

Ausarbeitung Iteration I

Domain Model

siehe [Lar02], Kap. 10

Domain Model

- Illustriert relevante Konzepte des Problembereichs
- Das wichtigste Dokument in OOA
 - (Use-Cases sind nicht OO;
 - Use-Cases sind ein wichtiges Dokument der Anforderungsanalyse)

Domain Model

- Repräsentation der Konzepte der *realen Welt*,
 - *nicht* von Software-Komponenten
- Umsetzung: UML class diagrams



- Bewältigung von Komplexität:
 - *divide-and-conquer*
- Strukturierte Analyse: Zerlegung entlang des Prozeß bzw. der Funktion
- Objekt-Orientierte Analyse: Zerlegung entlang Konzepten oder “Dingen” des Problembereichs

Empfehlung

- Das Domain Model ist *kein* ERM
- Konzepte, die nicht dauerhaft gespeichert werden, dürfen nicht entfallen
- Konzepte
 - Gegenstände, Definitionen/Spezifikationen, Orte, Vorgänge, Rollen von Personen, Behältnisse, Organisationen, Ereignisse, Regeln
- *Noun Phrase Identification*

Empfehlung

- “Kartograph”
 - Verwende die bekannten Namen
 - Das Vokabular des Anwendungsgebiets
 - Verzichte auf irrelevante Details
 - Zeichne nichts, was nicht da ist

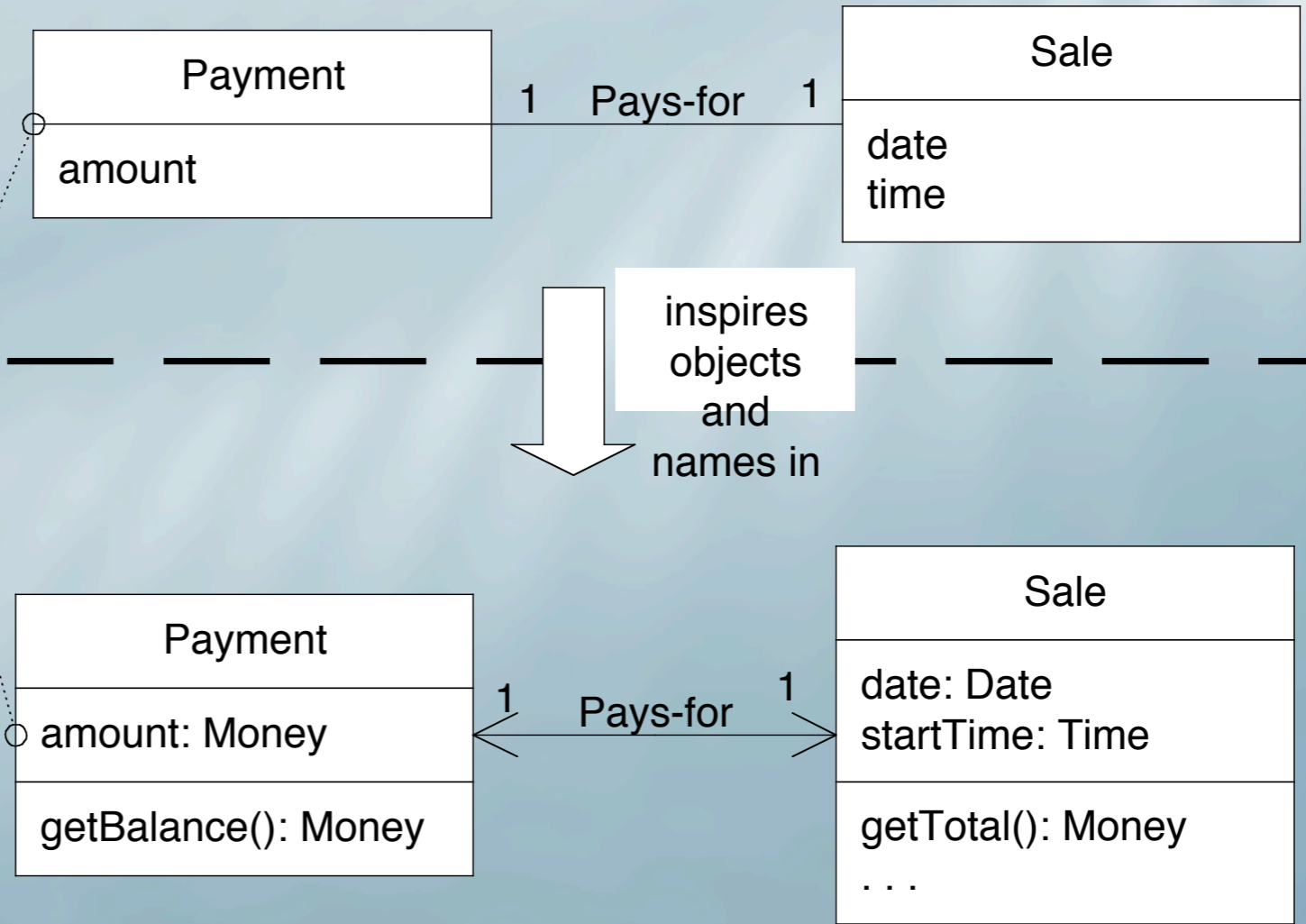
- Das Domain Model ist weder richtig oder falsch
- Es ist mehr oder weniger nützlich
- Es ist ein Kommunikationsmittel

A Payment in the Domain Model is a concept, but a Payment in the Design Model is a software class. They are not the same thing, but the former *inspired* the naming and definition of the latter.

This reduces the representational gap.

This is one of the big ideas in object technology.

UP Domain Model
 Stakeholder's view of the noteworthy concepts in the domain.



UP Design Model

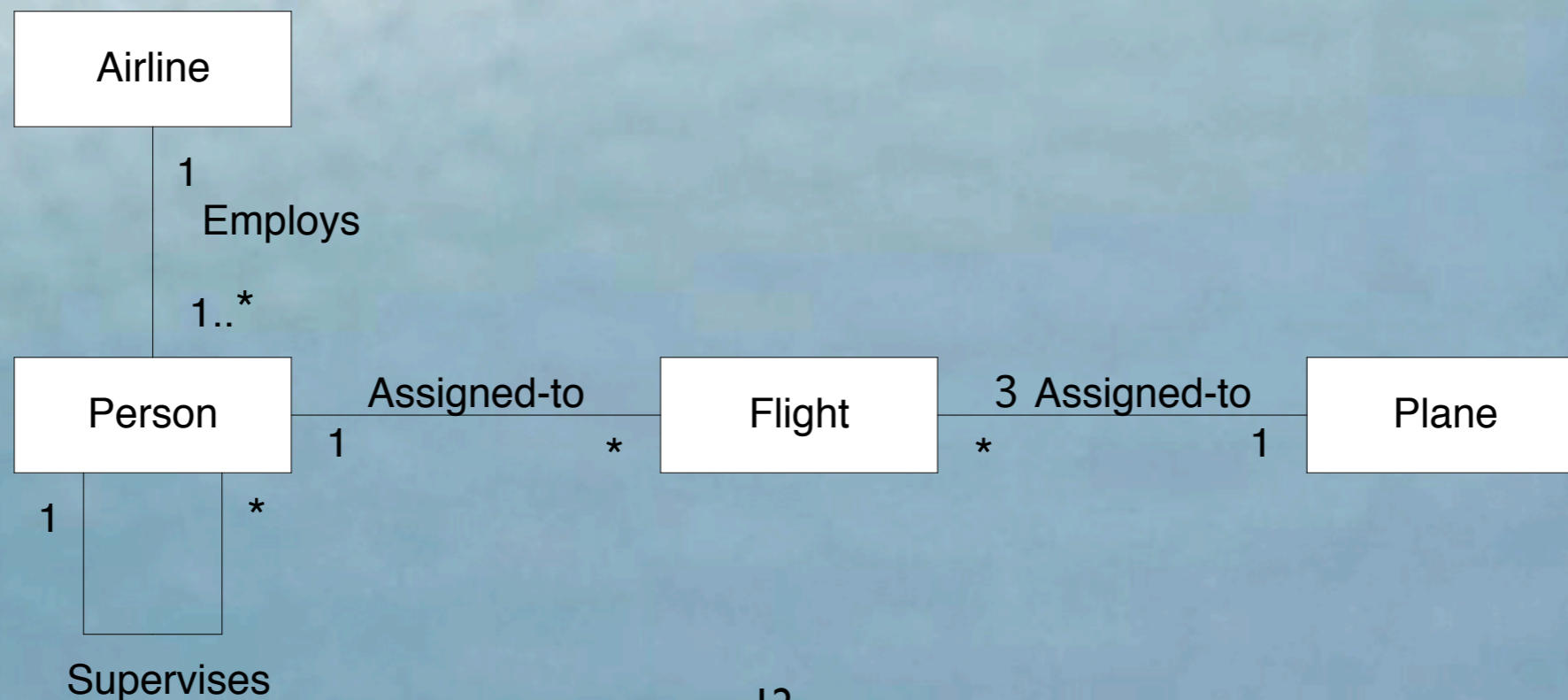
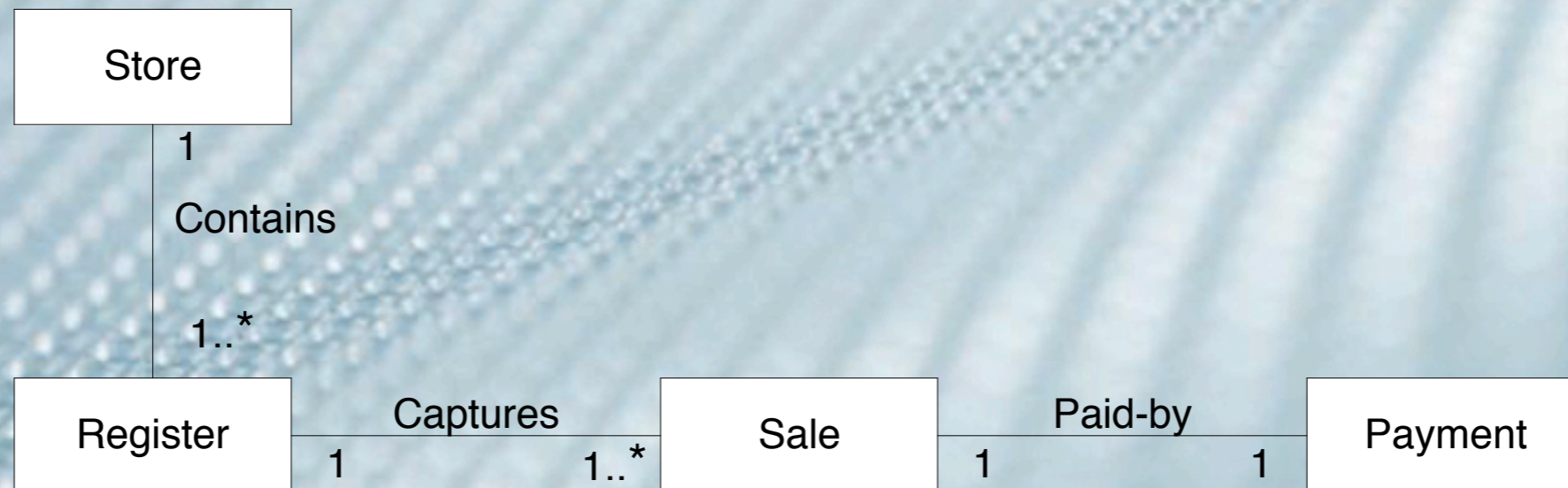
The object-oriented developer has taken inspiration from the real world domain in creating software classes.

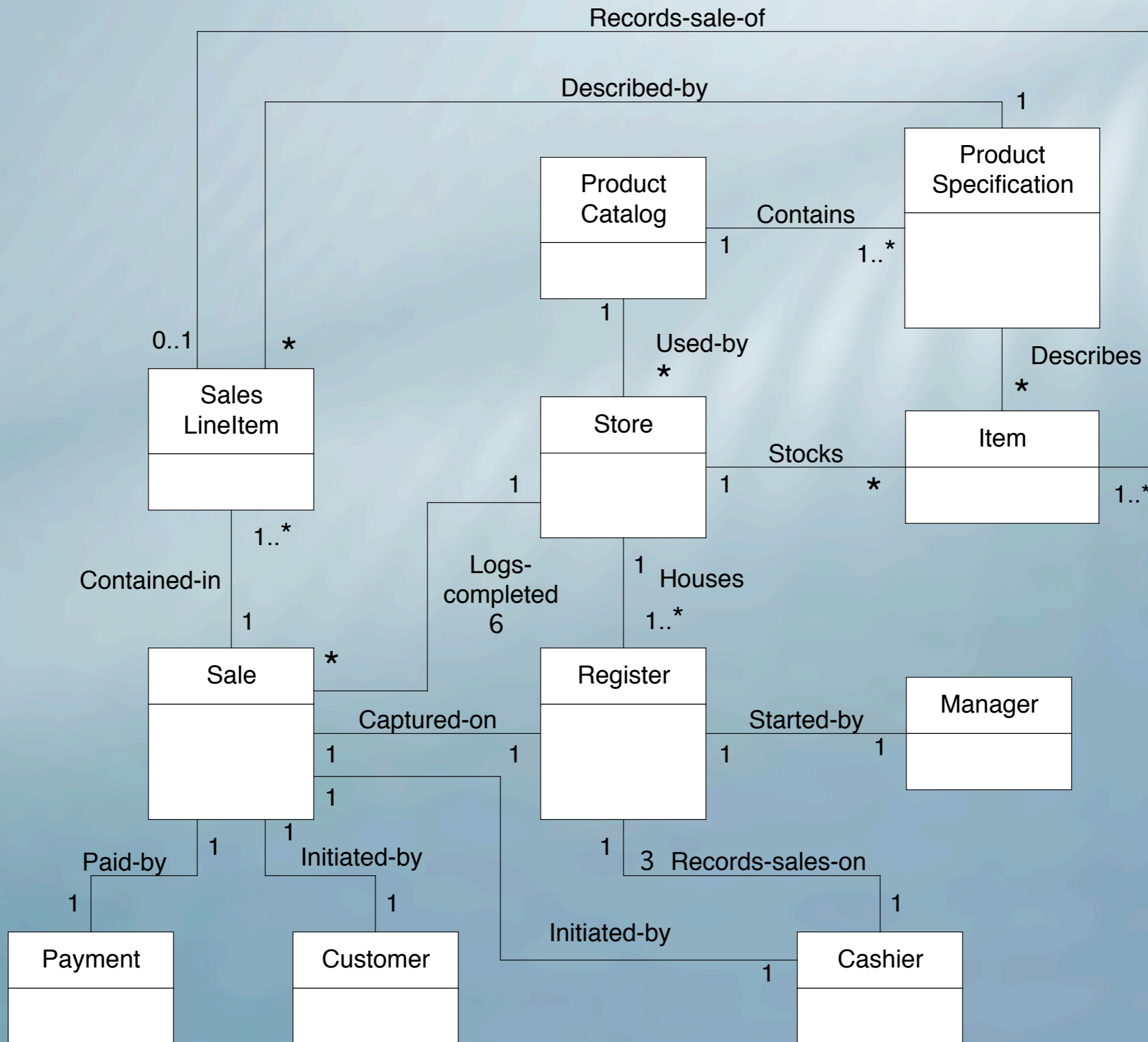
Therefore, the representational gap between how stakeholders conceive the domain, and its representation in software, has been lowered.

Assoziationen

- Inhaltlich Beziehung zwischen konzeptionellen Klassen, z.B.
- Nur solche Aufzeigen, die “gemerkt” werden müssen (dauerhaft gespeichert) *und* für die Anwendung notwendig sind.
- A ist Teil von/enthalten in B, A ist eine Beschreibung von B, A ist ein Posten von B, A verwendet, managt B, A kommuniziert mit B, A gehört B, A ist ein Ereignis im Zusammenhang von B

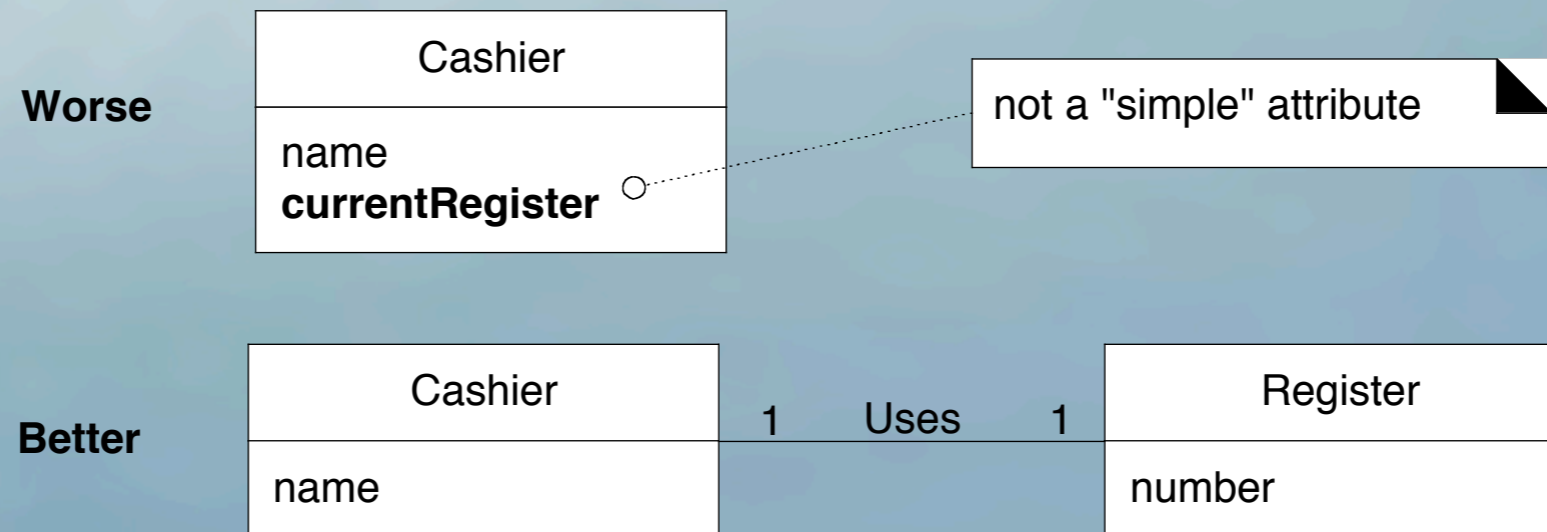
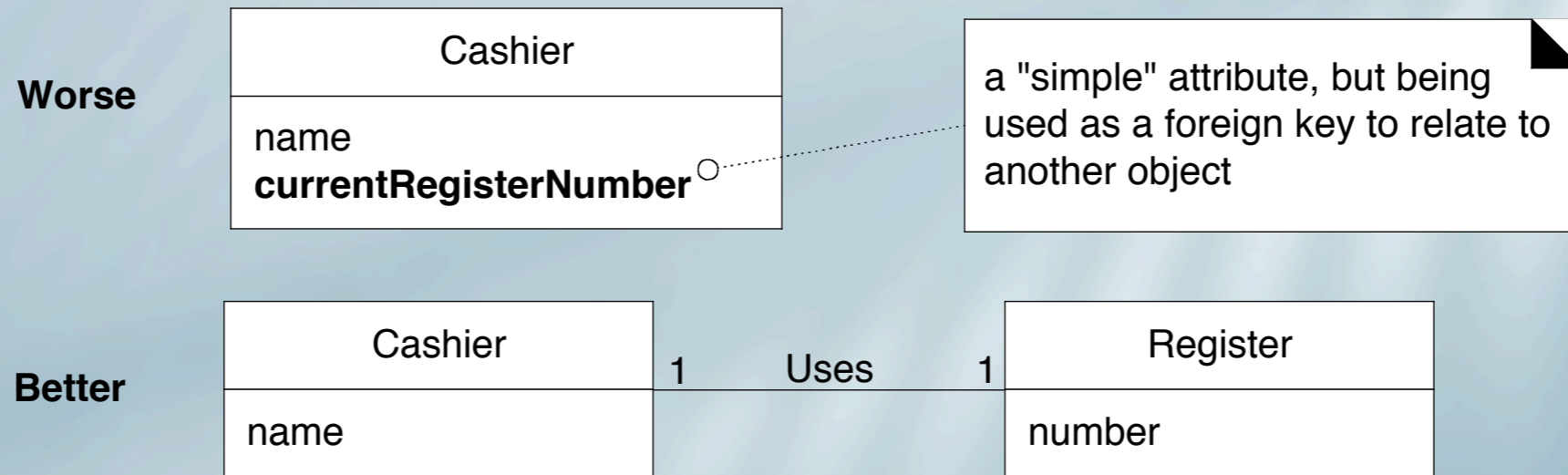
UML Notation: Assoziationen

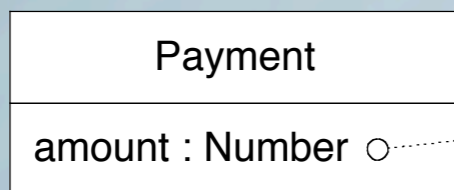




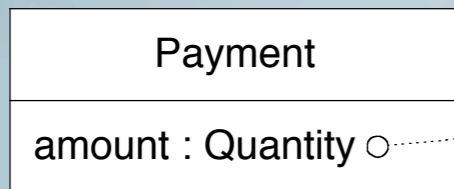
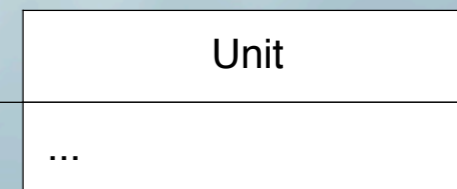
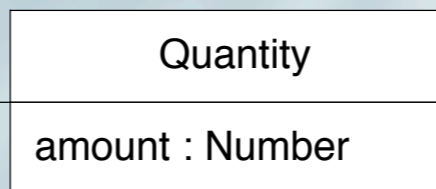
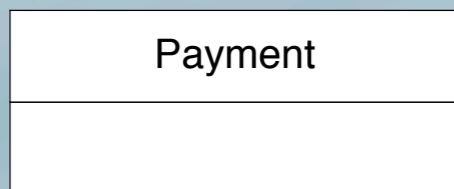
Attribute

- Ein Attribut ist ein (logischer) Wert einer Eigenschaft eines Objekts
- Attribute sind simple/primitive Datentypen
- Komplexe Ausprägungen sind meist Assoziationen zu anderen Objekten

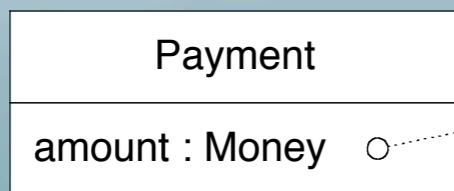




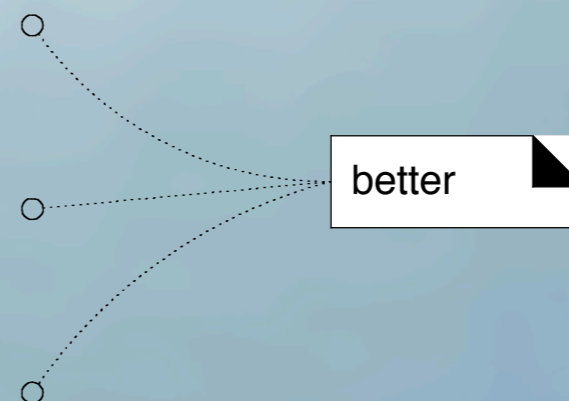
not useful



quantities are pure data values, so suitable to show in attribute section



variation: *Money* is a specialized *Quantity* whose unit is a currency



Sample UP Artifacts

