

Definition: Vereinfachtes UML-Klassendiagramm.

Aufbau: Fachliche Begriffe mit ihren Attributen, setzt Begriffe in Beziehung zueinander. Geht nur um die Problemstellung und das Fachgebiet.

0.1 Domänenmodell als vereinfachtes UML Klassendiagramm

Konzepte = Klassen

Eigenschaften = Attributen (Typenangabe entfällt)

Assoziationen = Beziehungen zwischen Konzepten mit Multiplizitäten an beiden Enden.

Nur wenn es einen guten Grund gibt:

- **Aggregation** = Keine echte Semantik, als Abkürzung für "hat".
- **Komposition** = z.B. wenn Produktkatalog gelöscht wird, dann auch die darin enthaltenen Produktbeschreibungen. Abkürzung "bietet an".

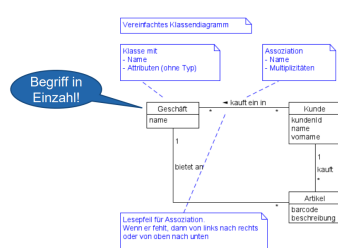


Abbildung 1: DM1

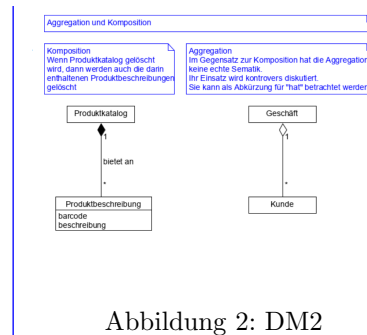


Abbildung 2: DM2

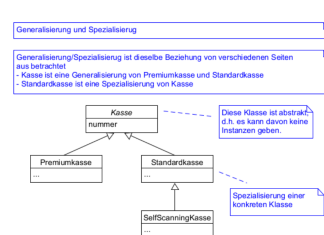


Abbildung 3: DM3

0.2 Vorgehen

1. Konzepte identifizieren
 - (a) Fachwissen und Erfahrung verwenden
 - (b) Substantive aus Anwendungsfällen
 - (c) Kategorienliste verwenden
2. Attributen
 - (a) Fachwissen verwenden
3. Konzepte in Verbindung zueinander setzen
 - (a) Fachwissen verwenden
 - (b) Kategorienliste verwenden
4. Auftraggeben und/oder Fachexperten beiziehen
5. Vorgehensweise eines Kartografen

0.2.1 Kategorienliste

Kategorie	Mögliche Konzepte für DM
Geschäftstransaktionen	
• Transaktionen als Ganzes	Sale
• Transaktionsposition	Sale, Item
• 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

Abbildung 4: Kategorienliste1

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

Abbildung 5: Kategorienliste2

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

Abbildung 6: Kategorienliste3

0.3 Datentypen von Attributen

- Wenn nötig: eigene Datentypen als **Konzepte**
- Dann definieren wenn:
 - Typ aus mehreren **Abschnitten** (wie Tel.Nr)
 - **Operationen** darauf sind möglich (Validierung Kreditkartennummer)
 - Hat selber **eigene Attribute** (Verkaufspreis mit Anfangs & Enddatum)
 - **Verknüpft** mit Einheit (Preis mit Währung)

0.3.1 Anti-Pattern



Abbildung 7: AntiPattern Bad

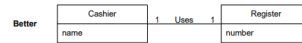


Abbildung 8: AntiPattern Good

Assoziationen statt Attribute

0.4 Vorgehensweise eines Kartografen

- Vorhandene Begriffe oder Wissen einsetzen (Gebiete besuchen, Bewohner nach Begriffen befragen)
- Unwichtiges weglassen
- Nichts hinzufügen, was es (noch) nicht gibt
 - **Ausnahme:** System, das entwickelt wird, kann eingetragen werden
- Nur analysieren, (noch) keine Lösungen entwerfen

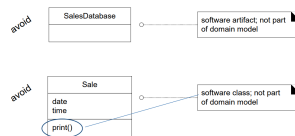


Abbildung 9: AntiPatternSoftware

0.5 Domänenmodell vollständig Beispiel

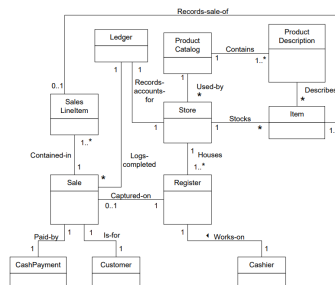


Abbildung 10: BeispielDomänenmodell

0.6 Analysemuster

• Beschreibungsklassen

- Item = Physischer Gegenstand oder Dienstleistung
- Mehrere Artikel desselben Typs
- Attribute (description, price, serial number, itemID)



Abbildung 11: Beschreibungsklassen

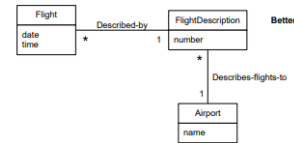


Abbildung 12: BeschreibungsklasseFlug

• Generalisierung / Spezialisierung

- Spezialisierung als is a" Beziehung zu



Abbildung
GeneralisierungSpezialisierung

13:

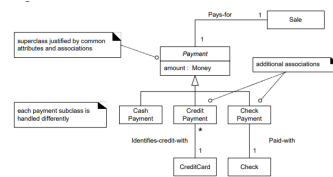


Abbildung
GeneralisierungSpezialisierungAttribute 14:

• Komposition

• Zustände

- Eigene Hierarchie für Zustände definieren:

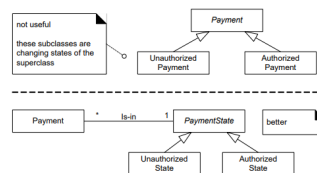


Abbildung 15: Zustände

• Rollen

- Dasselbe Konzept kann unterschiedliche Rollen einnehmen:

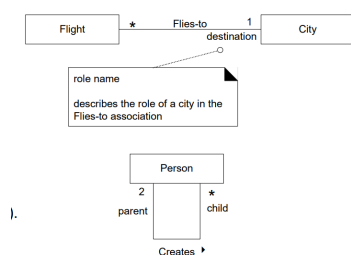


Abbildung 16: DMRolleName 3

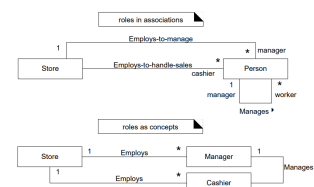


Abbildung 17: DMRolleKonzept

- **Assoziationsklasse**

- Wenn Assoziationen eigene Attribute haben (MerchantID für Kreditkarte Geschäft; \rightarrow AuthorizationService):

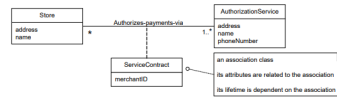


Abbildung 18: Assoziationsklasse

- **Einheiten**

- Manchmal sinnvoll explizit als Konzept zu modellieren :

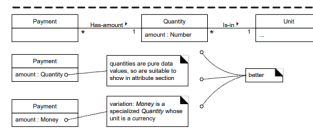


Abbildung 19: Einheiten

- **Zeitintervalle**

- Gültigkeitsintervall für sich ändernde Attribute :



Abbildung 20: Zeitintervall