

Objektorientierte Modellierung

Prof. Dr. Roland Dietrich

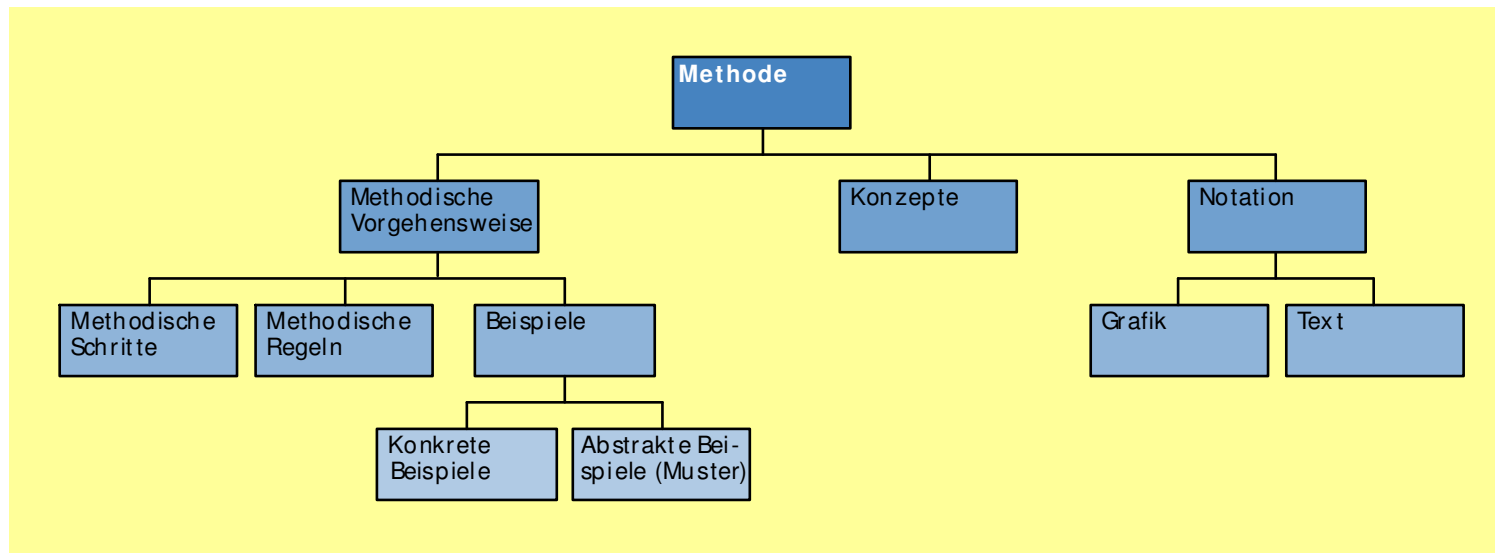
4. Analyseprozess und Analysemuster

Ein objektorientierter Analyseprozess

Ein Katalog von Analysemustern

1. Objektorientierte Softwareentwicklung ✓
2. Anforderungsanalyse mit UML ✓
 - Anwendungsfalldiagramme
3. Statische Modellierung mit UML ✓
 - Klassendiagramme
Objekte und Klassen, Assoziationen, Vererbung
 - Paketdiagramme
4. Der Analyseprozess und Analysemuster
2. Dynamische Modellierung mit UML
 - Interaktionsdiagramme (Sequenz- und Kollaborationsdiagramme)
 - Aktivitätsdiagramme
 - Zustandsautomaten
5. Entwurf mit UML
6. Implementierung in C++

- Die Entwicklung von größeren Softwaresystemen erfordert eine systematische, methodische Vorgehensweise (vgl. Kap.1, S.5)



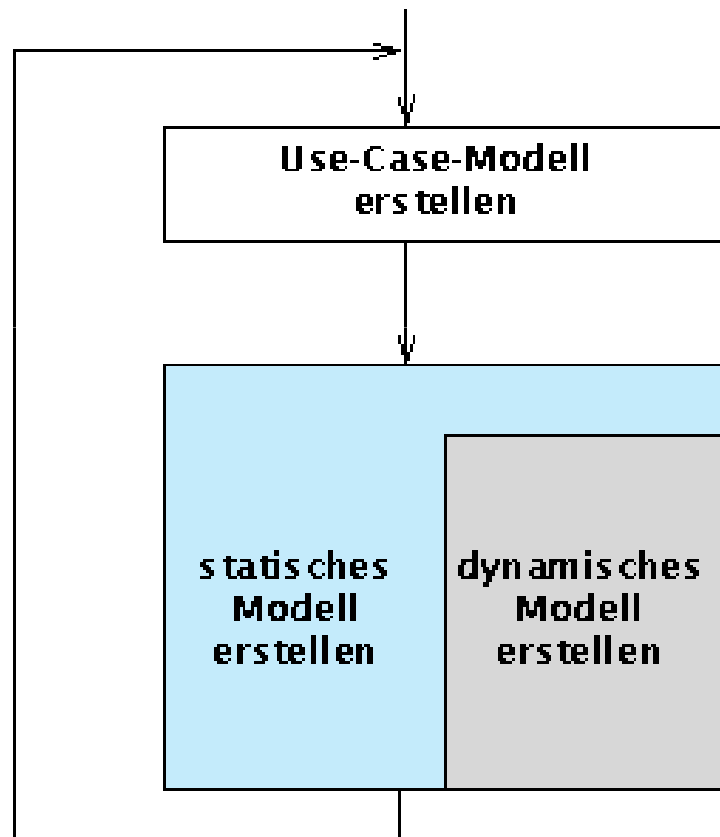
Quelle: [Balzert 96], Abb.13

- Wie formal muss die Vorgehensweise sein?
 - Richtige Vorgehensweise ist Gratwanderung zwischen Formalismus und Formlosigkeit
 - Sehr formelle Vorgehensweisen fordern zu viel Formalismus und Umstände
 - Formlose Vorgehensweisen sind chaotisch und nicht tragbar

- 5 „Schulen“
 - ***anarchists***: Sie ignorieren alle methodischen Vorgehensweisen und verlassen sich nur auf die eigene Kreativität
 - ***behaviorists***: Sie konzentrieren sich auf Rollen und Verantwortlichkeiten
 - ***storyboarder***: Sie sehen die Welt als Menge von Geschäftsprozessen (→ Anwendungsfälle)
 - ***information modeler***: Sie betrachten zunächst nur die Daten; das Verhalten ist sekundär
 - ***architects***: Sie haben ihren Fokus auf *frameworks* und *patterns* gerichtet

- Statisches vs. Dynamisches Modell
 - Betonung des statischen Modells
 - Entstehung eines semantischen Datenmodells in objektorientierter Notation
 - Dynamik des Systems wird außer Acht gelassen
 - Betonung des dynamischen Modells (→ Kap. 5)
 - *use case driven approach*
 - *scenario driven approach*
 - Eine erfolgreiche Modellierung bedingt das Zusammenwirken von statischem und dynamischem Modell

- **Makroprozess**



Quelle: [Balzert 05], Abb.4.1-1

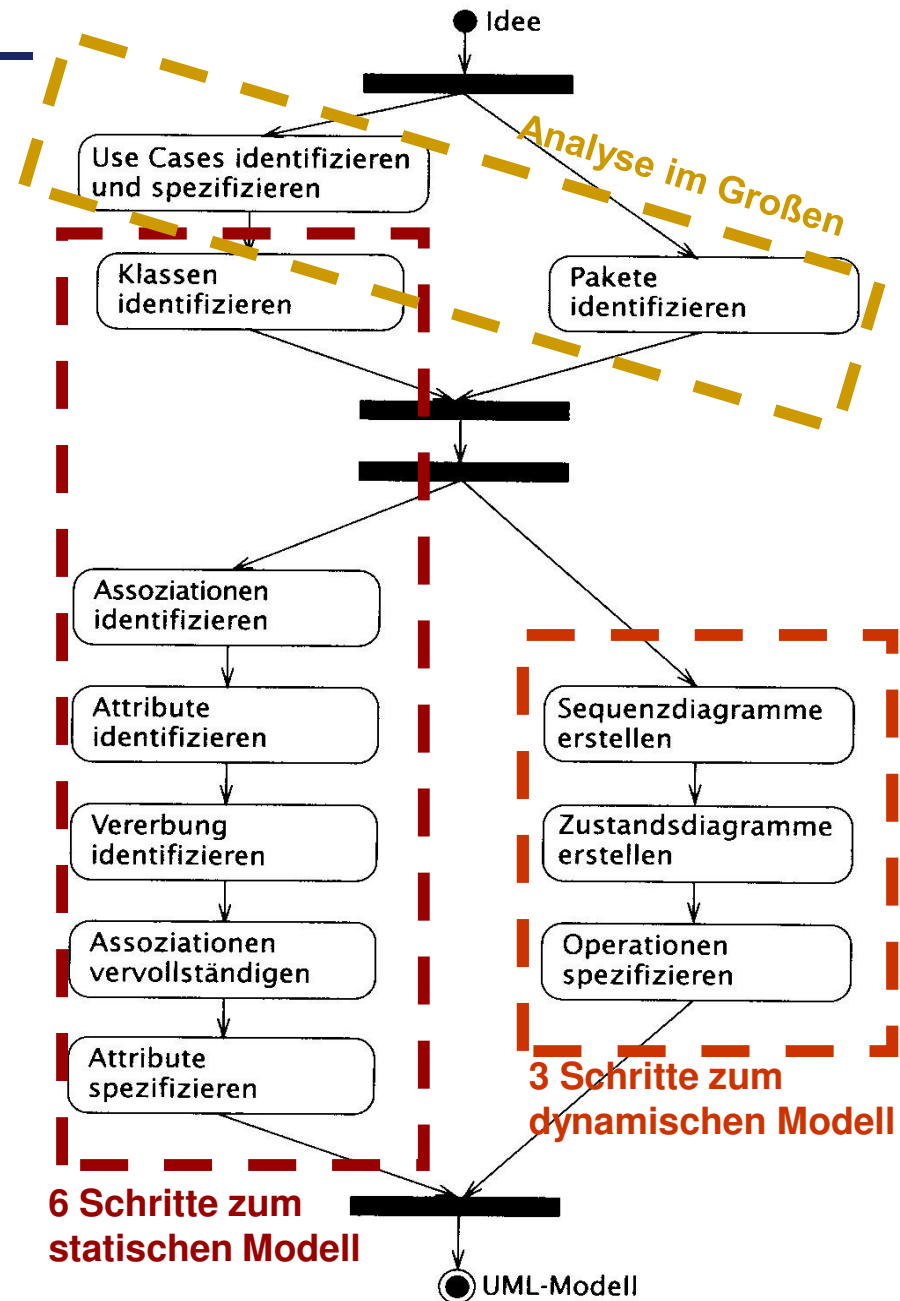
1. Anwendungsfälle ermitteln
2. Klassen ableiten
3. Statisches und dynamisches Modell parallel erstellen

Wechselwirkungen beider Modelle berücksichtigen

- Der Makroprozess berücksichtigt die Gleichgewichtigkeit (*balancing*) von statischem und dynamischem Modell
- Konzentration auf das statische Modell vor dem dynamischen
 - Größere Stabilität des Modells
 - Schaffung einer wesentlichen Abstraktionsebene durch die Bildung von Klassen
- Steht zu Beginn der Modellierung nur das dynamische Modell im Vordergrund, so kann der Analytiker in der großen Menge von Funktionen leicht den Überblick verlieren
- Wichtig: Paralleles Entwickeln beider Modelle und Berücksichtigung von Wechselwirkungen

Analyseprozess

- Der Makroprozess im Detail
 - Analyse im Großen
 - 6 Schritte zum statischen Modell
 - 3 Schritte zum dynamischen Modell
- Analyse im Großen nicht spezifisch für eine objektorientierte Entwicklung
- Statische und dynamische Modellierung besitzen objektorientierten Charakter



- Analyse im Großen

- 1 Anwendungsfall-Modell erstellen

- Übersichtliche Darstellung in **Anwendungsfalldiagrammen**
- **Beschreibung der Anwendungsfälle** mit der Anwendungsfallschablone
- Eventuell Beschreibung der Anwendungsfälle mit anderen Diagrammen
 - **Aktivitätsdiagramm** betont die Abfolge von Verarbeitungsschritten
 - **Zustandsdiagramm** betont die im Laufe der Verarbeitung auftretenden Systemzustände

- 2 Pakete bilden

- Zusammenfassung von Modellelementen zu Paketen
 - In der Analyse insbesondere Klassen und Anwendungsfälle
- Bei großen Systemen durch mehrere Teams muss die Bildung von Paketen am Anfang stehen
- Darstellung der Paket-Struktur als **Paketdiagramm**

- 6 Schritte zum statischen Modell
 - 1 Klassen identifizieren
 - Nur so viele Attribute und Operationen, wie für das Problemverständnis erforderlich
 - **Klassendiagramm**
 - **Kurzbeschreibung Klassen**
 - 2 Assoziationen identifizieren
 - Nur Verbindungen, keine genauen Angaben über Multiplizitäten und Art der Assoziation erforderlich
 - **Klassendiagramm**
 - 3 Attribute identifizieren
 - Alle Attribute des Fachkonzepts
 - **Klassendiagramm**

- 4 Generalisierungsstrukturen identifizieren
 - aufgrund der identifizierten Attribute
 - **Klassendiagramm**
- 5 Assoziationen vervollständigen
 - Endgültige Festlegung auf einfache Assoziation, Aggregation oder Komposition
 - Festlegung der Multiplizität, Rollen, Namen, Einschränkungen
 - **Klassendiagramm**,
 - **Objektdiagramm**
- 6 Attribute spezifizieren
 - Vollständige Spezifikation aller identifizierten Attribute (Typ, Multiplizität, Eigenschaftswerte)
 - **Attributspezifikation**

Anmerkung: Schritte müssen nicht immer streng sequentiell durchlaufen werden.

- 3 Schritte zum dynamischen Modell (vgl. Kap. 4)
 - 1 Szenarien erstellen
 - Jeden Anwendungsfall durch eine Menge von Szenarien beschreiben
 - **Sequenzdiagramm**
 - **Kommunikationsdiagramm**
 - 2 Zustandsautomat erstellen
 - Jede Klasse ist daraufhin zu überprüfen, ob das dynamische Verhalten ihrer Objekte durch ein Zustandsdiagramm präziser spezifiziert werden kann
 - **Zustandsdiagramm**
 - 3 Operationen beschreiben
 - Operationen aus Szenarien und Zustandsdiagrammen in Klassen eintragen
 - **Klassendiagramm**
 - Fachliche **Beschreibung der Operationen**

- Alternative Makroprozesse (1)
 - **Szenariobasierter** Makroprozess
 - Bei umfangreichen funktionalen Anforderungen
 - Keine alten Datenbestände
- 1 Anwendungsfälle formulieren
 - 2 Szenarios aus den Anwendungsfällen ableiten
 - 3 Interaktionsdiagramme aus den Szenarios ableiten
 - 4 Klassendiagramme erstellen
 - 5 Zustandsdiagramme erstellen

- Alternative Makroprozesse (2)
 - **Datenbasierter** Makroprozess
 - Bei umfangreichem Datenmodell
 - Wenn alte Datenbestände existieren
 - Umfang der funktionalen Anforderungen ist zunächst unbekannt
 - Bei Auskunftssystemen mit flexibel gestalteten Anfragen
- 1 Klassendiagramme erstellen
 - 2 Anwendungsfälle formulieren
 - 3 Szenarios aus den Anwendungsfällen ableiten
 - 4 Interaktionsdiagramme aus den Szenarien und dem Klassendiagramm ableiten
 - 5 Zustandsdiagramme erstellen

- Typisch für Objektorientierung: **Evolutionärer Entwicklungsprozess**
 - Erstellung einer objektorientierten Analyse für den Produktkern
 - Entwurf und Implementierung des Kerns
 - Erweiterung des Kerns in weiteren Iterationen, bis ein auslieferbares System entsteht
 - Arbeit früherer Zyklen soll nicht neu gemacht, sondern korrigiert und verbessert werden
- Alternativ: **Inkrementeller Prozess**
 - Analyse für das vollständige Produkt
 - Entwurf und Implementierung evolutionär

- Grundsätze
 - Es gibt keine richtigen oder falschen Modelle, sondern solche, die mehr oder weniger gut ihren Zweck erfüllen
 - Ein gutes Modell ist immer verständlich
 - Die Erstellung verständlicher Modelle erfordert viel Aufwand
 - Das Wissen von kompetenten Fachexperten ist absolut notwendig für ein gutes Modell
 - Modellieren Sie kein System, das zu flexibel ist und zu viele Sonderfälle enthält
 - Prüfen Sie für jeden Sonderfall, ob er es wert ist, die Komplexität des Modells und des Systems zu erhöhen

- Arbeitstechnik
 - Wichtig: Eine schnelle Entwicklung der ersten Version des Modells
 - Zügiger Projektfortschritt
 - Unterstützung der Kommunikation im Team
 - Erste Modelle sind wahrscheinlich weder besonders gut, noch in jedem Fall korrekt sein
 - Perfekte Ideen sind nicht plötzlich da, sie entwickeln sich
 - *DeMarco*
 - *If you wait for a complete and perfect concept to germinate in your mind, you are likely to wait forever.*

- Häufige Fehler
 - 1 Das 100%-Syndrom
 - 2 Zu frühe Qualitätsoptimierung
 - Konzentration zunächst auf das fachliche Konzept
 - Danach Optimierung des fachlich korrekten Modells unter Gesichtspunkten eines optimalen OOA-Modells
 - 3 Bürokratische Auslegung der Methode
 - *follow the spirit, not the letter of a method*
 - 4 Entwurfskriterien in der Analyse berücksichtigen

- Definitionen
 - Ein **Muster** (*pattern*) ist eine Idee, die sich in einem praktischen Kontext als nützlich erwiesen hat und es wahrscheinlich auch in anderen sein wird [Fowler 00]
 - Ein **Analysemuster** ist eine Gruppe von Klassen mit feststehenden Verantwortlichkeiten und Interaktionen [Coad 95]

- 1. Muster: **Liste**

Bestellung

Nr.

Hiermit bestelle ich folgende Artikel

Nr.	Bezeichnung	Einzelpreis	Anzahl	Preis
47	Kugelschreiber	12,00	2	24,00
11	Folienstift	2,50	5	12,50
Summe				<input type="text" value="36,50"/>

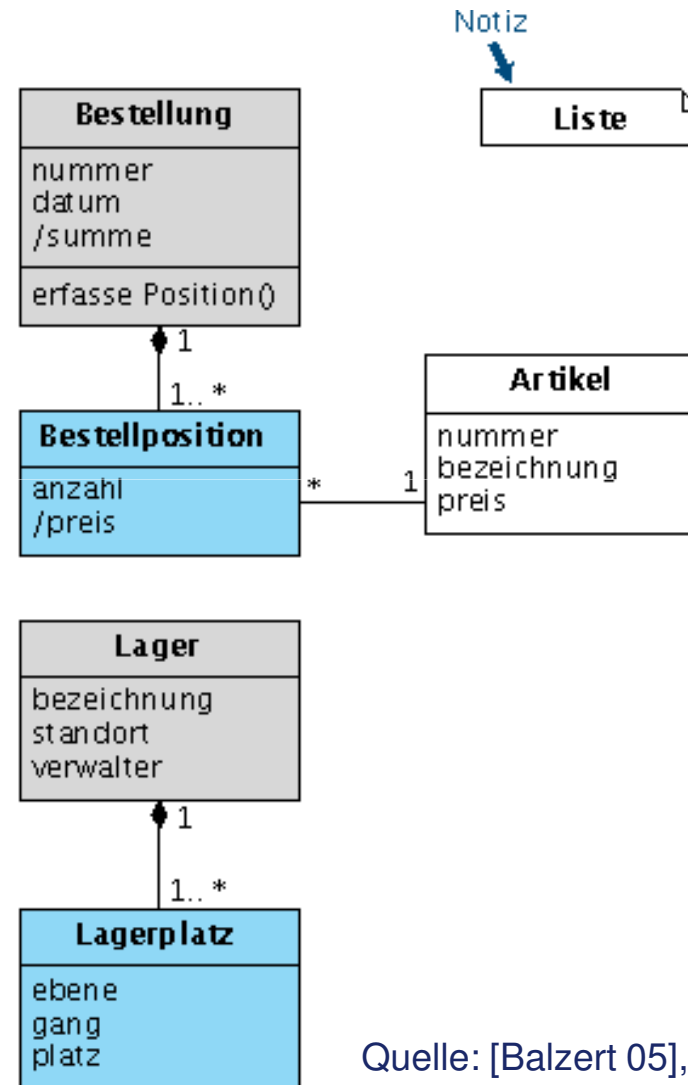
Lager

Bezeichnung

Standort

Verwalter

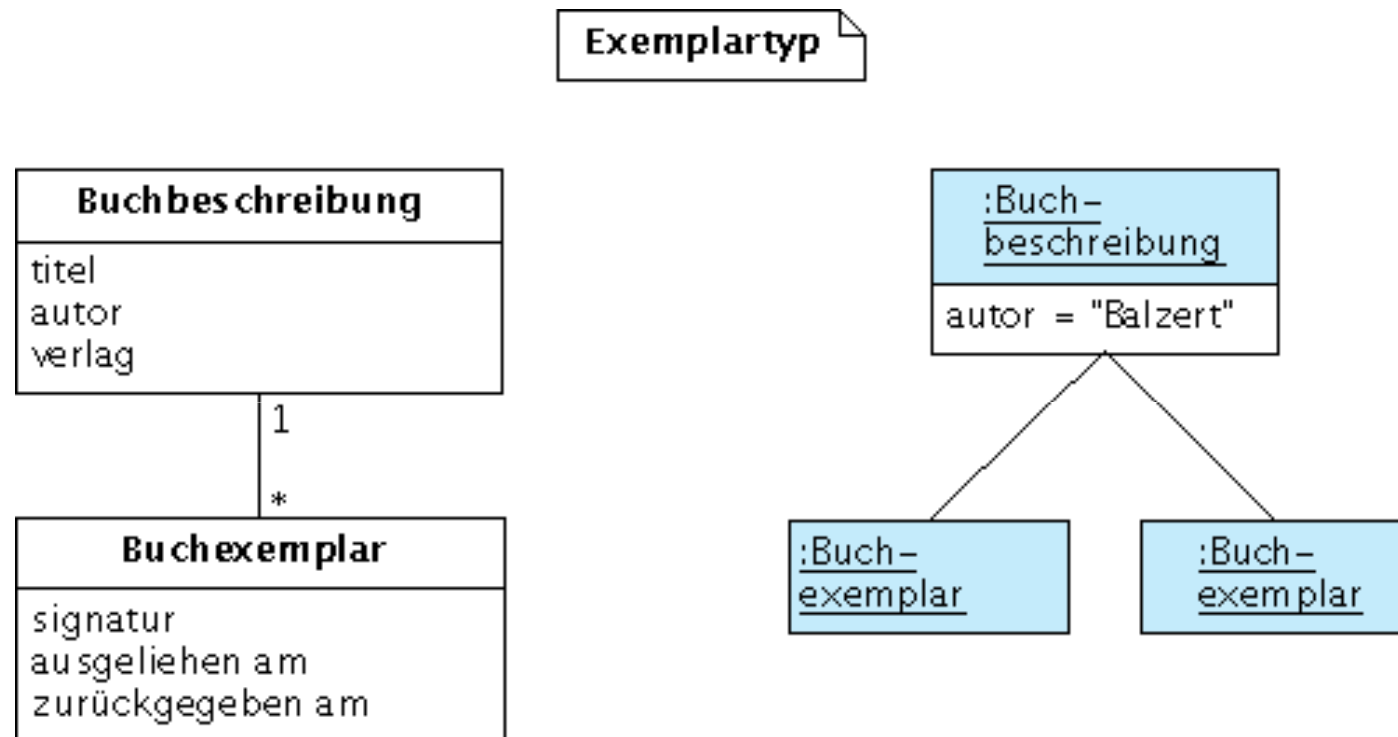
Ebene	Gang	Platz	Belegt mit
1	1	1	XYZ
1	1	2	ABC
1	2	1	



Quelle: [Balzert 05], Abb.3.1-1

- 1. Muster: Liste
 - Komposition
 - Ein Ganzes besteht aus gleichartigen Teilen
 - Teil-Objekte bleiben einem Aggregat-Objekt fest zugeordnet, können jedoch vor dem Aggregat-Objekt gelöscht werden
 - Attributwerte des Aggregat-Objekts gelten auch für die zugehörigen Teil-Objekte
 - Das Aggregat-Objekt enthält i. allg. mindestens ein Teil-Objekt, d.h. die Kardinalität ist meist 1..*

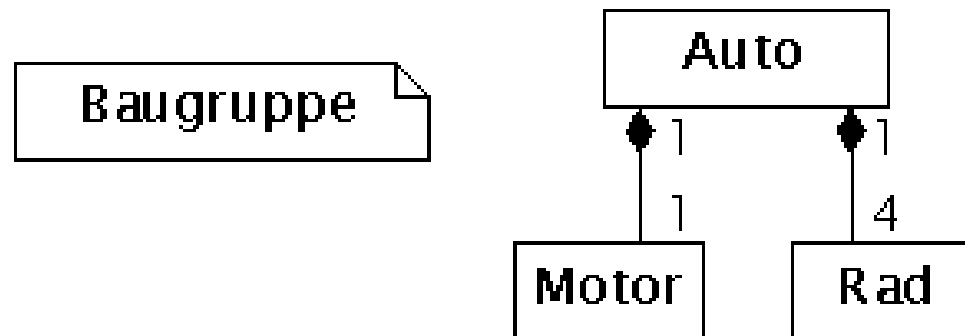
- 2. Muster: **Exemplartyp**



Quelle: [Balzert 05], Abb.3.1-2

- 2. Muster: Exemplartyp
 - Einfache Assoziation
 - Erstellte Objektverbindungen werden nicht verändert. Sie werden nur gelöscht, wenn das betreffende Exemplar gelöscht wird.
 - Name der neuen Klasse enthält oft Begriffe wie Typ, Gruppe, Beschreibung, Spezifikation
 - Eine Beschreibung kann – zeitweise – unabhängig von Exemplaren existieren
 - Beim Verzicht auf die neue Klasse, würde als Nachteil lediglich die redundante „Speicherung“ von Attributwerten auftreten

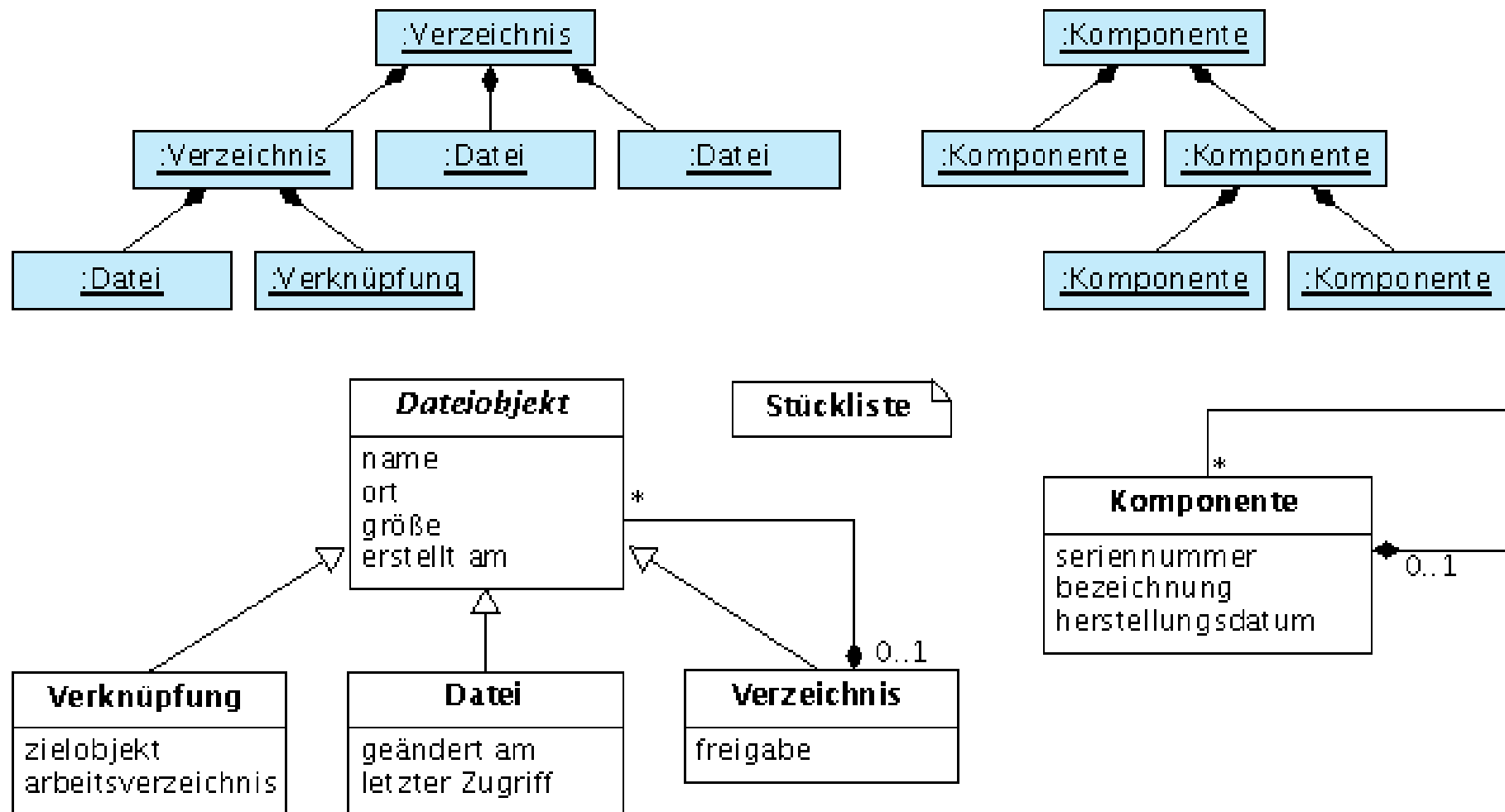
- 3. Muster: **Baugruppe**



Quelle: [Balzert 05], Abb.3.1-3

- 3. Muster: Baugruppe
 - Beschreibung physischer Objekte
 - Eine Kompositionsbeziehung liegt vor
 - Objektverbindungen bestehen meist über längere Zeit
 - Teil-Objekt kann aus seinem Aggregat-Objekt entfernt werden und einem anderen zugeordnet werden.
 - Ein Ganzes kann aus unterschiedlichen Teilen bestehen

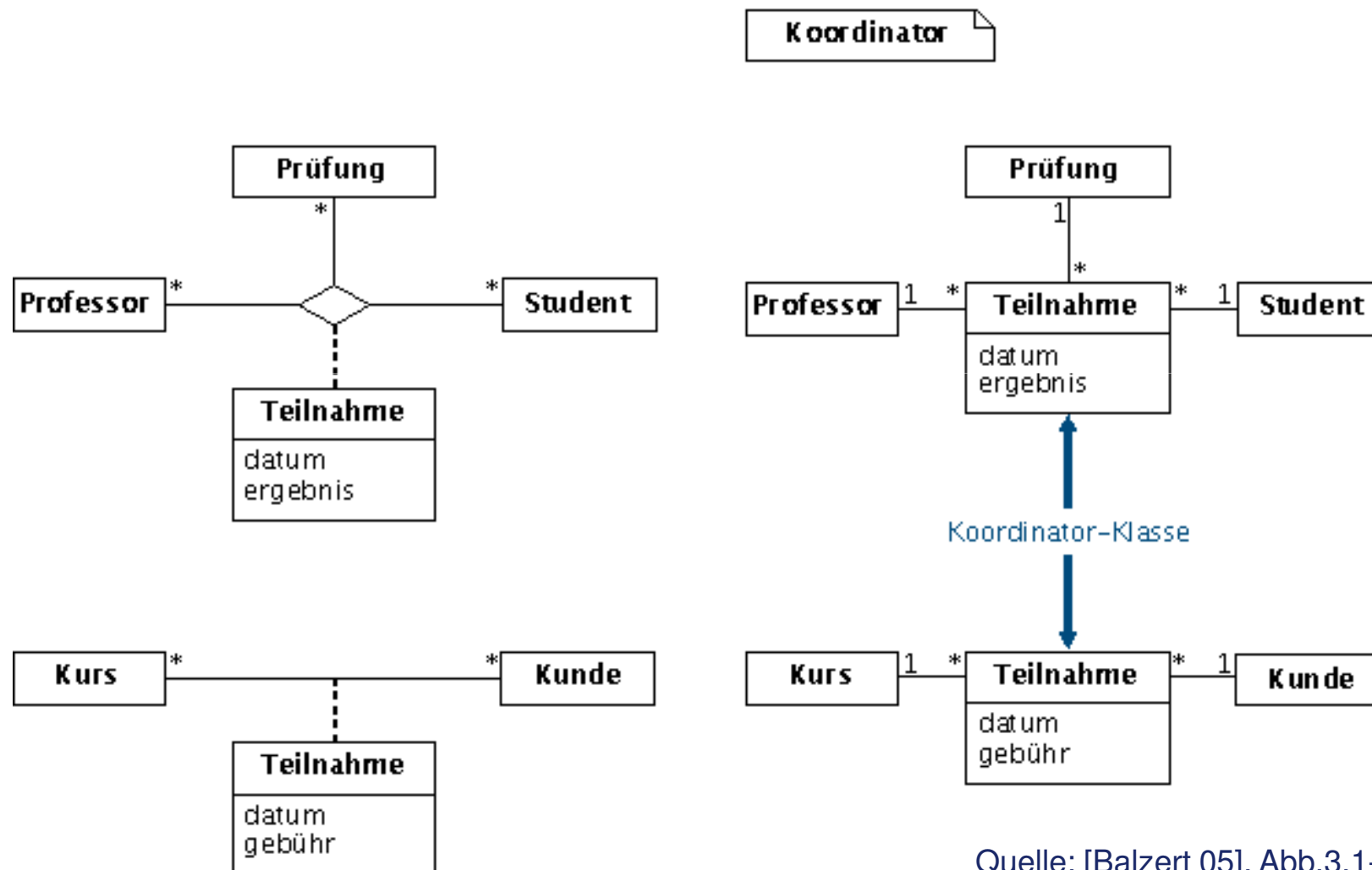
- 4. Muster: **Stückliste**



Quelle: [Balzert 05], Abb.3.1-4

- 4. Muster: Stückliste
 - Es liegt eine hierarchische Kompositionsstruktur vor:
 - Ein Ganzes besteht aus vielen unterschiedlichen Teilen
 - Ein Teil eines Ganzen kann wieder ein Ganzes sein
 - Objekt der Art A kann sich aus mehreren Objekten der Arten A, B und C zusammensetzen
 - Aggregat-Objekt und seine Teile müssen sowohl als Einheit als auch einzeln behandelt werden können
 - Teile können einem anderen Ganzen zugeordnet werden.
 - Kardinalität der Aggregatklasse ist 0..1
 - Sonderfall: Ein Stück besteht nur aus einer einzigen Art

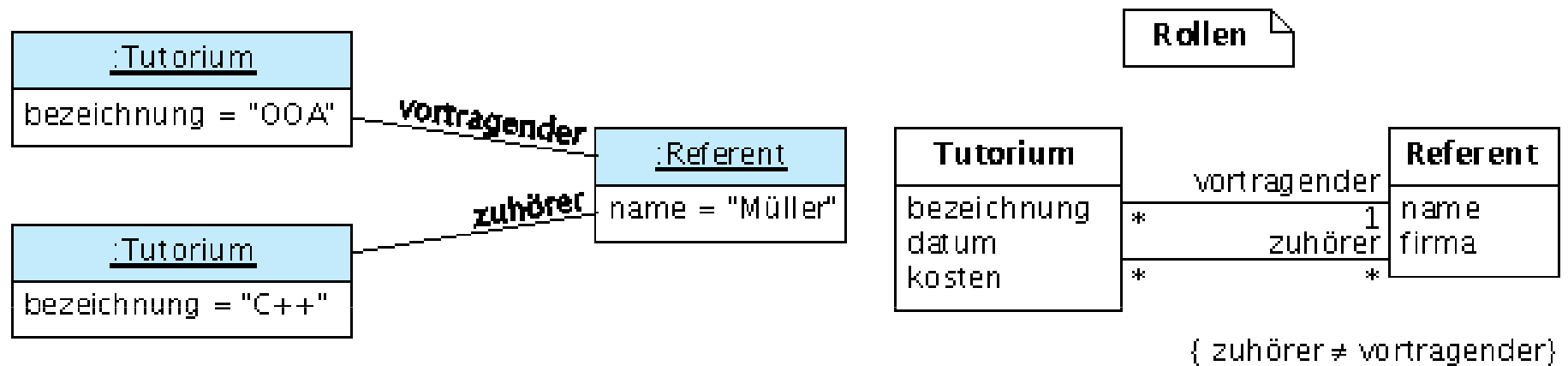
- 5. Muster: **Koordinator**



Quelle: [Balzert 05], Abb.3.1-5

- 5. Muster: Koordinator
 - Einfache Assoziation
 - Koordinator-Klasse ersetzt n-äre ($n \geq 2$) Assoziation mit Assoziationsklasse
 - Koordinator-Klasse besitzt kaum Attribute/Operation jedoch Assoziationen zu mehreren anderen Klassen, im allgemeinen zu genau einem Objekt jeder Klasse

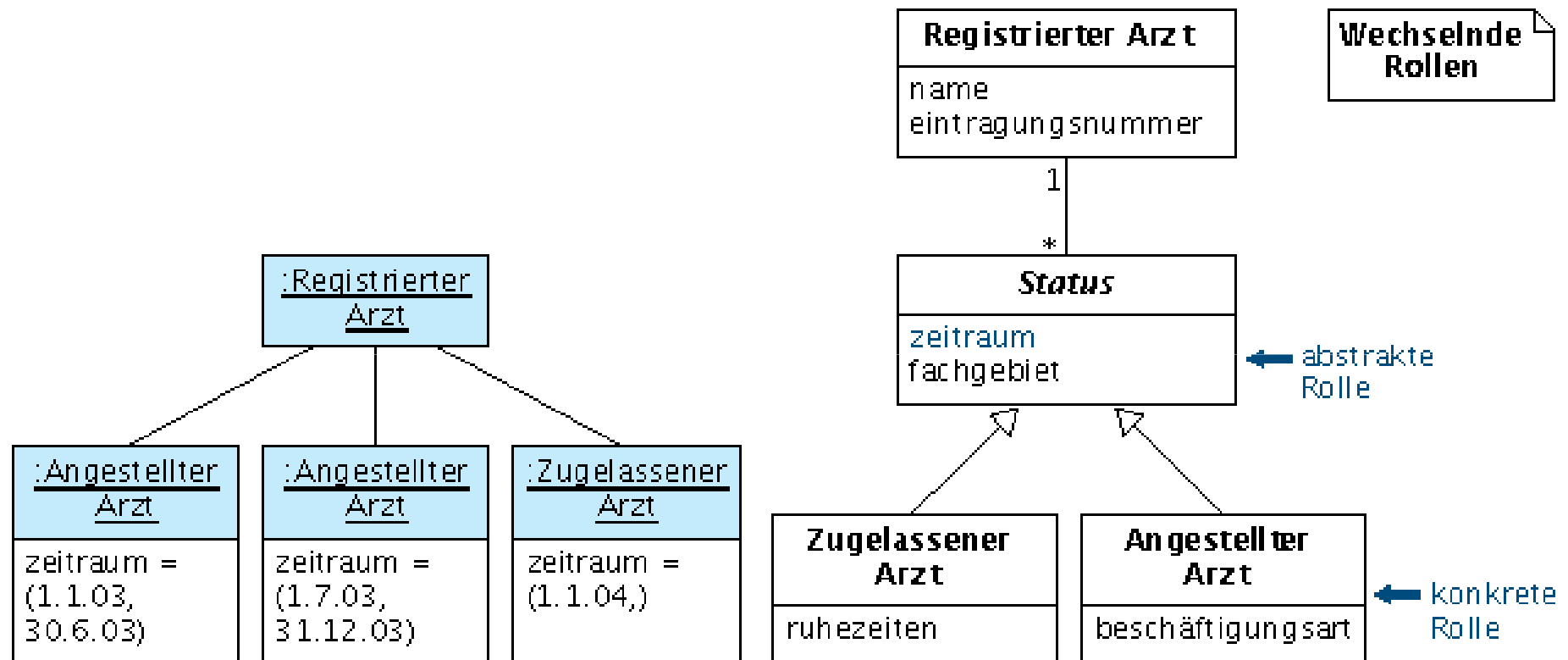
- 6. Muster: **Rollen**



Quelle: [Balzert 05], Abb.3.1-6

- 6. Muster: Rollen
 - Zwischen zwei Klassen existieren zwei oder mehrere einfache Assoziationen
 - Ein Objekt kann – zu einem Zeitpunkt – in Bezug auf Objekte anderer Klassen verschiedene Rollen spielen
 - Objekte, die verschiedene Rollen spielen können, besitzen unabhängig von der der jeweiligen Rolle gleiche
 - Eigenschaften (Attribute, Assoziationen) und
 - Funktionalität (Operationen)

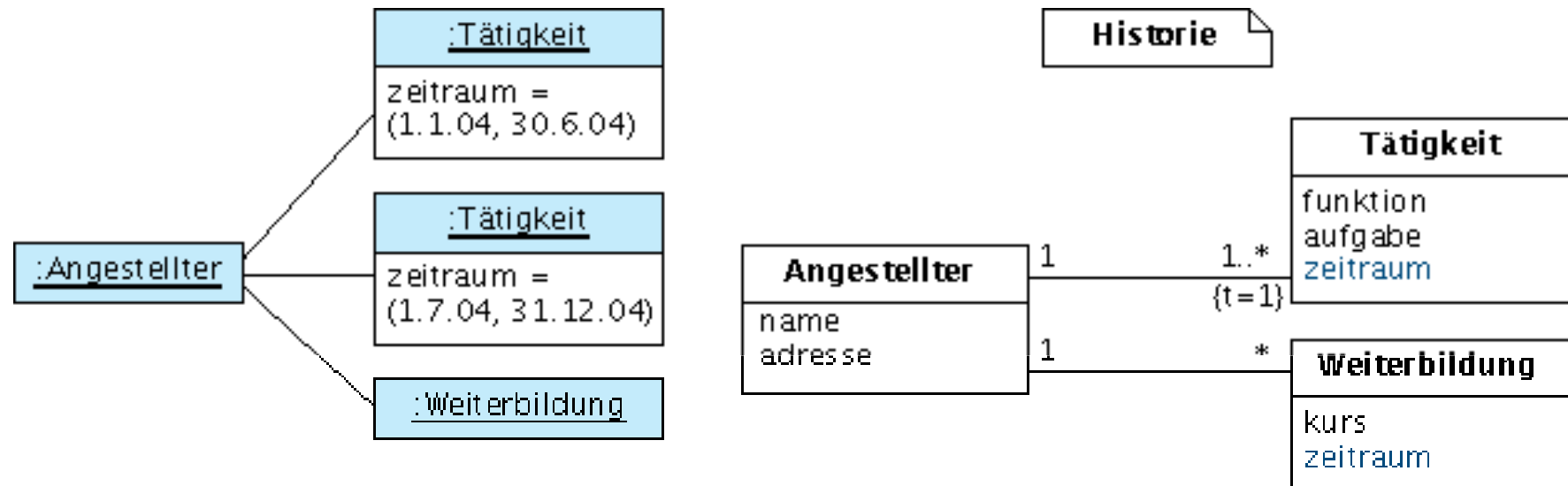
- 7. Muster: **Wechselnde Rollen**



Quelle: [Balzert 05], Abb.3.1-7

- 7. Muster: Wechselnde Rollen
 - Objekt der realen Welt kann – zu verschiedenen Zeiten – verschiedene Rollen spielen
 - In jeder Rolle kann es unterschiedliche Eigenschaften und Operationen besitzen
 - Alle Rollen werden in einer allgemeine Rollen-Klasse generalisiert
 - Die unterschiedlichen Rollen werden mittels Spezialisierung modelliert
 - Objektverbindungen zwischen dem Objekt und seinen Rollen werden nur erweitert, d.h. weder gelöscht, noch zu anderen Objekten aufgebaut

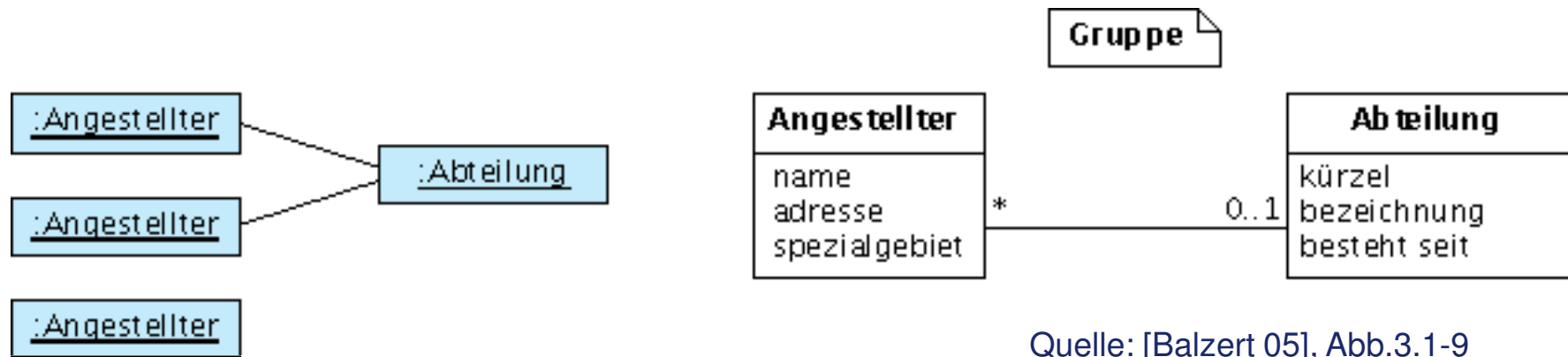
- 8. Muster: **Historie**



Quelle: [Balzert 05], Abb.3.1-8

- 8. Muster: Historie
 - Einfache Assoziation
 - Für ein Objekt sind mehrere Vorgänge bzw. Fakten zu dokumentieren
 - Für jeden Vorgang bzw. jedes Faktum ist der Zeitraum festzuhalten
 - Objektverbindungen zu den Fakten bzw. Vorgängen werden nur erweitert
 - Die zeitliche Restriktion $\{t = k\}$ (k = gültige Multiplizität) sagt aus, was zu einem Zeitpunkt gelten muss

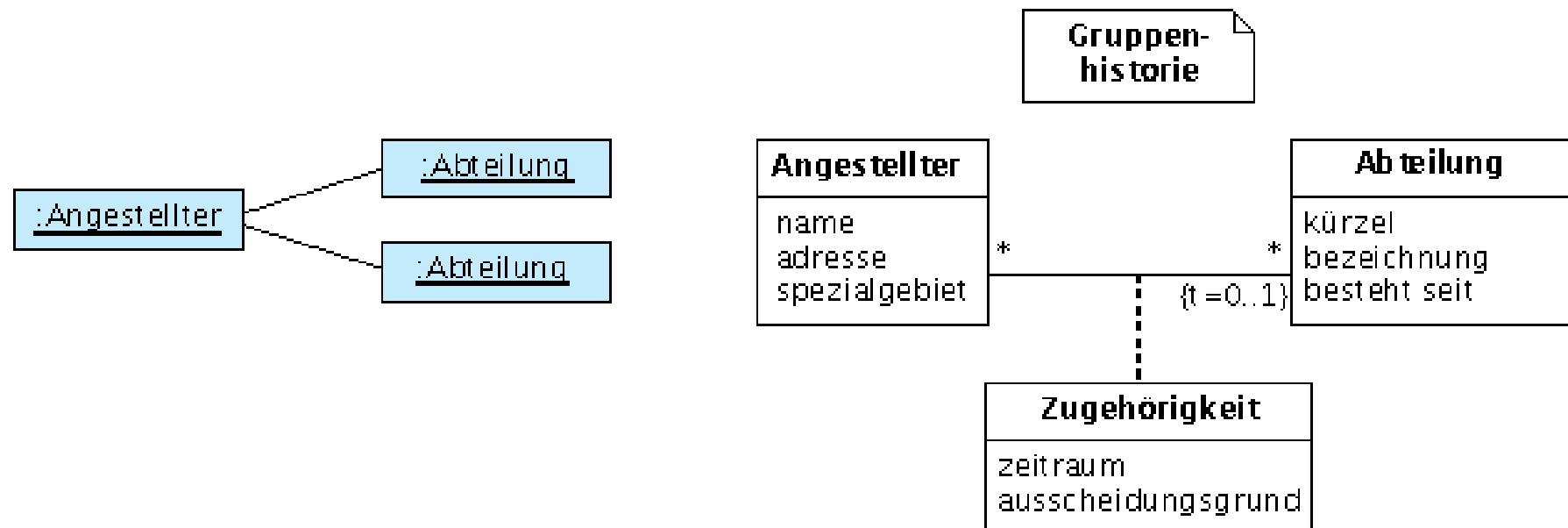
- 9. Muster: **Gruppe**



Quelle: [Balzert 05], Abb.3.1-9

- 9. Muster: Gruppe
 - Einfache Assoziation
 - Mehrere Einzel-Objekte gehören – zu einem Zeitpunkt – zum selben Gruppen-Objekt
 - Gruppen ...
 - können entweder zeitweilig ohne Einzel-Objekte existieren oder
 - müssen immer eine Mindestanzahl von Einzel-Objekten besitzen
 - Objektverbindungen können auf- und abgebaut werden

- 10. Muster: **Gruppenhistorie**



Quelle: [Balzert 05], Abb.3.1-10

- 10. Muster: Gruppenhistorie
 - Ein Einzel-Objekt gehört – im Laufe der Zeit – zu unterschiedlichen Gruppen-Objekten.
 - Die Historie (Zeitraum) wird mittels einer Assoziationsklasse modelliert. Dadurch ist die Zuordnung der Einzel-Objekte zur Gruppe deutlich sichtbar.
 - Die zeitliche Restriktion $\{t = k\}$ (k = gültige Multiplizität) sagt aus, was zu einem Zeitpunkt gelten muß.
 - Objektverbindungen bleiben bestehen, neue Verbindungen werden hinzugefügt.

- Modellieren und Erkennen von Mustern
 - Modellieren Sie folgende Problemstellungen als Klassendiagramme, jeweils unter Verwendung eines geeigneten Analysemodells. Fügen Sie den Klassen auch geeignete Attribute hinzu.
 - a. Mehrere Personen schließen sich zu einer Fahrgemeinschaft zusammen.
 - b. Ein Projektplan (z.B. ein Balkendiagramm) besteht aus mehreren Planungsschritten.
 - c. Ein Mitarbeiter tritt als Programmierer in ein Unternehmen ein. Später wird er als Manager und Geschäftsführer tätig. Für Programmierer, Manager und Geschäftsführer sind unterschiedliche Eigenschaften festzuhalten.
 - d. In einem Sportverein sind Sportler zu verschiedenen Zeiten in unterschiedlichen Mannschaften aktiv.

- e. In einem Grafiksystem bilden Kreise und Rechtecke eine Gruppe. Diese Gruppe kann wiederum Teil einer anderen Gruppe sein.
- f. Zu einem Inventarstück in einem Museum sollen der derzeitige Eigentümer, der Vorbesitzer, der Finder und/oder der Überbringer festgehalten werden, die jeweils die gleichen Eigenschaften besitzen. Eine Person kann beispielsweise sowohl Eigentümer als auch Finder sein.
- g. Bei mehreren Videokassetten in einer Videothek handelt es sich um den gleichen Film.
- h. Für Personen sollen die Wohnsitze der letzten 10 Jahre ermittelt werden können. Zu einem Zeitpunkt muß jede Person mindestens einen und kann höchstens zwei Wohnsitze besitzen.

- Systematisches Identifizieren von Analysemustern beim Erstellen von Klassendiagrammen
 - Modellieren Sie den im Folgenden beschriebenen Sachverhalt als Klassendiagramm und verwenden Sie dabei geeignete Analysemuster
 - Eine Praxis mit mehreren Ärzten soll verwaltet werden. Für jeden Patienten sind Name, Adresse und Geburtsdatum zu speichern. Jeder Arzt vertritt bestimmte Fachgebiete. Der Patient kann mehrere Ärzte dieser Praxis konsultieren. Für jede Behandlung werden das Datum, die Diagnose und die erteilten Verordnungen festgehalten. Jede Verordnung umfaßt die Packungsgröße, das Medikament und ggf. eine Vorschrift für die Anwendung.
 - Mehrere Behandlungen werden gemeinsam abgerechnet. Die Abrechnung enthält das Rechnungsdatum und den Behandlungszeitraum, sowie die einzelnen Abrechnungspositionen. Jede Position besteht aus einer laufenden Nummer, der Leistung, dem Abrechnungssatz und den Kosten.

[Fowler 00] M. Fowler: *Analysis Patterns – Reusable Object Models*, Addison Wesley, 2000

[Coad 95] P. Coad mit D. North, M. Mayfield: *Object Models, Strategies, Patterns, and Applications*, Prentice Hall, 1995