Kapitel 12: Datenmodellierung mit ERM & UML

Einordnung

Ziel und Infrastruktur der Datenmodellierung:
Beschreibung der für ein IS relevanten Informationszusammenhänge der realen Welt in einer hinreichend präzisen und konzisen Weise unter Verwendung von:

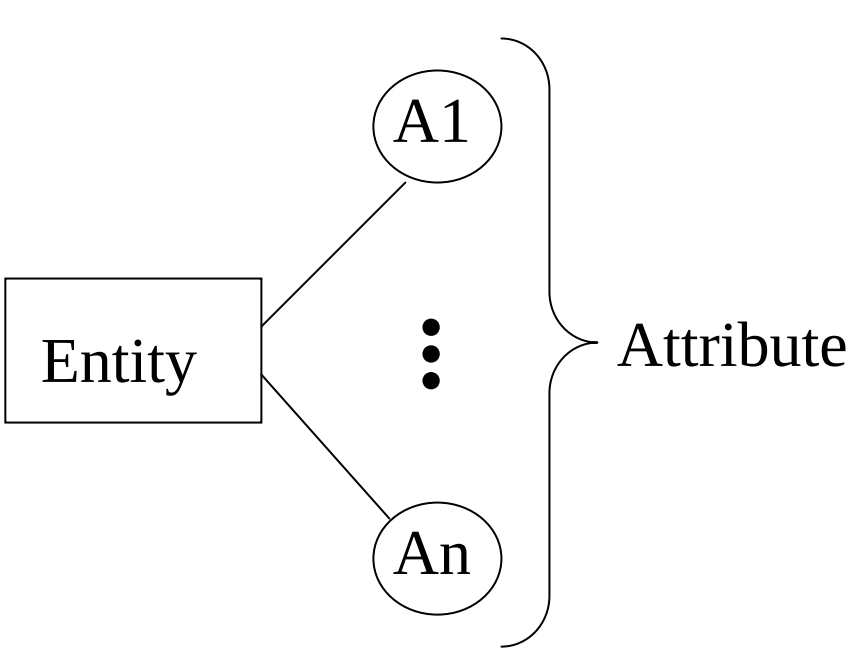
- einer „semantischen“ Modellierungssprache
- einer (i.d.R. informell-groben) Entwurfsmethodologie
- Sofwarewerkzeugen (DB-Design-Tools, Repositories)

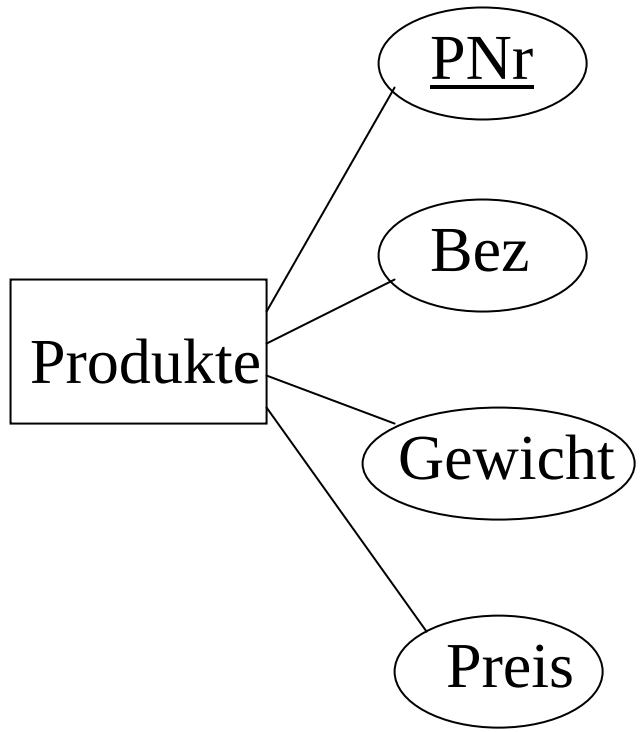
Grobe Vorgehensweise bei der Entwicklung von IS:

1) Anforderungsanalyse
2) Datenmodellierung
   (plus Funktions- und Prozessmodellierung,
   $\rightarrow$ Kapitel 13 sowie Vorlesungen über Softwaretechnik)
3) Abbildung des „semantischen“ Datenmodells
   auf (relationale) DB-Schema

12.1 Entities (Objekte) und Entity-Mengen

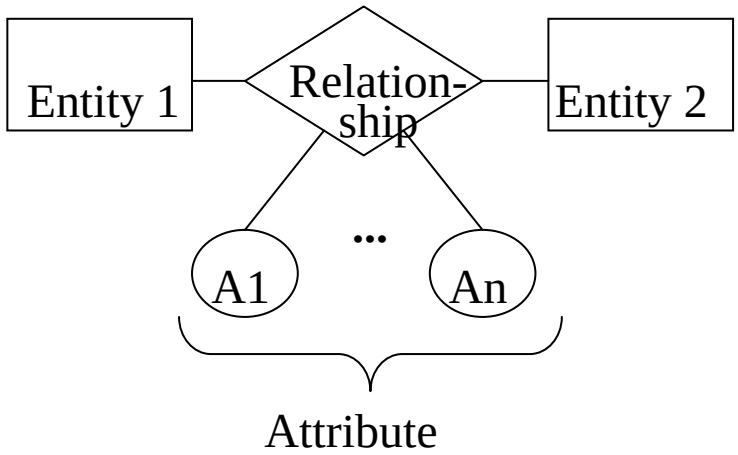
Ein ***Entity*** ist ein durch ***Attribute*** beschriebenes Objekt der realen Welt.
Gleichartige Entities werden zu einem Entity-Typ zusammengefasst.
Entities desselben Typs bilden eine ***Entity-Menge***.
Eine minimale Attributmenge, die jedes Entity einer Entity-Menge eindeutig identifiziert, heißt ***Schlüssel(kandidat).***

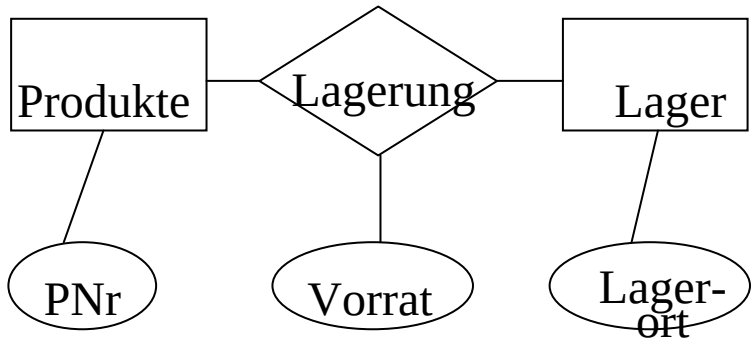


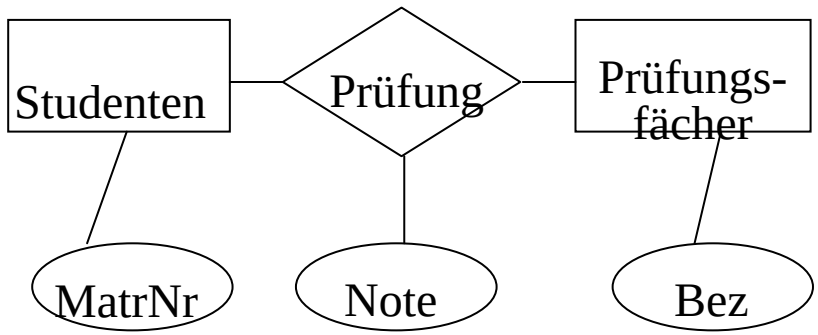


Relationships (Beziehungen, Assoziationen)

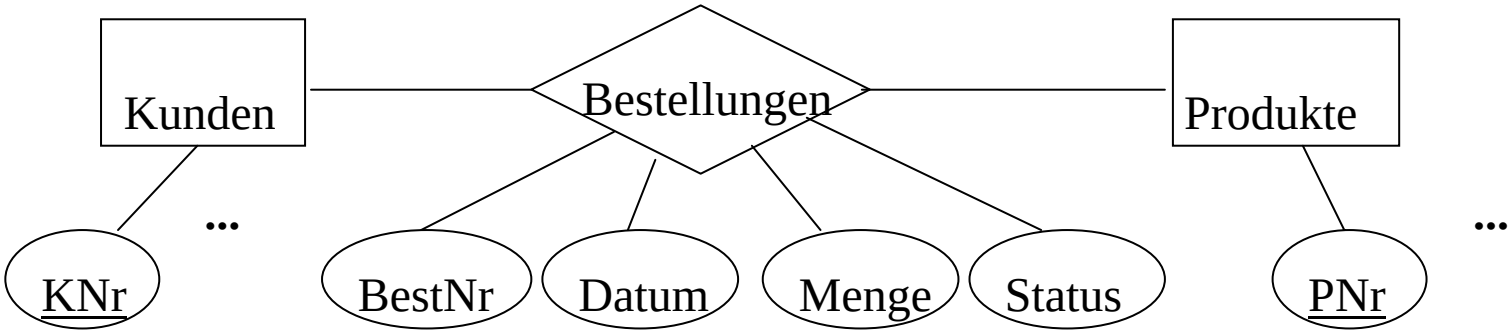
Eine ***Relationship-Menge*** R zwischen Entity-Mengen E1, ..., En ist eine Teilmenge von E1 × ... × En. Ein Element von <e1, ..., en> von R heißt ***Relationship*** zwischen den Entities e1, ..., en.
Eine minimale Teilmenge der Schlüssel von E1, ..., En, die jede Relationship in R eindeutig identifiziert, heißt ***Schlüssel(kandidat).***



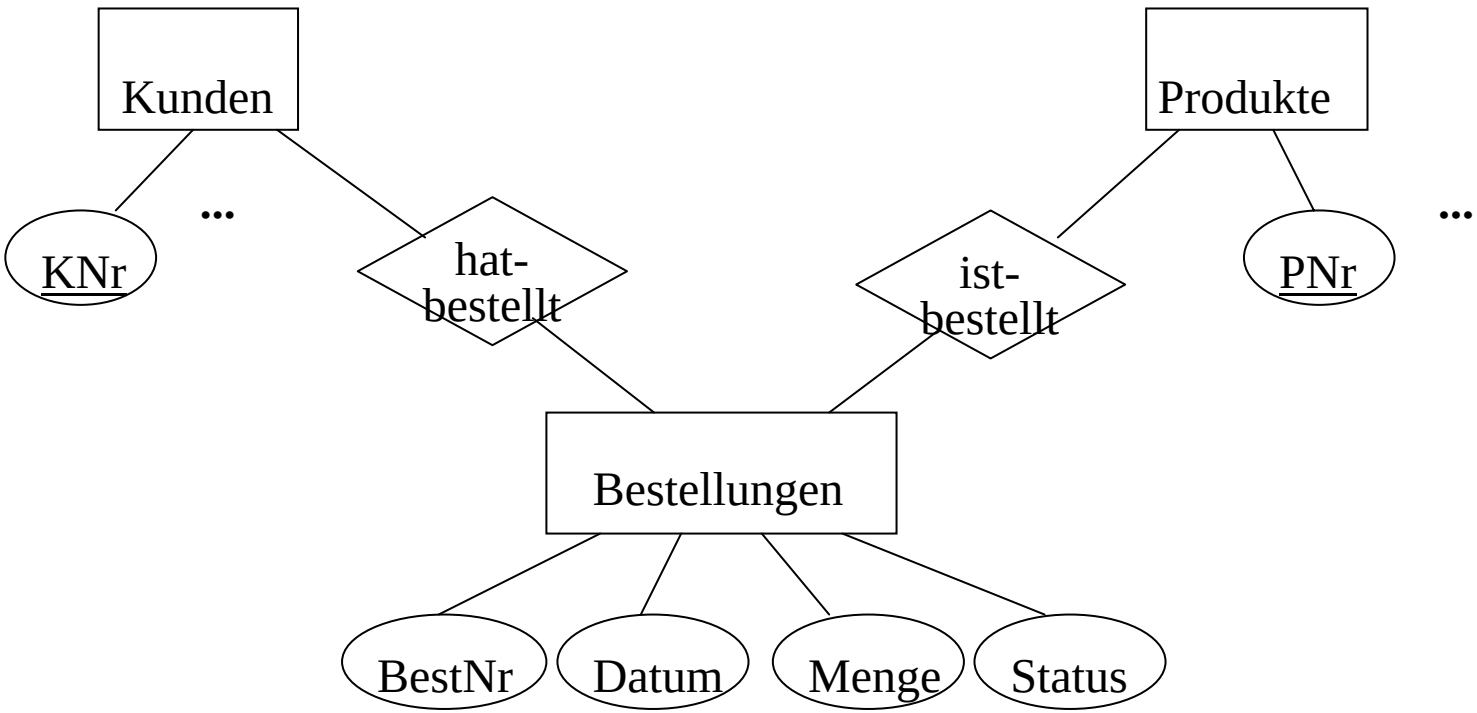




Entity oder Relationship ? (Varianten 1 und 2)



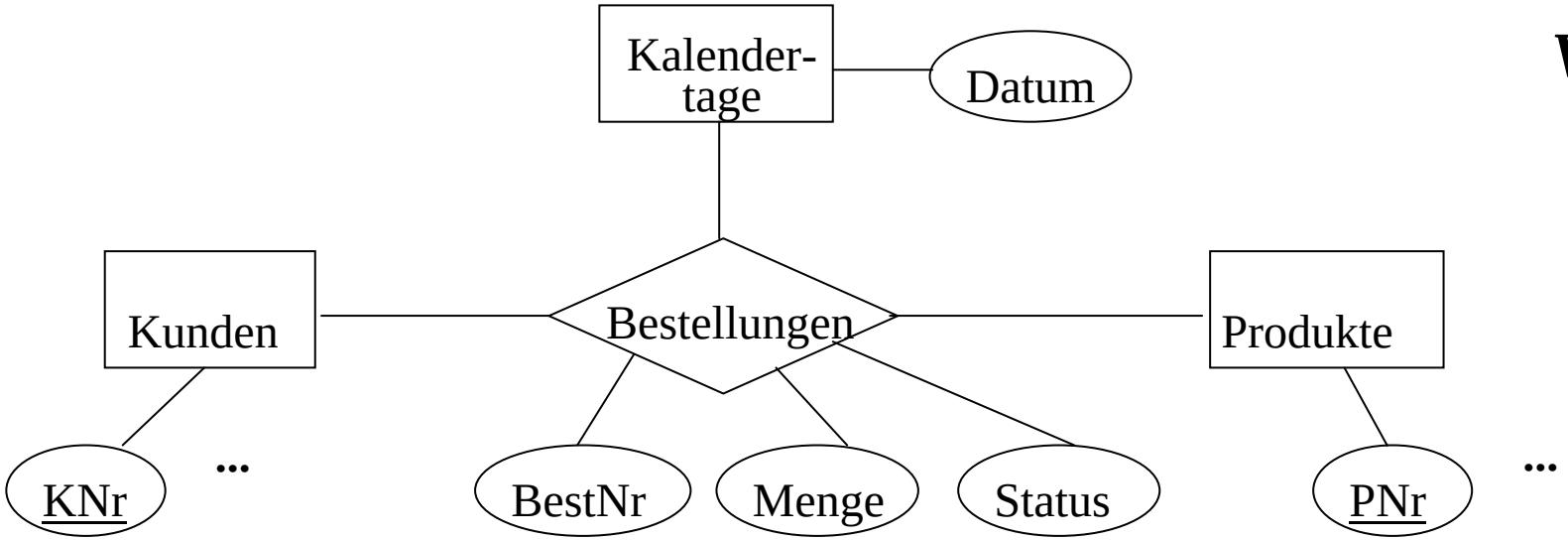
Variante 1



Variante 2

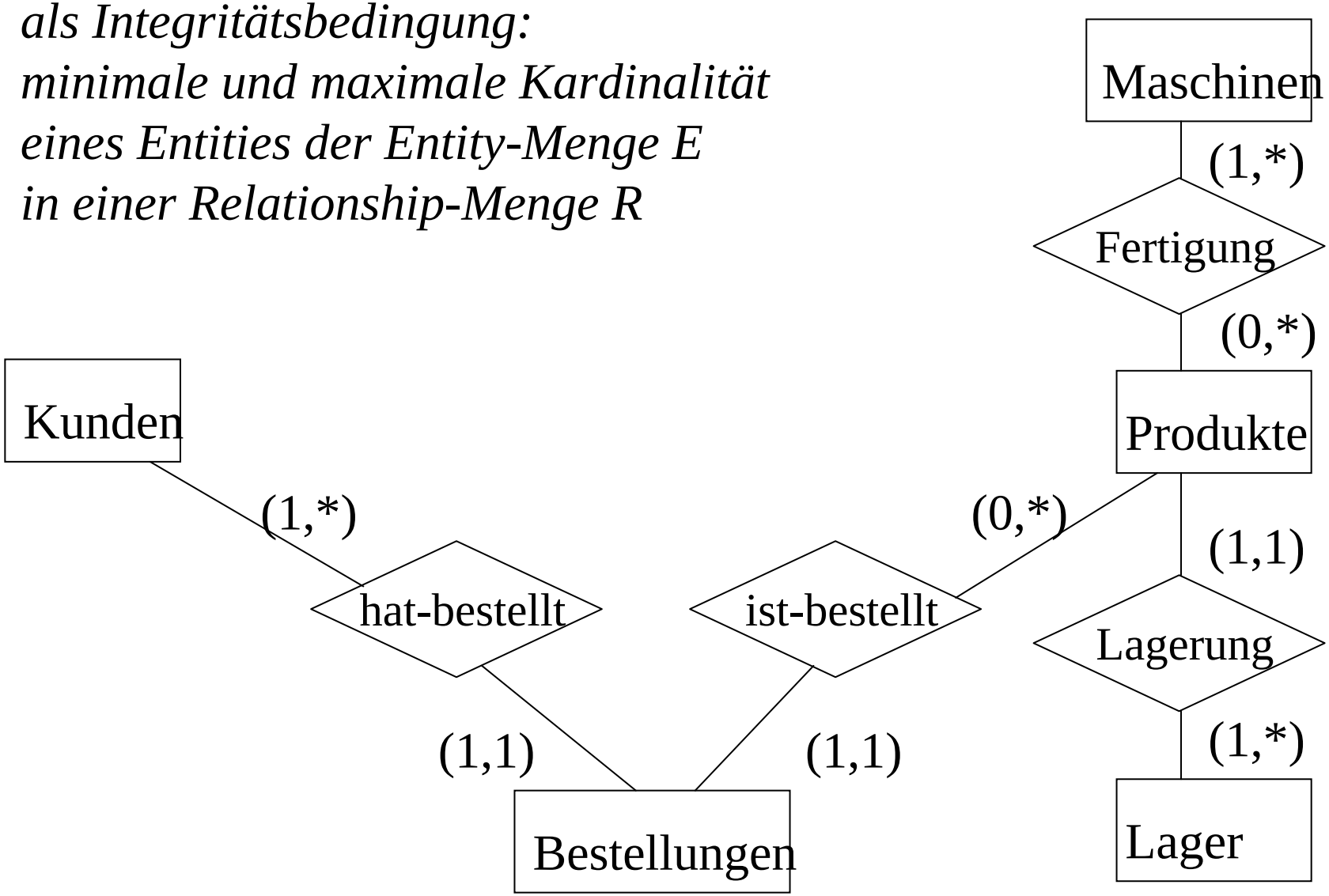
Entity oder Relationship ? (Variante 3)

Variante 3

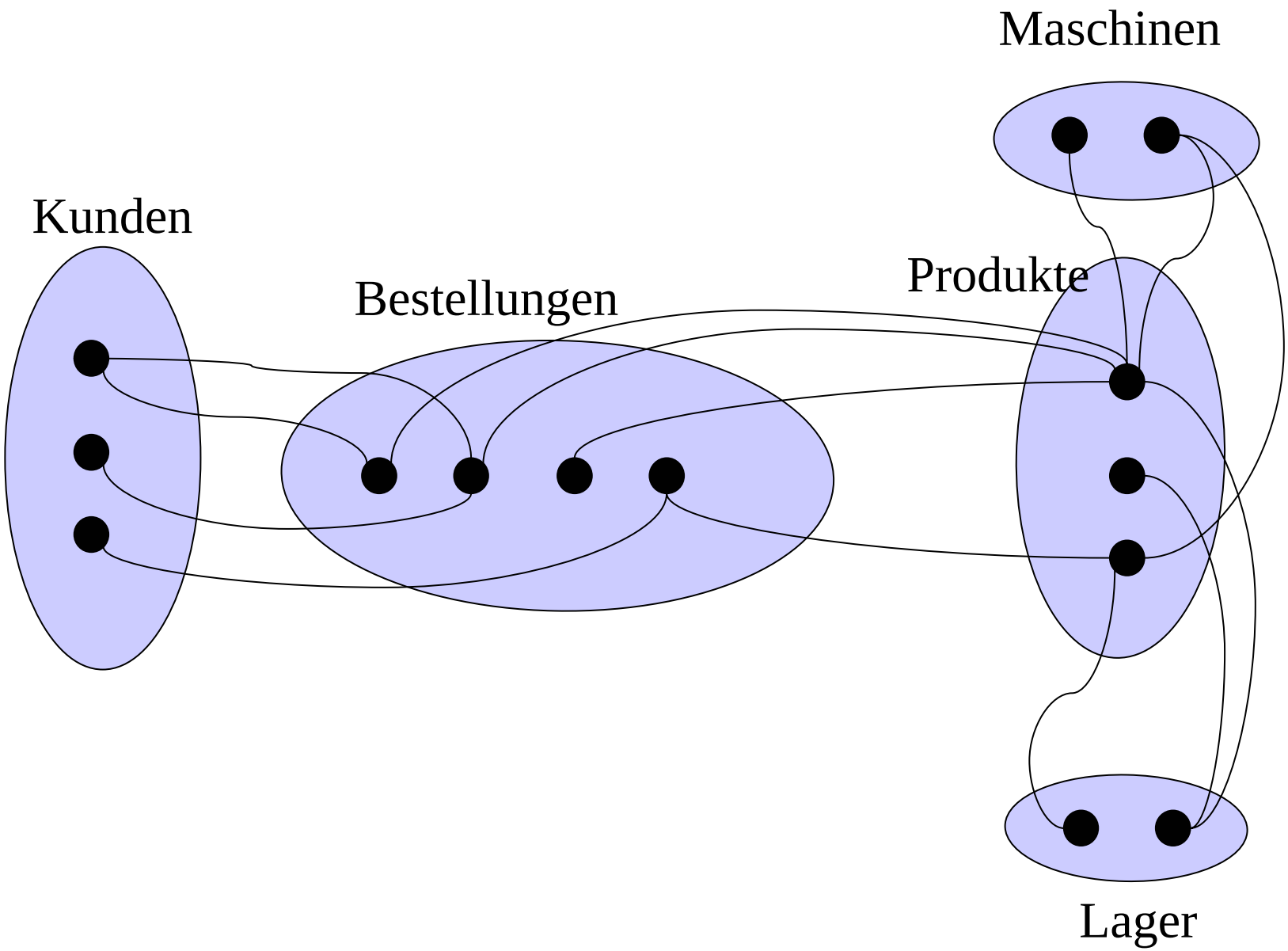


Kardinalitätsbedingungen für Relationships (1)

als Integritätsbedingung:
minimale und maximale Kardinalität
eines Entities der Entity-Menge E
in einer Relationship-Menge R

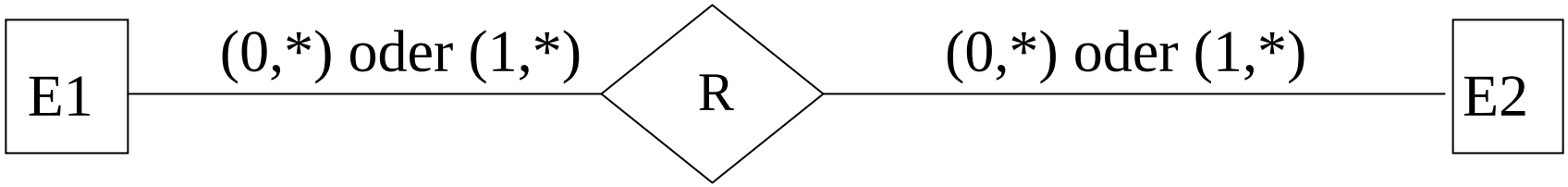


Kardinalitätsbedingungen für Relationships (2)

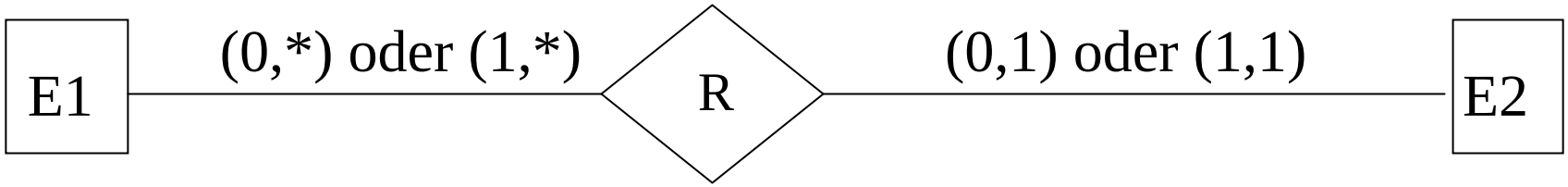


Typen von Kardinalitätsbedingungen (1)

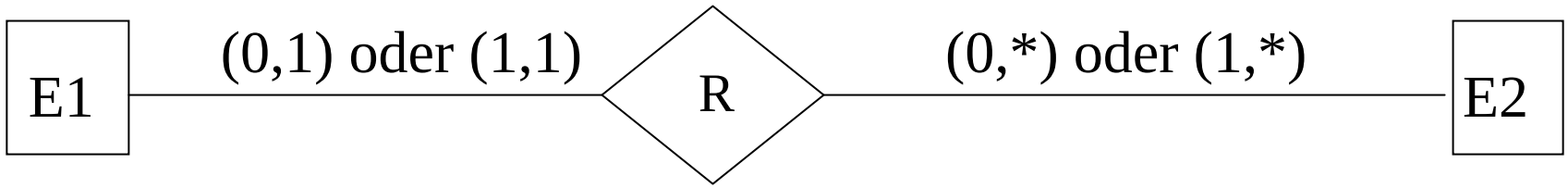
M:N-Relationship



1:N-Relationship

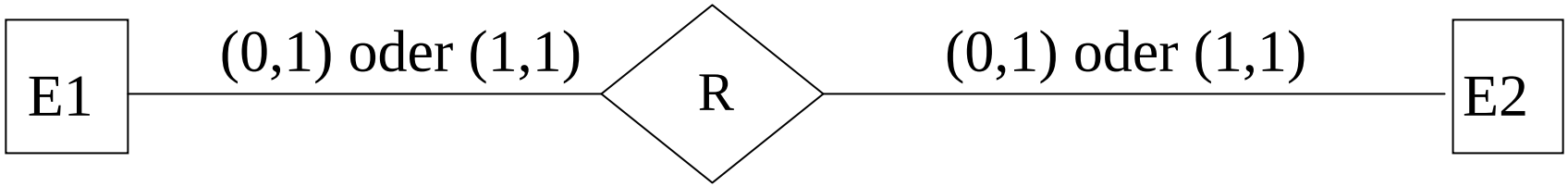


N:1-Relationship

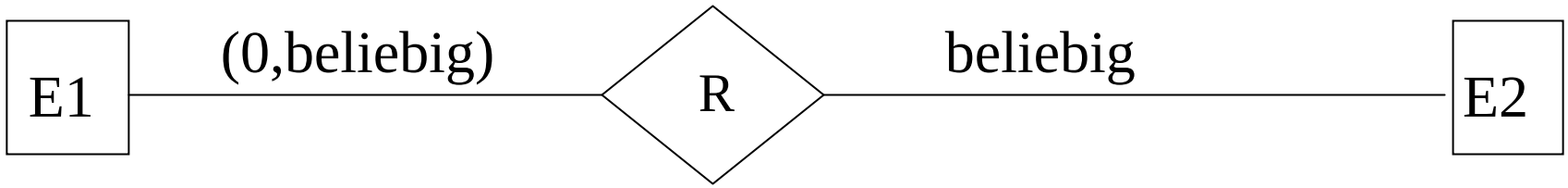


Typen von Kardinalitätsbedingungen (2)

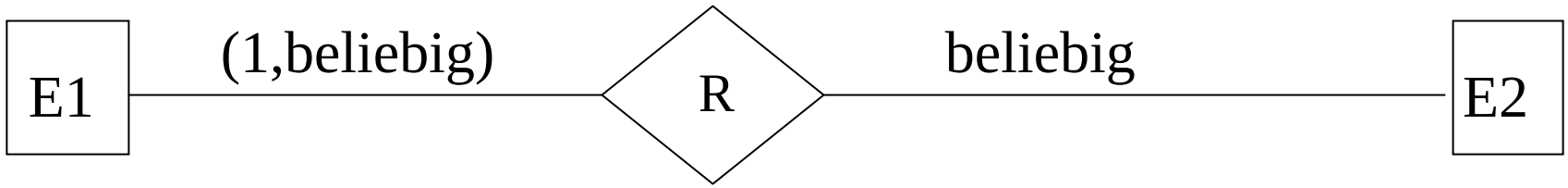
1:1-Relationship



Optionale Rolle von E1 in R

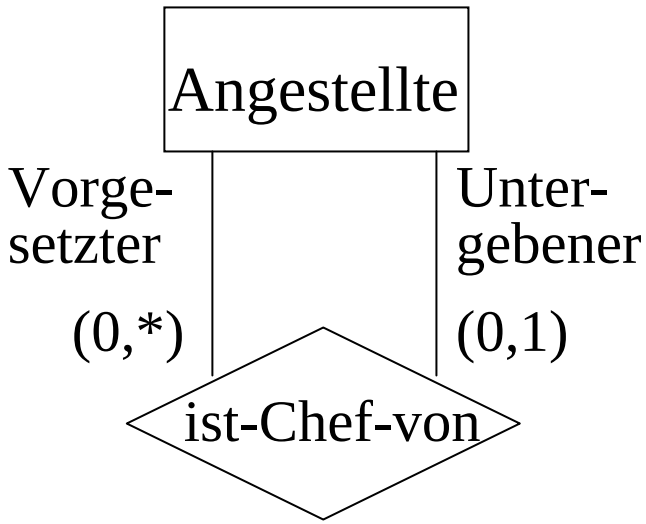


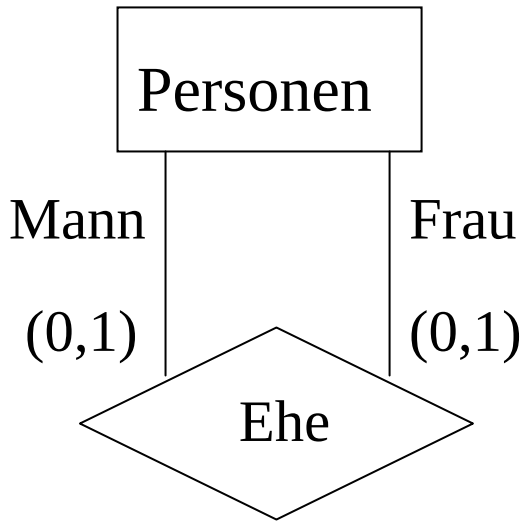
Verpflichtende Rolle von E1 in R



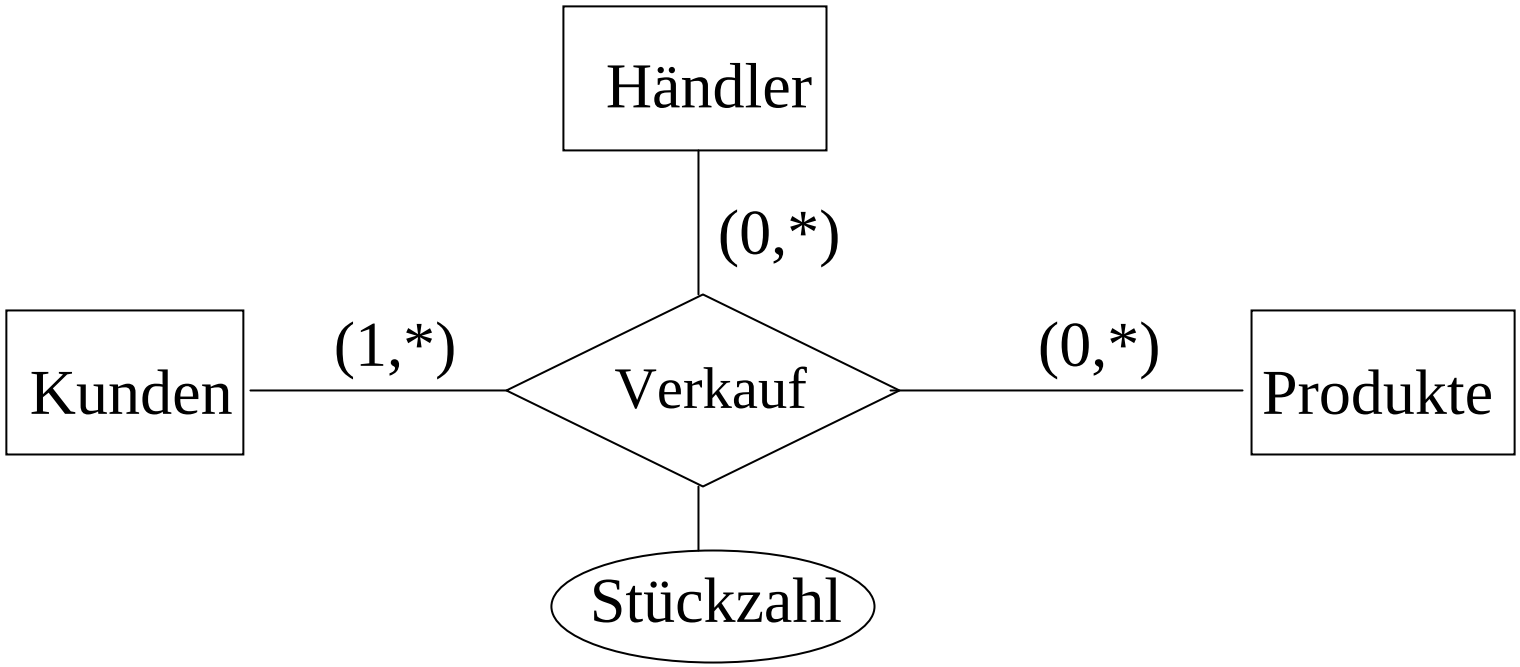
Rekursive Relationships

Relationships auf einer Menge erfordern Rollennamen



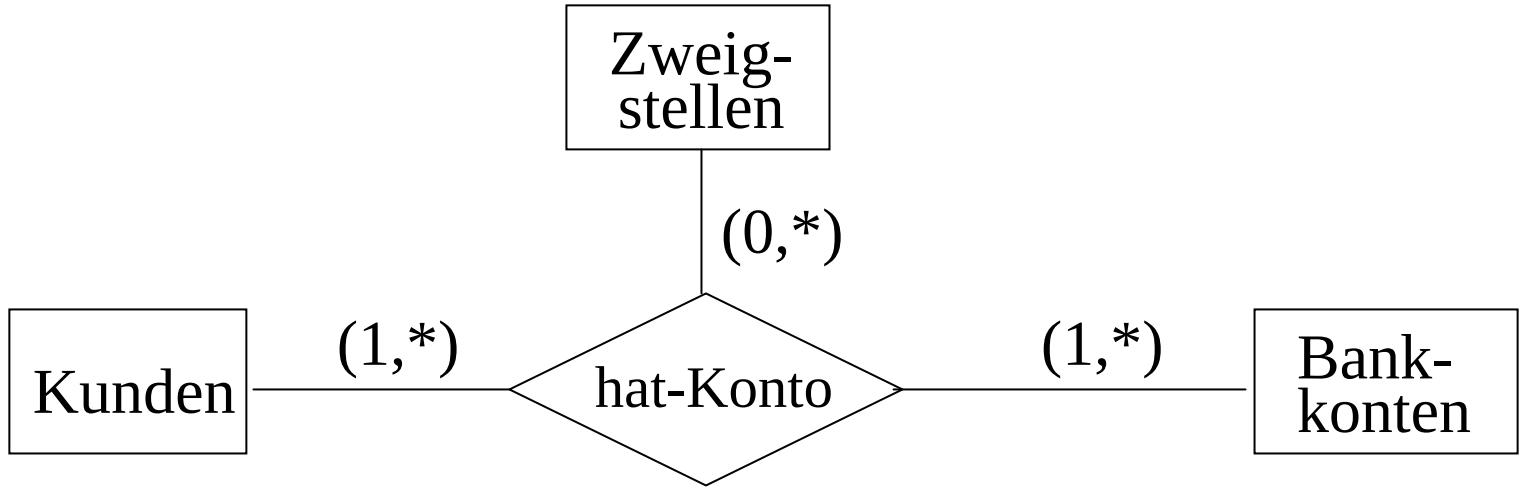


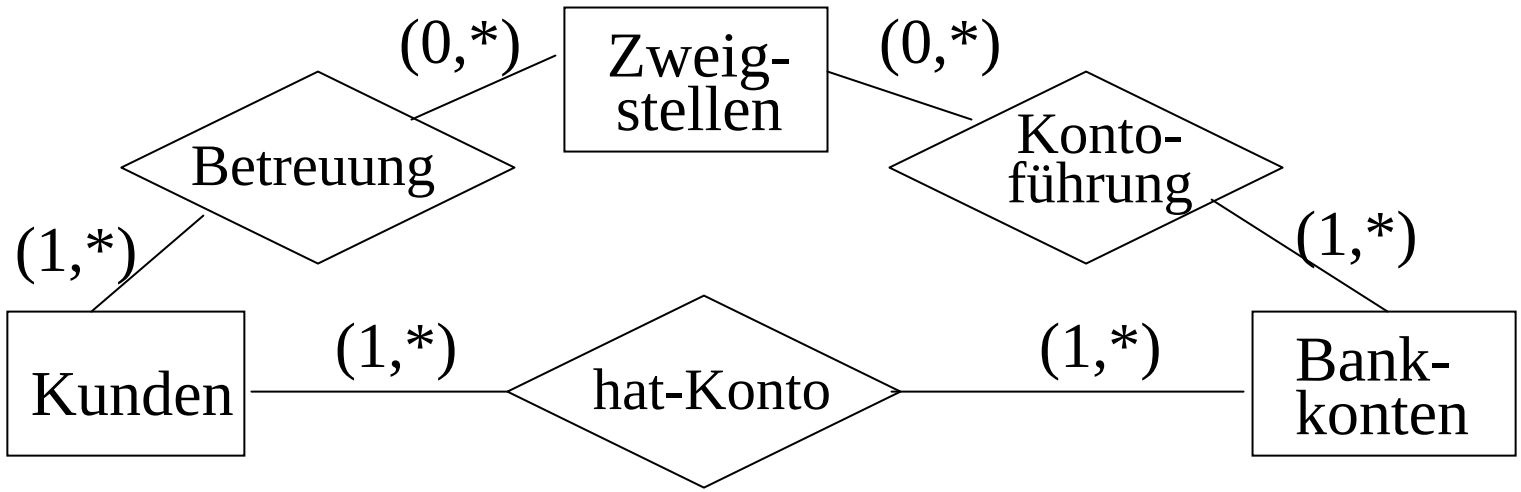
Ternäre Relationships: Beispiel 1



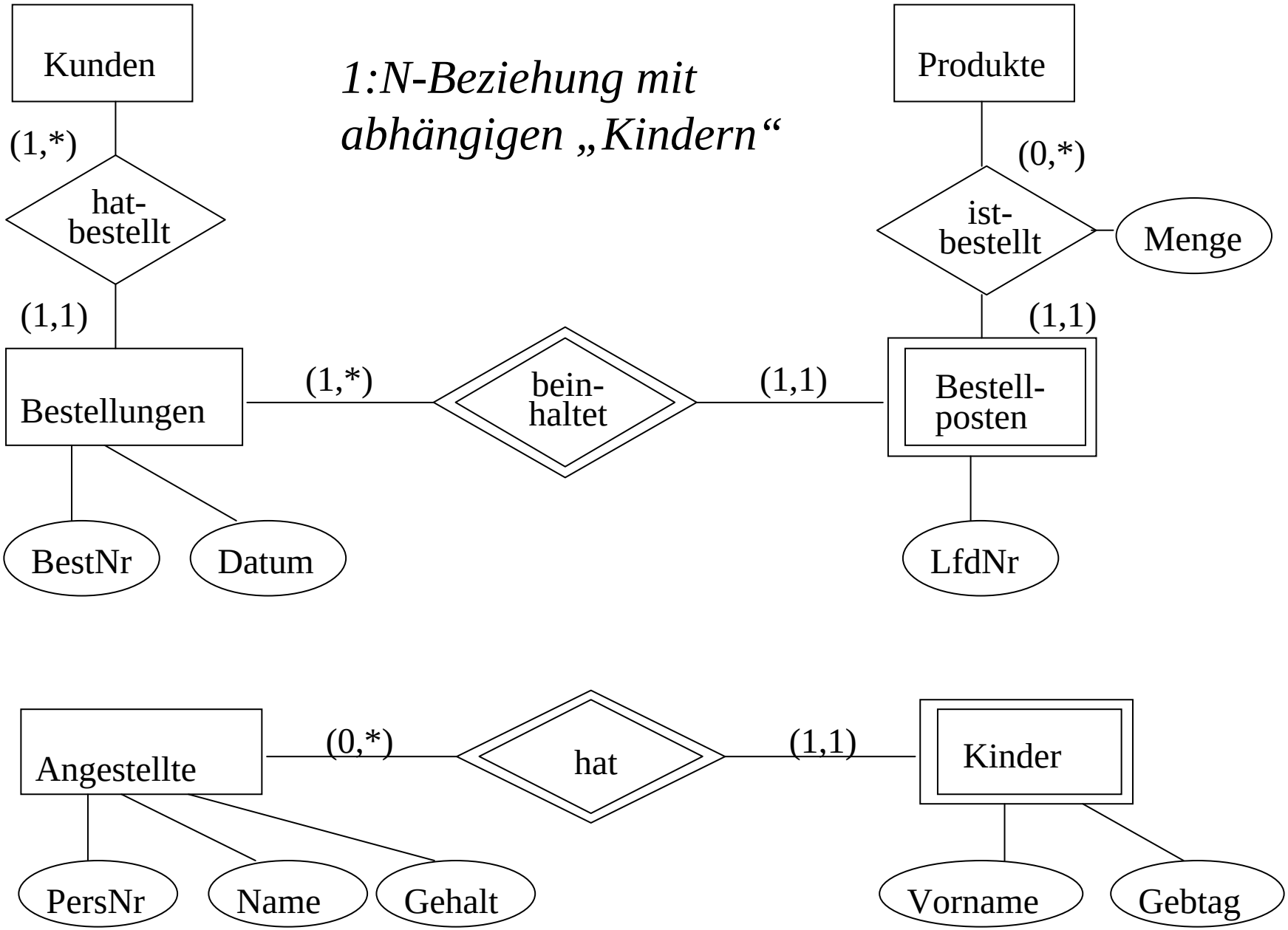
Achtung: Ternäre (und höherstellige) Relationships sind nicht immer zerlegbar in mehrere binäre Relationships

Ternäre Relationships: Beispiel 2



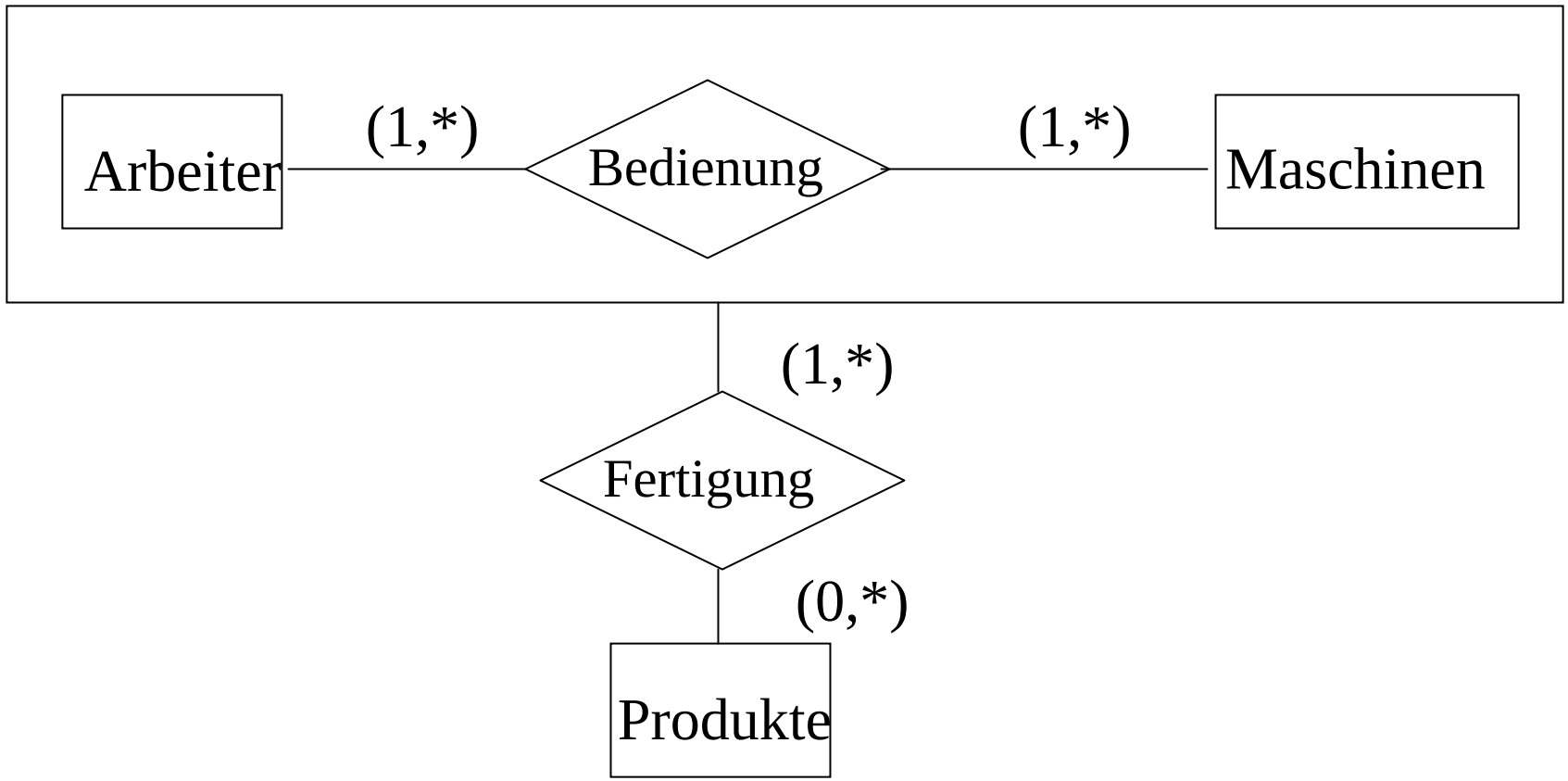


Schwache Entities und Relationships



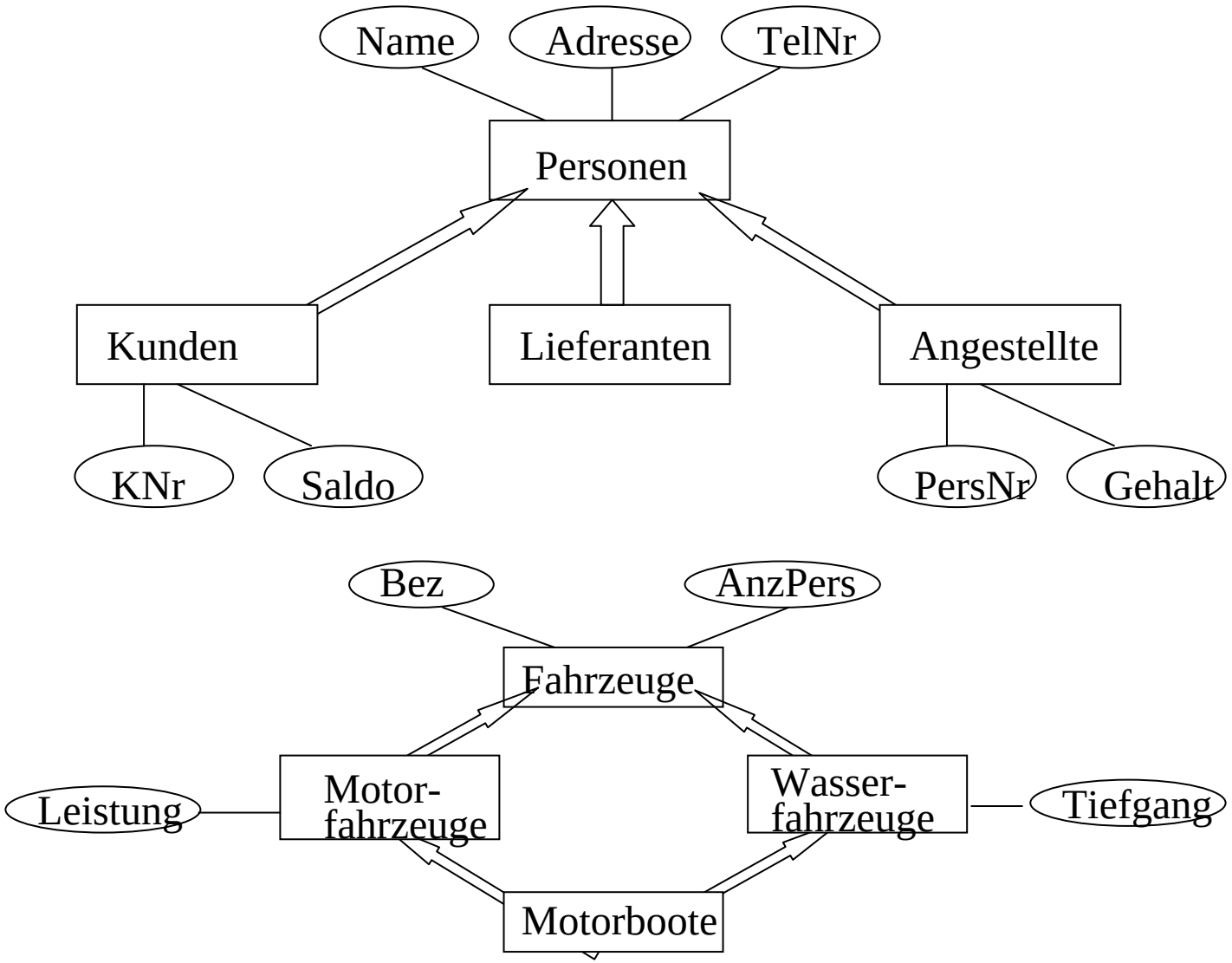
Aggregation

Zusammengesetzte Entities oder Relationships
(→ ERM-Erweiterungen wie z.B. SERM,
siehe z.B. Aris Toolset)



Generalisierung (ISA-Relationships)

1:1-Beziehung zwischen E (Superklasse) und F (Subklasse) mit $F \subseteq E$ und Vererbung der E-Attribute an F



12.2 Entwurfsmethodologie (grob)

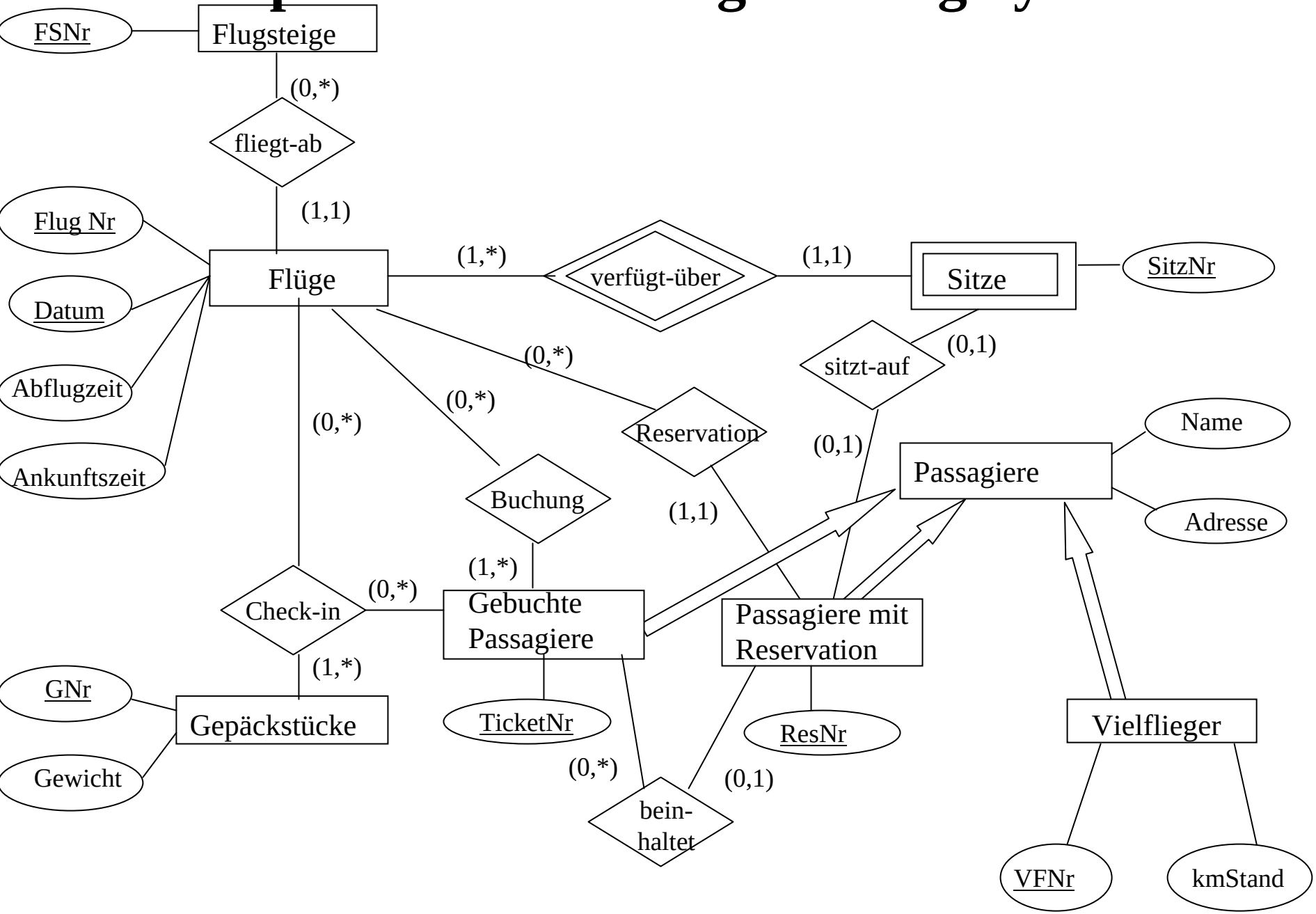
1) Analyse des Informationsbedarfs
2) Festlegung von Entity-Mengen mit ihren Attributen
3) Festlegung von Primärschlüsseln
4) Bildung von Generalisierungsbeziehungen
5) Festlegung von Relationship-Mengen
6) Spezifikation von Kardinalitätsbedingungen
   und ggf. weiteren Integritätsbedingungen

Beispiel Flugbuchungssystem

Es sollen die Informationszusammenhänge für ein Flugbuchungssystem einer Fluggesellschaft modelliert werden.

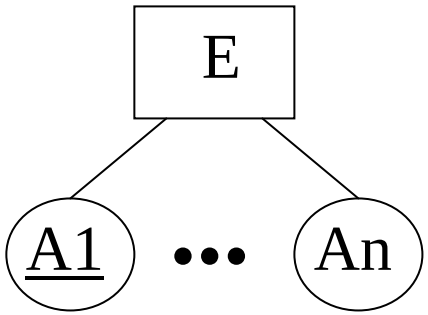
Flüge werden durch eine Flugnummer identifiziert, die für Flüge am selben Tag eindeutig ist.
Passagiere können einen Flug reservieren, was durch eine Reservationsnummer bestätigt wird.
Eine Reservation wird zu einer festen Buchung, indem man ein Ticket käuft.
Bei der Reservation oder später können Passagiere auch eine Sitzplatzreservierung vornehmen.
Für Teilnehmer des Vielfliegerprogramms ist die gesamte mit der Fluggesellschaft geflogene Kilometerzahl von Bedeutung.
Flüge fliegen von einem bestimmten Flugsteig ab.
Passagiere müssen vor dem Abflug eine Check-in-Prozedur durchlaufen.
Dabei können sie auch Gepäckstücke aufgeben.

Simples ERM für Flugbuchungssystem



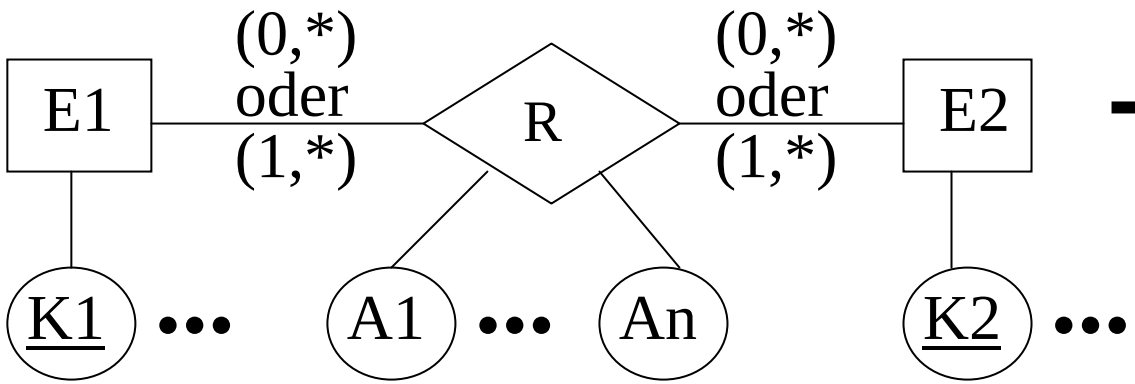
12.3 Abbildung ERM auf Relationenschema (1)

Regel 1: Jede Entity-Menge bildet eine Relation



E (A1, ..., An)

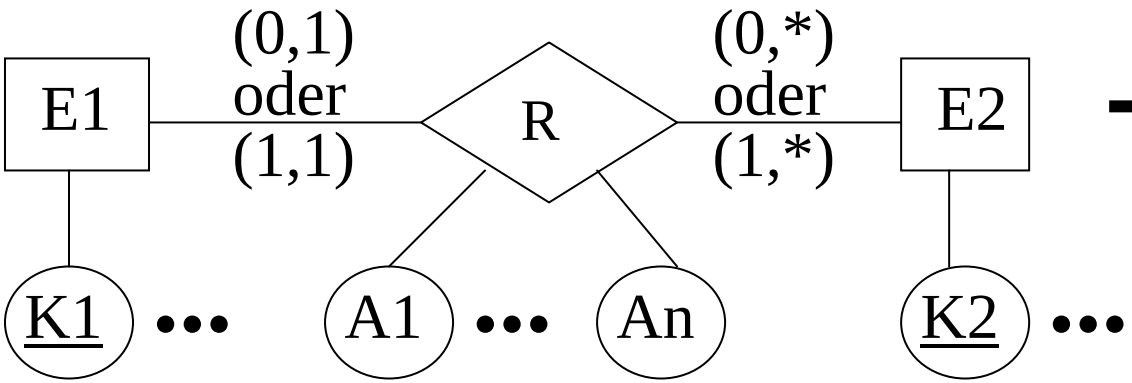
Regel 2: M:N-Relationships bilden eigene Relationen



R (K1, K2,
A1, ..., An)

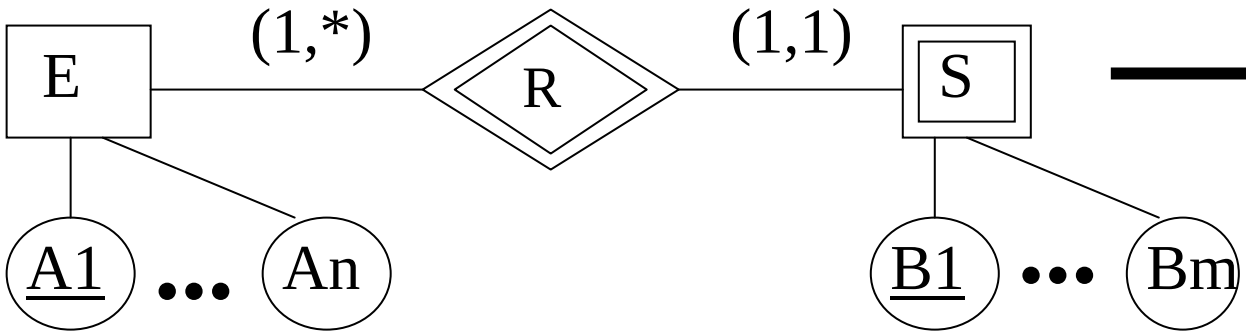
12.3 Abbildung ERM auf Relationenschema (2)

Regel 3: N:1-, 1:N- und 1:1-Relationships können in eine Entity-Relation integriert werden



E1 (<u>K1</u>, ..., A1, ..., An, K2)
E2 (<u>K2</u>, ...)

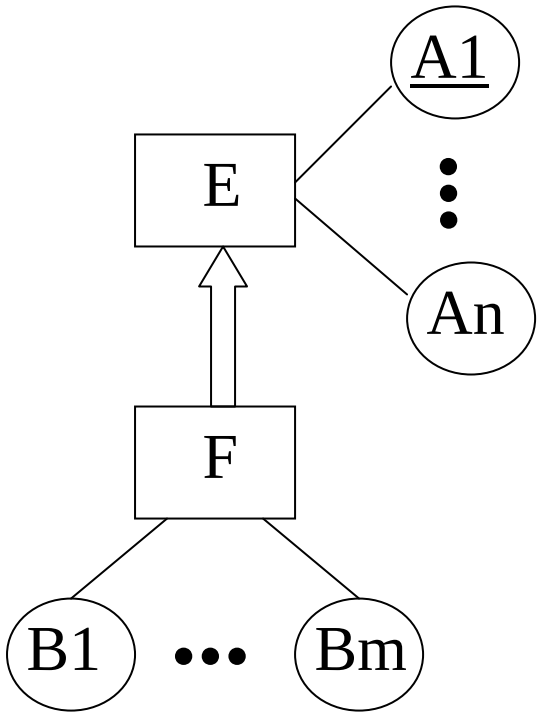
Regel 4: Eine schwache Entity-Menge bildet eine eigene Relation



E (<u>A1</u>, ..., An)
S (<u>A1, B1</u>, ..., Bm)

12.3 Abbildung ERM auf Relationenschema (3)

Regel 5: Generalisierung und Spezialisierung bilden jeweils eigene Relationen



→ E (A1, ..., An)
F (A1, B1, ..., Bm)

→ E (A1, ..., An)
F (A1, ..., An,
B1, ..., Bm)
wobei E nur Tupel enthält für Entities aus E - F

Abbildung für Flugbuchungs-Beispiel

Flüge (FlugNr, Datum, Abflugzeit, Ankunftszeit, FSNr)
Flugsteige (FSNr)
Sitze (FlugNr, Datum, SitzNr)
PassagiereMitReservation (ResNr, Name, Adresse, FlugNr, Datum, SitzNr, TicketNr)
GebuchtePassagiere (TicketNr, Name, Adresse)
Buchungen (TicketNr, FlugNr, Datum)
Vielflieger (VFNr, Name, Adresse, kmStand)
Gepäckstücke (GNr, Gewicht)
Check-In (FlugNr, Datum, TicketNr, GNr)

12.4 Datenmodellierung mit UML

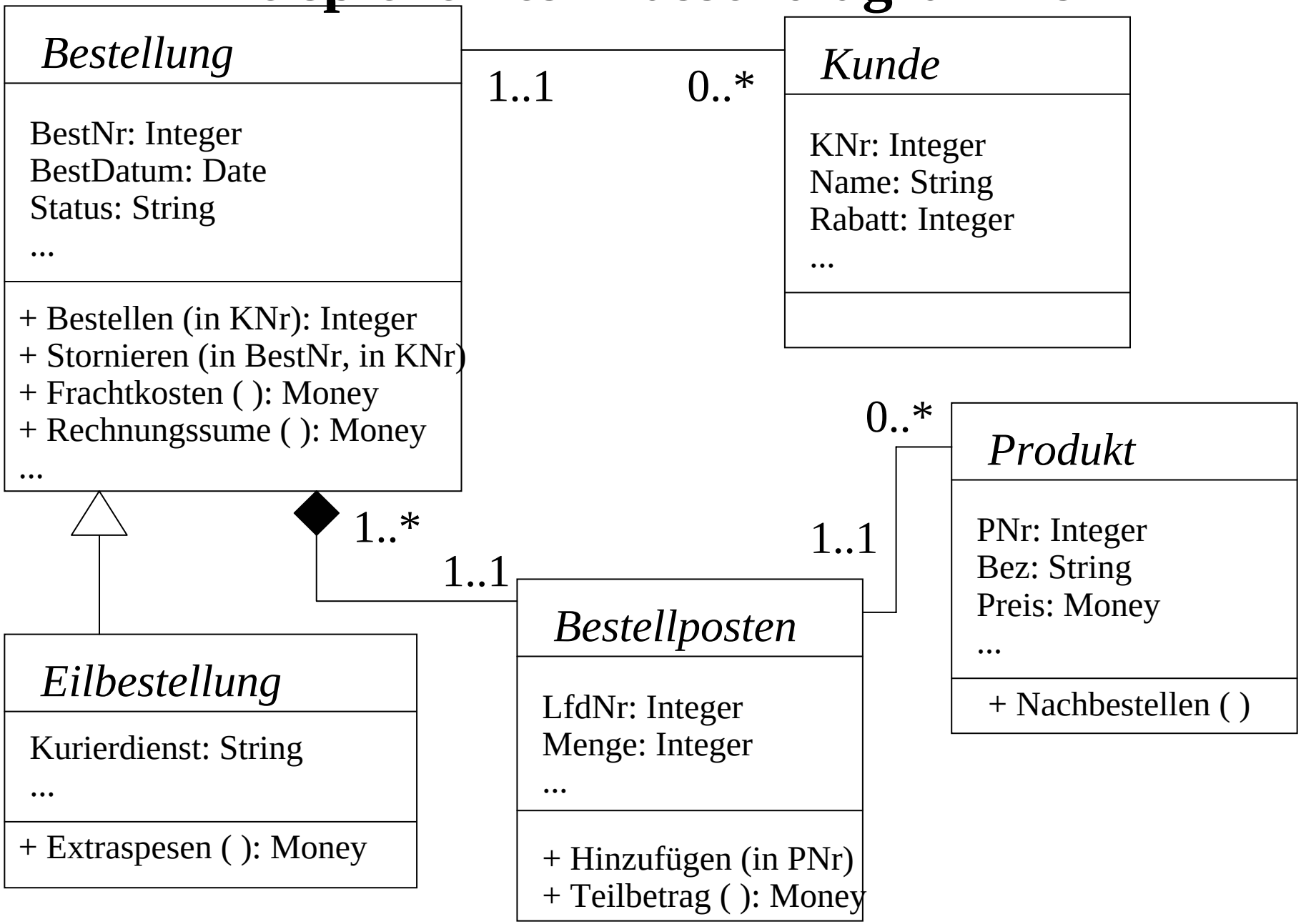
UML (Unified Modeling Language) ist der Industriestandard zur objektorientierten Anforderungsanalyse, Modellierung, Spezifikation und Dokumentation von Daten und Programmen

Gegenüber der Datenmodellierung mit ERM kann man mit UML sowohl statische als auch dynamische Aspekte eines IS beschreiben; insbesondere werden unterstützt:

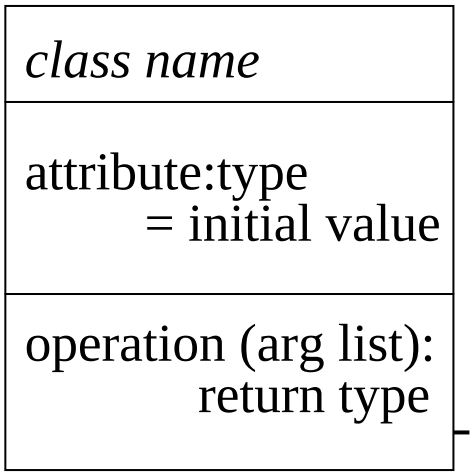
- Objektaggregationen
- Funktionen auf Objekten (Methoden)
- Interaktionen zwischen Objekten und ganze (Geschäfts-) Prozesse ($\rightarrow$ Kapitel 11)

UML zielt auf das ganze Spektrum von der informellen Kommunikation mit Anwendern bis hin zur Programmdokumentation. Dazu wird eine Vielfalt an Diagrammtypen angeboten, deren formale Semantik jedoch nicht immer klar ist.

Beispiel eines Klassendiagramms



Klassen und Objekte



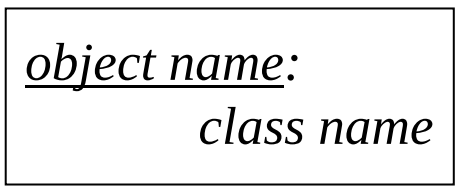
Klasse (Objekttyp, Komponente)

Attribute

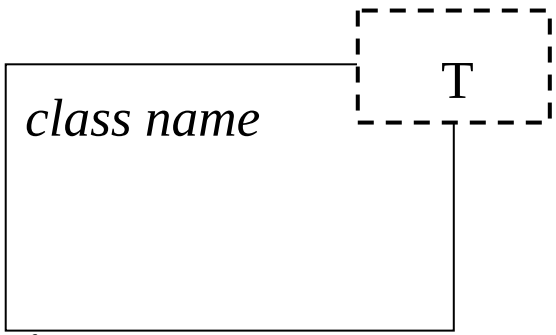
Methoden mit in-, out- und inout-Parametern
+: public, -: private, #: protected

constraints

Konsistenzbedingungen, Vor- und Nachbedingungen (constraints) in Textform

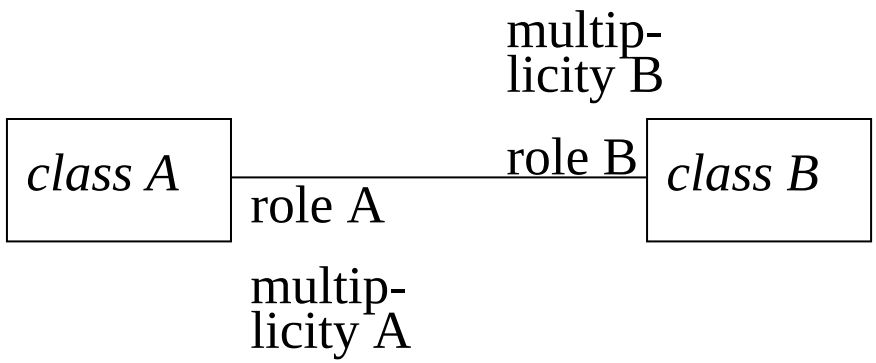


Objekt (Instanz einer Klasse)

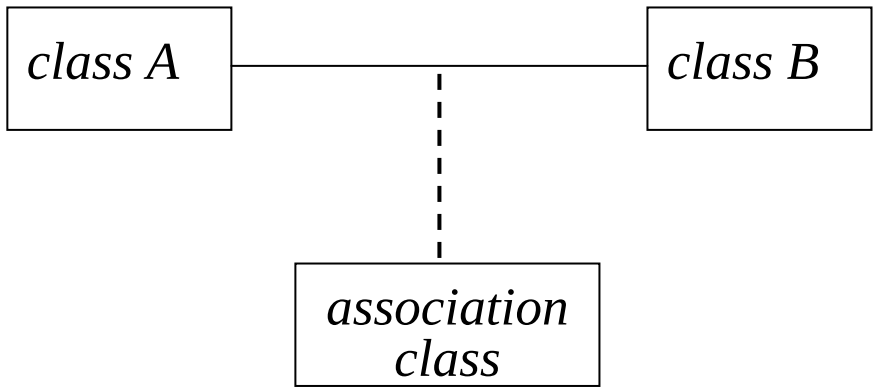


Parametrisierte Klasse (Template Class)
z.B. List <T>

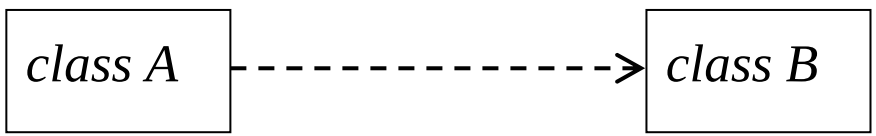
Assoziationen (Relationships)



Assoziation / Relationship (assocations) zwischen Klassen mit Rollennamen und Kardinalitäten (min..max) und ggf. der Option ordered

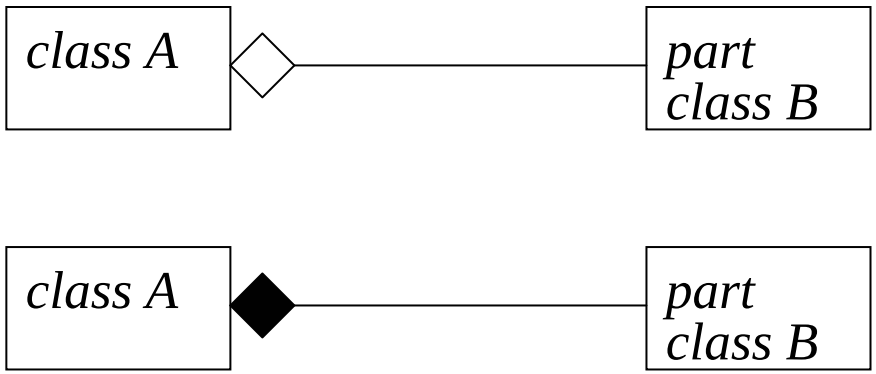


Beziehungsklasse (assocation class): entspricht den ERM-Relationships mit eigenen Attributen



Abhängigkeit (dependeny) der Klasse A von Klasse B (z.B. Server-API-Klasse B und Client-Klasse A)

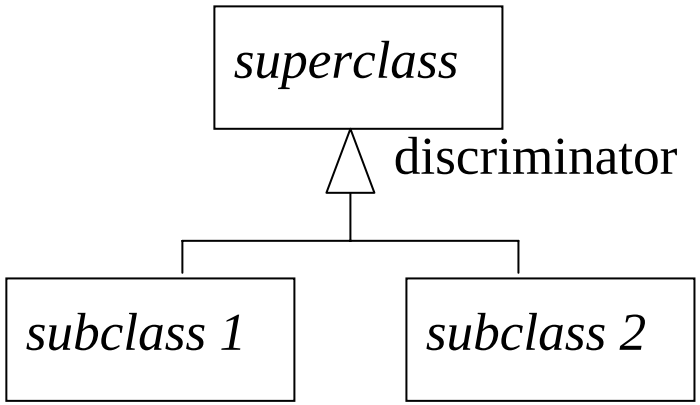
Aggregationen und Kompositionen



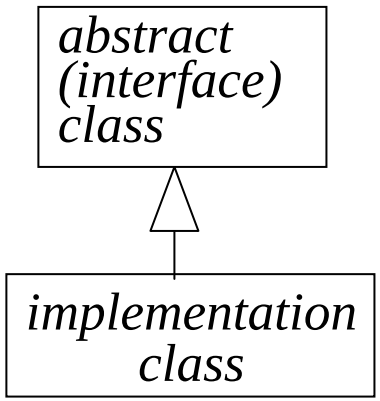
Aggregation (aggregation)
mit shared objects
(Sonderfall der Assoziation)

Komposition (composition)
mit non-shared objects
(Sonderfall der Assoziation)

Generalisierungen



Generalisierungshierarchie
(generalization)

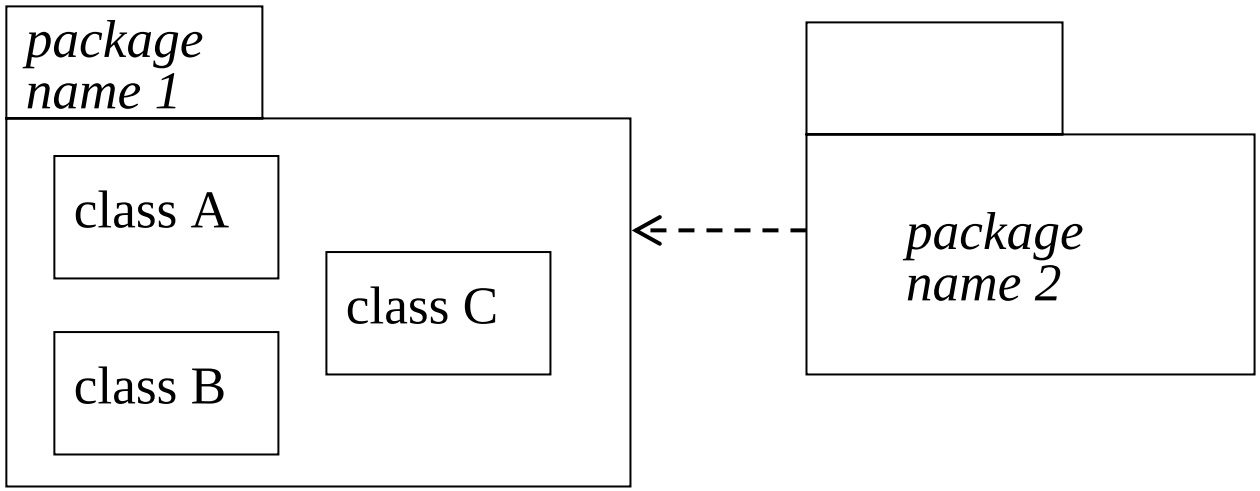


Beziehung zwischen Schnittstelle (ADT
und Implementierung
(Sonderfall der Generalisierung /
Spezialisierung)

Weitere Diagrammtypen in UML

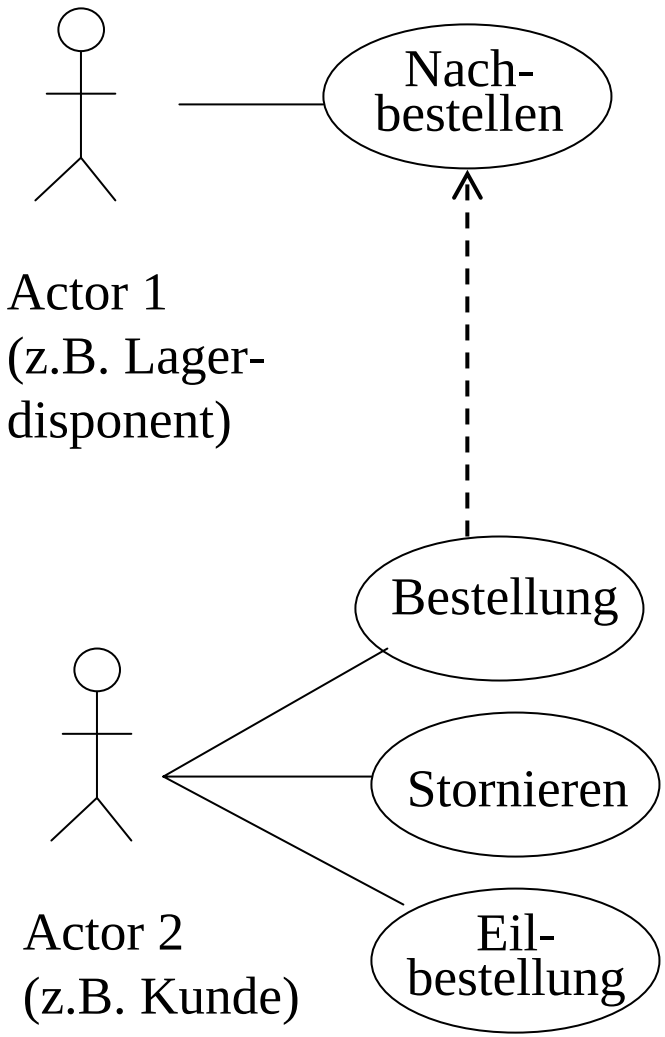
- ***Paket-Diagramme (Package Diagrams)*** als vergröberte Sicht auf große Klassendiagramme,
- ***Anwendungsfall-Diagramme (Use-Case Diagrams)*** zur groben Darstellung der Beziehung zwischen Benutzern und Funktionen,
- ***Interaktionsdiagramme (Sequence Diagrams)*** zur Darstellung der Methodenaufrufe zwischen Objektklassen,
- ***Zustandsübergangsdiagramme (State-Transition Diagrams)*** zur Darstellung des Verhaltens von Objekten in Anwendungsabläufen,
- ***Aktivitätsdiagrame (Activity Diagrams)*** zur Darstellung des Kontroll- und Datenflusses zwischen Objekten.

Paketdiagramme



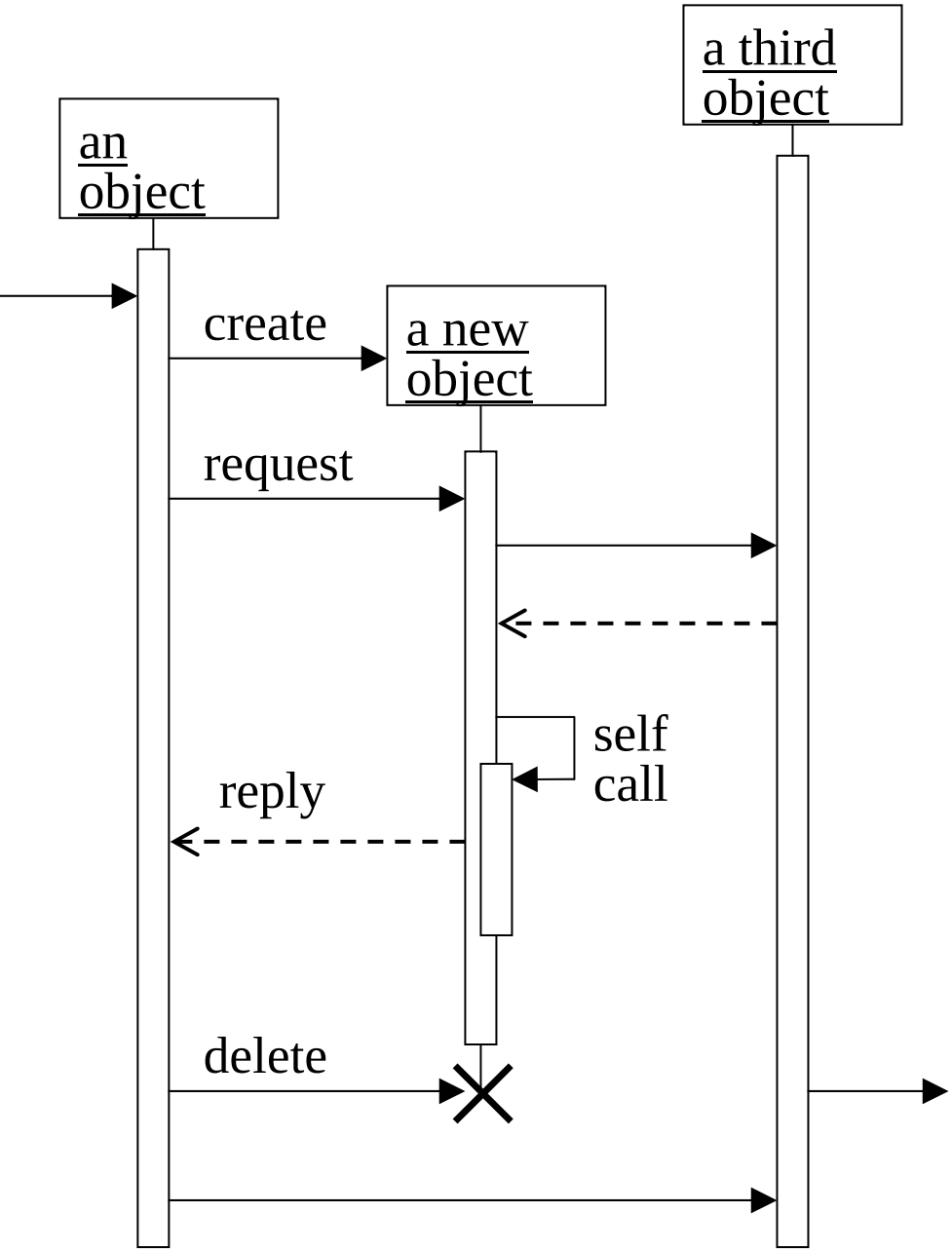
Paketdiagramm
(package diagram)

Anwendungsfalldiagramme



Anwendungsfalldiagramm (use-case diagram): Zusammenhang zwischen Benutzern bzw. Rollen und Funktionen / Ereignissen

Interaktionsdiagramme



Interaktionsdiagramm (sequence diagram): Nachrichtenfluss zwischen Objekten (bzw. Threads)

12.5 Objektorientierte Analyse- und Entwurfsmethodologie

z.B. OMT (Object Modeling Technique):

1) strukturelle Modellierung der Datenobjekte
2) Dynamikmodellierung
3) Funktionsmodellierung

mit iterativen Verfeinerungen

Softwarewerkzeuge (z.B. Rational Rose)
begleiten diesen Entwurfsprozess.