedene Zustände, in denen sie sich befinden.
- Naheliegende Lösung
  - Zustände mittels Spezialisierung modellieren.
  - Das Problem: Wie können so Zustandsänderungen durchgeführt werden?
- Bessere Lösung: Eine eigene Hierarchie für die Zustände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

Zustände


not useful
these subclasses are changing states of the superclass
Payment
Unauthorized Payment
Authorized Payment
Payment
*
Is-in
1
PaymentState
better
Unauthorized State
Authorized State

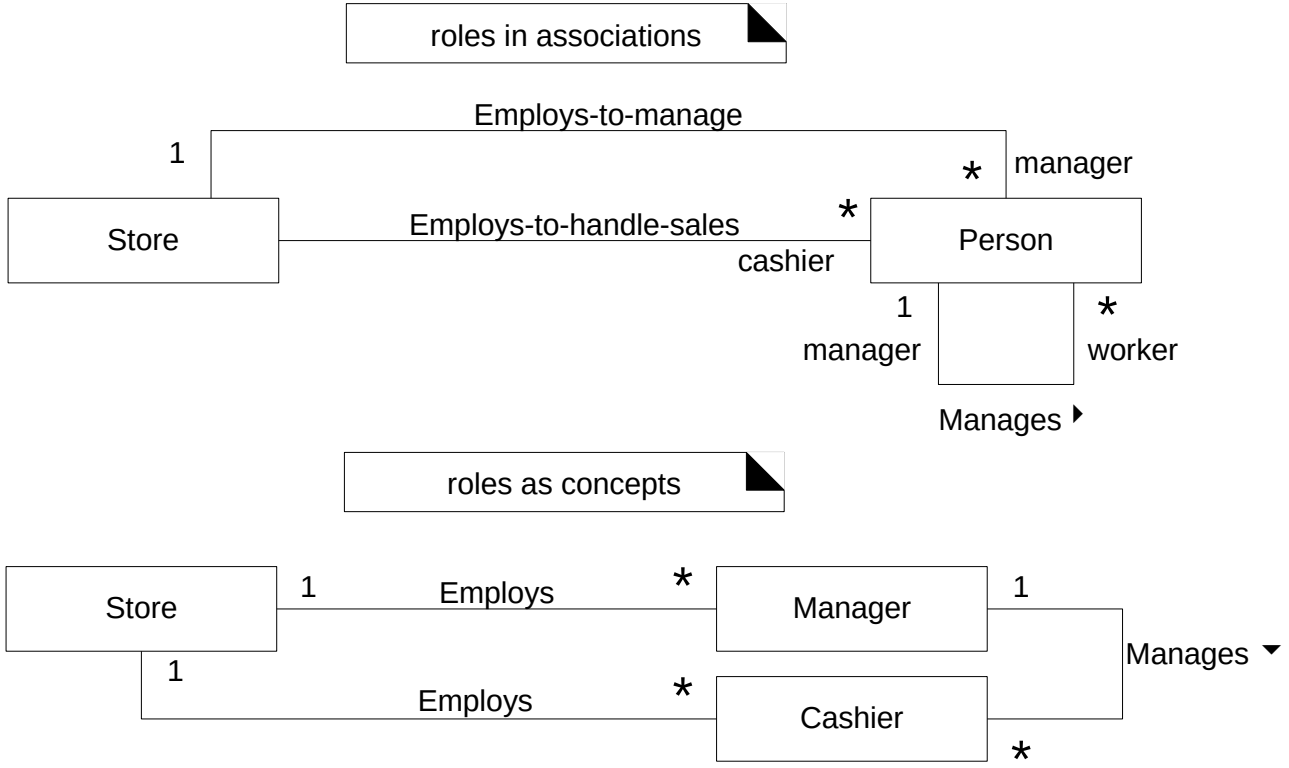
Rollen im DM

- 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).



Rollen

- Zweite Möglichkeit: Rollen als Konzepte zu modellieren.
  - Dann hat man die Möglichkeit, den Rollen noch Attribute zu geben.



Assoziationsklassen

- Assoziationen, die Beziehung zwischen Konzepten anzeigen, können noch eigene Attribute haben.
- Als Beispiel dient die Beziehung zwischen einem Geschäft und dem Kreditkartenherausgeber.
- 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?

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?

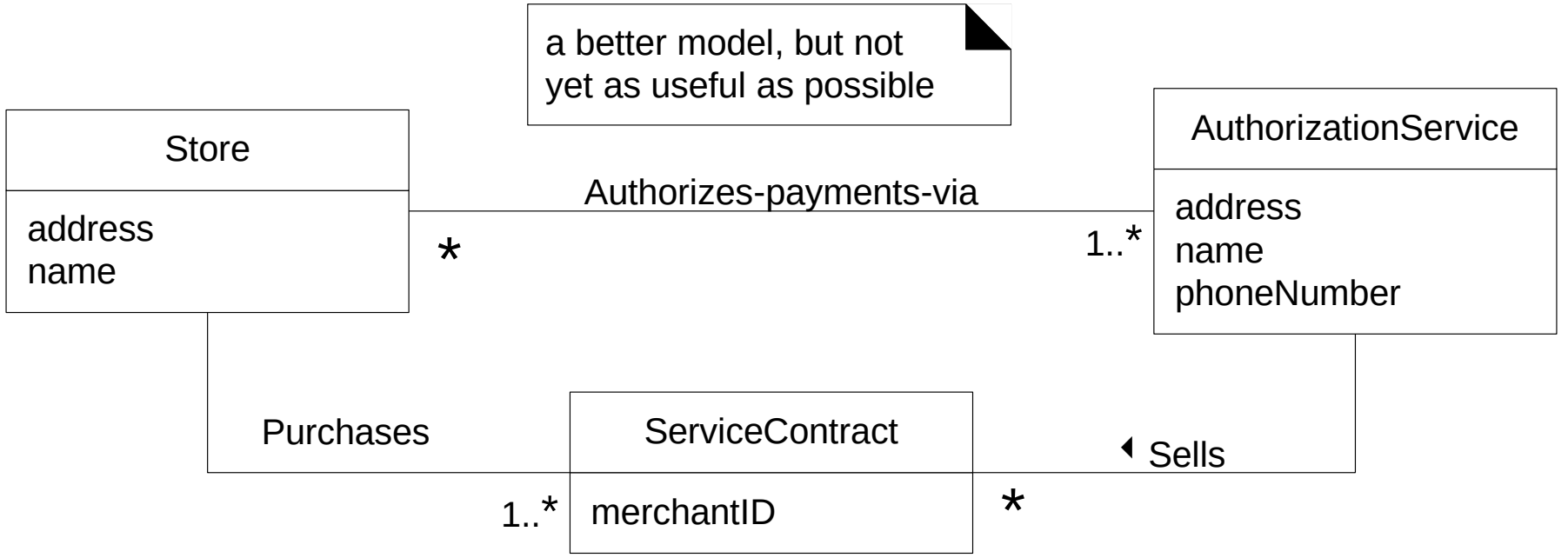
Store
address **merchantID** name

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

AuthorizationService
address **merchantID** name phoneNumber

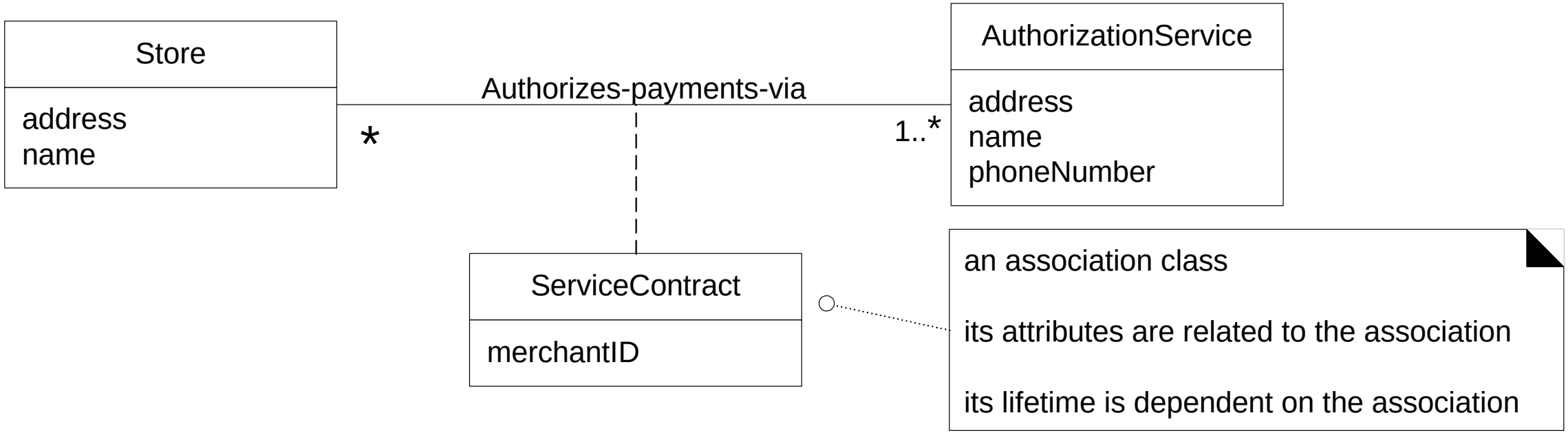
Assoziationsklassen

- 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.



Assoziationsklassen

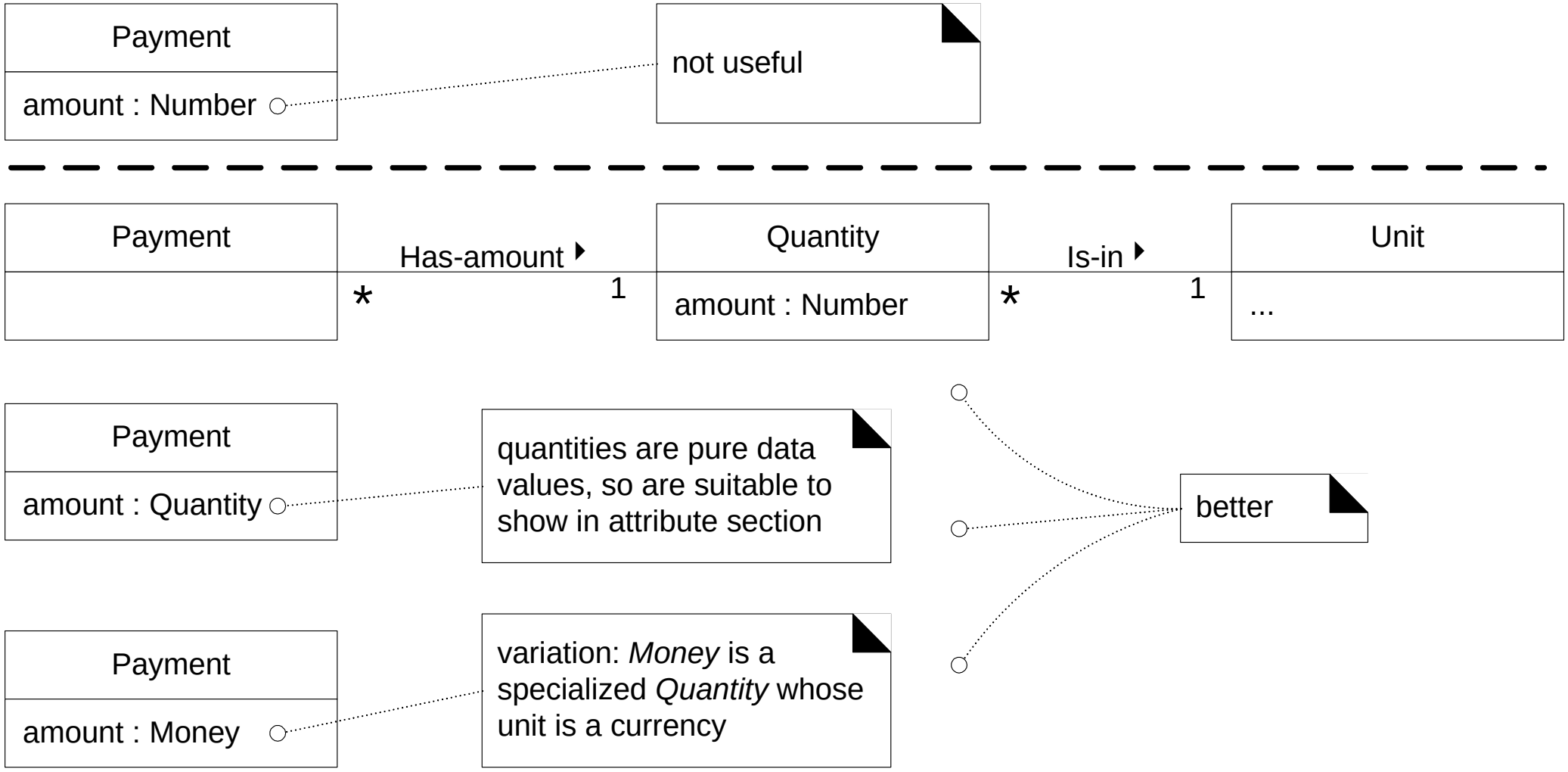
- Für dieses Problem kennt UML eine Lösung: Assoziationsklassen



Masseinheiten

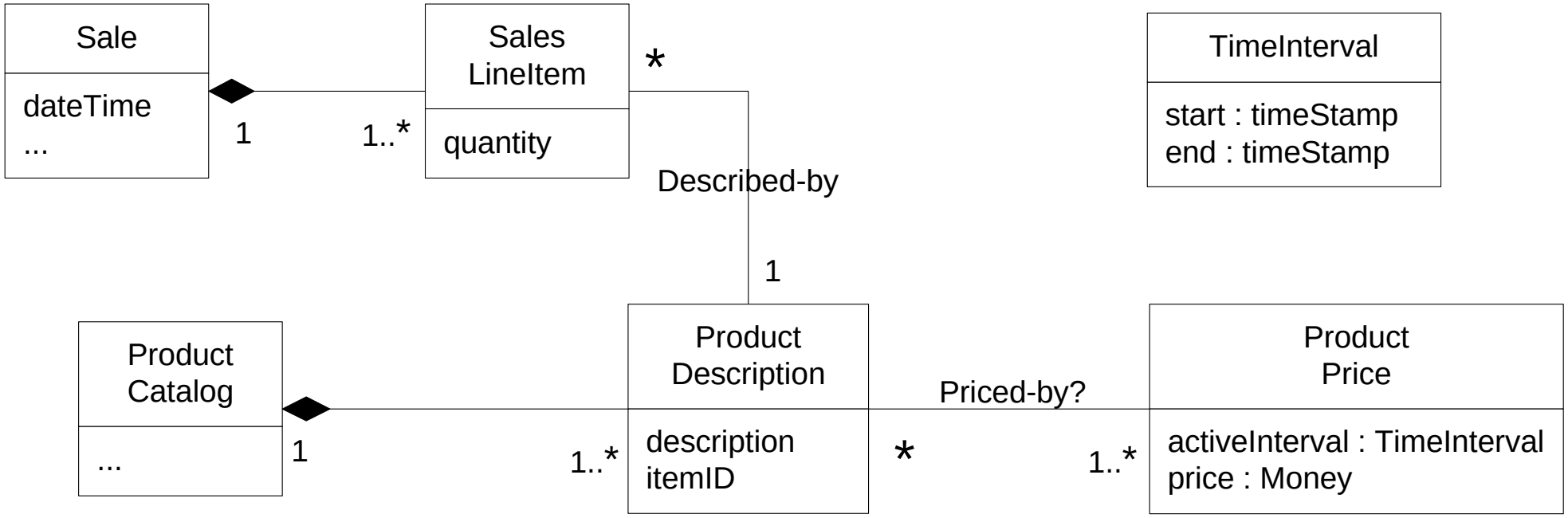
- Gerade numerische Angaben sind oft mit einer Masseinheit verbunden.
  - Preis, Gewicht, Volumen, Geschwindigkeit
  - Ohne Masseinheit kann die angegebene Zahl nicht korrekt interpretiert werden
- Häufig macht es Sinn, diese Masseinheit im DM explizit als Konzept zu modellieren.
  - Money, Weight, Volume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

Einheiten


Payment
amount : Number
not useful
Payment
Has-amount
*
1
Quantity
amount : Number
Is-in
*
1
Unit
...
Payment
amount : Quantity
quantities are pure data values, so are suitable to show in attribute section
better
Payment
amount : Money
variation: *Money* is a specialized *Quantity* whose unit is a currency

Zeitintervalle

- Attribute von Konzepten sind meistens ziemlich stabil (z.B. der Name einer Person), andere Attribute werden jedoch häufig geändert.
- Ist es wichtig, den Verlauf der Änderungen nachzuvollziehen und zukünftige Änderungen zu planen, muss das Attribut mit einem Gültigkeitsintervall versehen werden.



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