Vorlesung Softwaretechnik
- Objektorientierte Analyse (OOA) -

Prof. Klaus-Peter Fähnrich

Wintersemester 2008/2009

Überblick LE 13

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

LE 13: Objektorientierte Analyse

- Einführung
- OOA-Muster
- Methodische Vorgehensweise
- Zusammenfassung

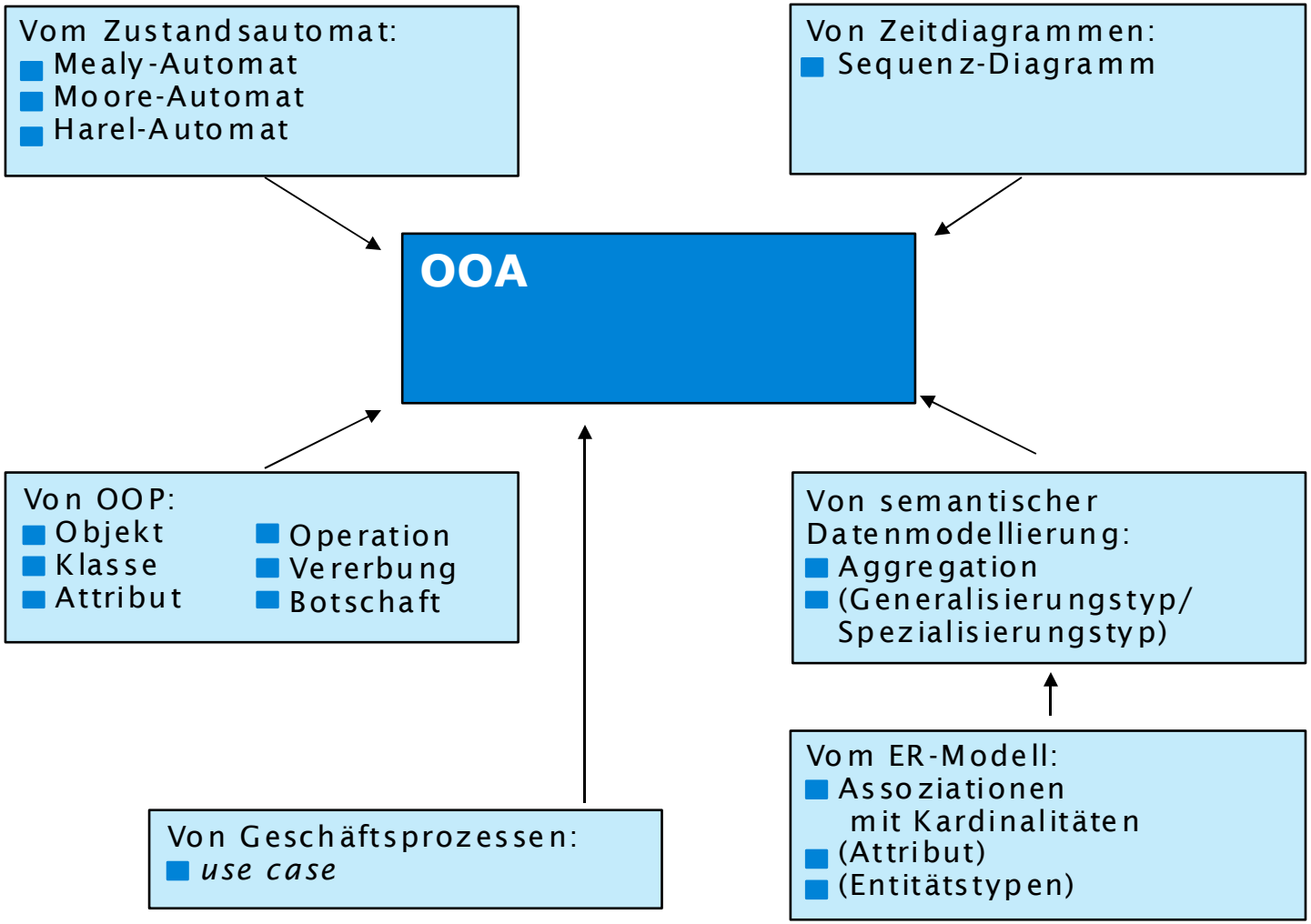
UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Lernziele

1. Wichtige Muster der Systemanalyse
2. Methodische Schritte zur Erstellung eines OOA-Modells
3. Die verschiedenen OOA-Muster und OOA-Modellen
4. Erstellung und Überprüfung eines OOA-Modells

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

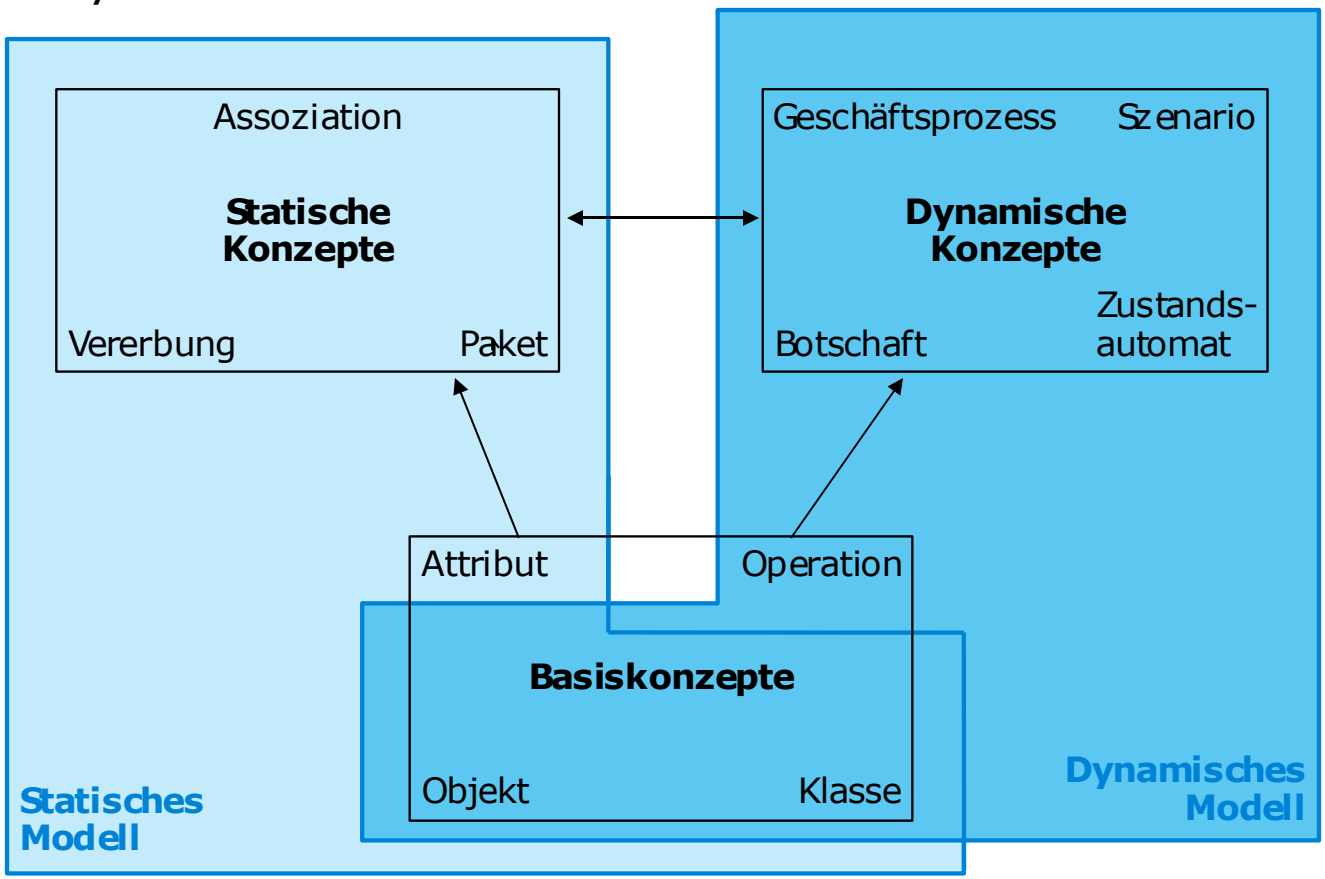
Entstehung der OOA



UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Statistisches und dynamisches Modell

- Ziel der OOA ist es, die fachliche Lösung eines neuen Software-Produkts mit Hilfe objektorientierter Konzepte zu modellieren.
- Die entstehende Spezifikation besteht aus einem statischen und einem dynamischen Modell.



OOA-Muster

- Die Analyse vorhandener OOA-Modelle zeigt, dass sich bestimmte Grundmuster in ähnlicher Form immer wiederholen.
- Vereinfacht: Muster (Pattern) ist eine Idee, die sich in der Praxis bewährt hat.
- OOA-Muster besteht aus einer Gruppe von Klassen mit feststehenden Verantwortlichkeiten und Interaktionen [Coad 95]
  - Gruppe von Klassen verknüpft durch Beziehungen oder
  - Gruppe von kommunizierenden Objekten
- Muster ermöglichen effektive Kommunikation und erlauben standardisierte Lösung bestimmter Probleme.
- Einsatz vorhandener Muster hängt stark vom Verwendungsbereich ab. Daher Unterscheidung allgemeine Muster und anwendungs-spezifische Muster.
- Im folgenden werden allgemeine Muster vorgestellt.

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Muster 1: Liste

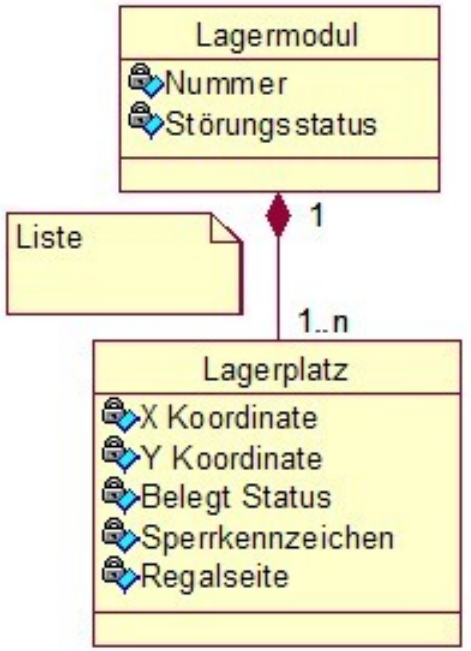
- Das Beispiel Lagerverwaltung zeigt ein Lagermodul bestehend aus verschiedenen Lagerplätzen. Das Muster zeigt eine Lagerliste und die zugehörige OOA-Modellierung.

Lagerliste

Nummer: 4

Störungsstatus: ok

X Koordinate	Y Koordinate	Belegt Status	Sperrkenn-zeichen	Regal-seite
50	100	Frei	kein	A
50	200	Belegt	kein	A
50	300	Frei	kein	A

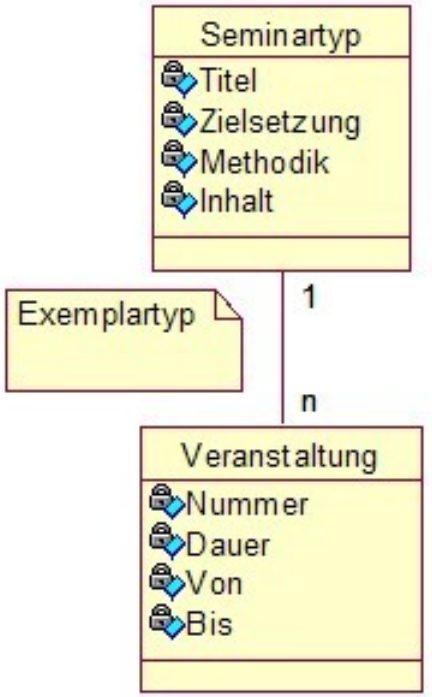


- Eigenschaften:
  - Es liegt eine Komposition vor.
  - Ein Ganzes besteht aus gleichartigen Teilen, d. h. es gibt nur eine Teil-Klasse.
  - Teil-Objekte bleiben einem Aggregat-Objekt fest zugeordnet.
  - Sie können jedoch gelöscht werden, bevor das Ganze gelöscht wird.
  - Die Attributwerte des Aggregat-Objekts gelten auch für die zugehörigen Teil-Objekte.
  - Das Aggregat-Objekt enthält mindestens ein Teil-Objekt, d. h. die Kardinalität ist meist 1..*.

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Muster 2: Exemplartyp

- In der Fallstudie Seminarorganisation sind von einem Seminartyp mehrere Veranstaltungen zu verwalten.

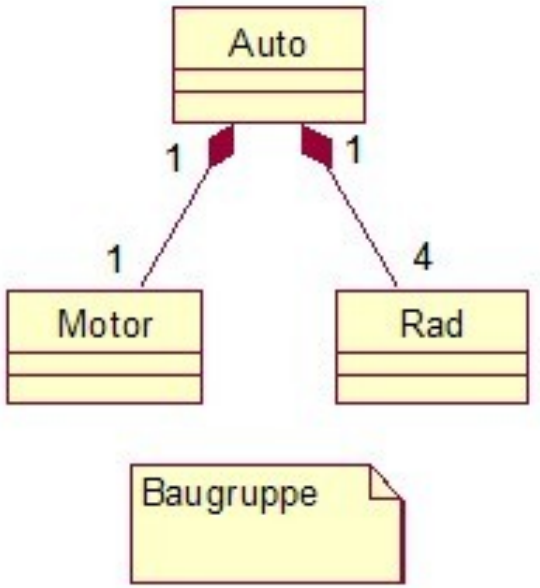


- Eigenschaften:
  - Es liegt eine einfache Assoziation vor, denn es besteht keine Ist-Teil-von-Beziehung.
  - Einmal erstellte Objektverbindungen werden i. A. nicht verändert. Sie werden nur gelöscht, wenn das betreffende Exemplar entfernt wird.
  - Der Name der neuen Klasse enthält oft Begriffe wie Typ, Gruppe, Beschreibung, Spezifikation.
  - Eine Beschreibung kann – zeitweise – unabhängig von konkreten Exemplaren existieren. Daher ist die Kardinalität i. A. *many (0..m)*.
  - Würde auf die neue Klasse verzichtet, so würde als Nachteil lediglich die redundante »Speicherung« von Attributwerten auftreten.

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Muster 3: Baugruppe

- Das Beispiel soll zeigen, dass jedes Auto exakt einen Motor und vier Räder hat.

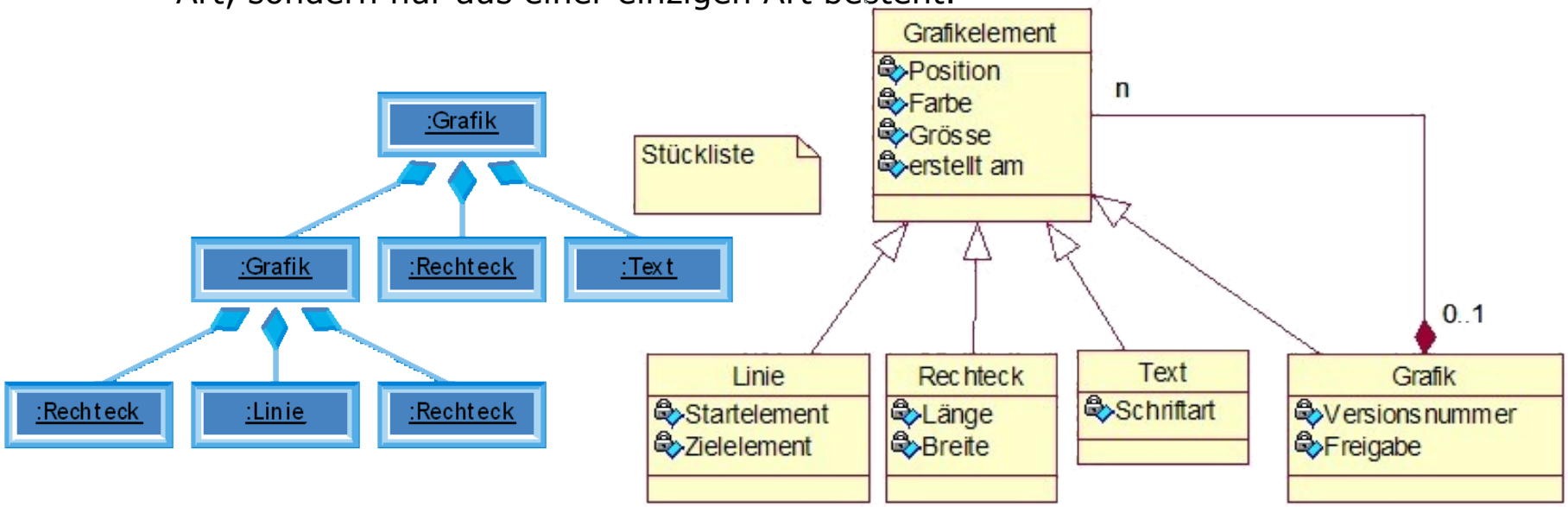


- Eigenschaften:
  - Es handelt sich um physische Objekte.
  - Es liegt eine Komposition vor.
  - Objektverbindungen bestehen über eine längere Zeit.
  - Ein Teil-Objekt kann von seinem Aggregat-Objekt getrennt werden und einem anderen Ganzen zugeordnet werden.
  - Ein Ganzes kann aus unterschiedlichen Teilen bestehen.

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Muster 4: Stückliste

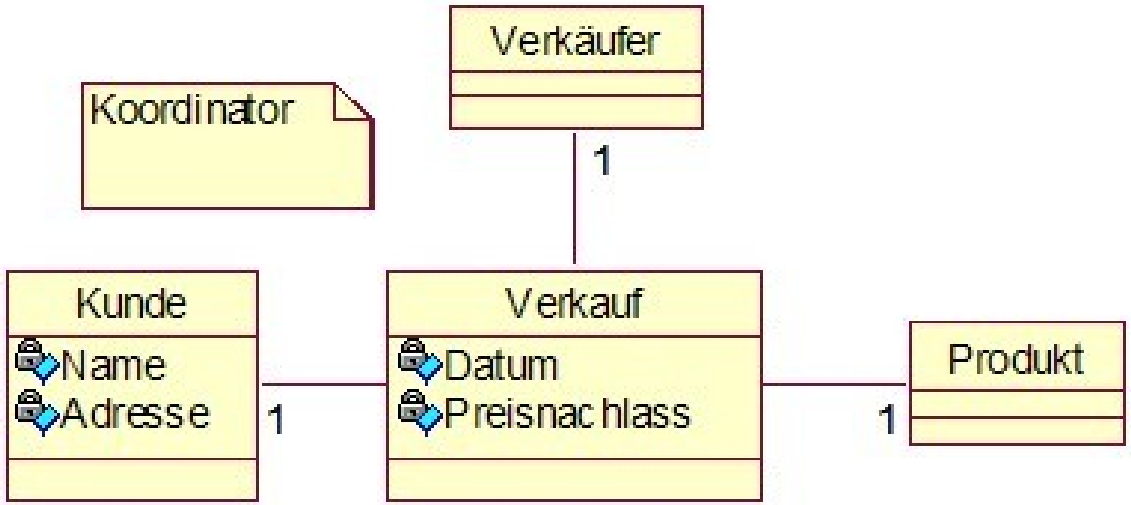
- Eine Grafik besteht aus Grafikelementen. Sie kann andere Grafiken enthalten.
- Eigenschaften
  - Es liegt eine Komposition vor.
  - Das Aggregat-Objekt und seine Teil-Objekte müssen sowohl als Einheit als auch einzeln behandelt werden können.
  - Teil-Objekte können anderen Aggregat-Objekten zugeordnet werden.
  - Die Kardinalität bei der Aggregat-Klasse ist 0..1.
  - Ein Objekt der Art A kann sich aus mehreren Objekten der Arten A, B und C zusammensetzen.
  - Der Sonderfall der Stückliste ist, dass ein Stück nicht aus Objekten unterschiedlicher Art, sondern nur aus einer einzigen Art besteht.



UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Muster 5: Koordinator

- Eine ternäre Assoziation verbindet Objekte der Klasse Kunde, Verkäufer und Produkt und »merkt« sich in der assoziativen Klasse *Verkauf*, welcher Verkäufer welchem Kunden welches Produkt verkauft hat.

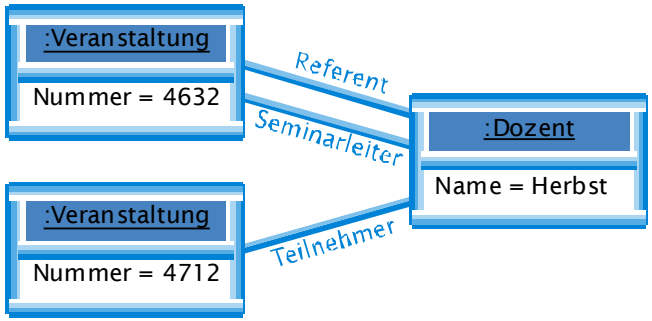


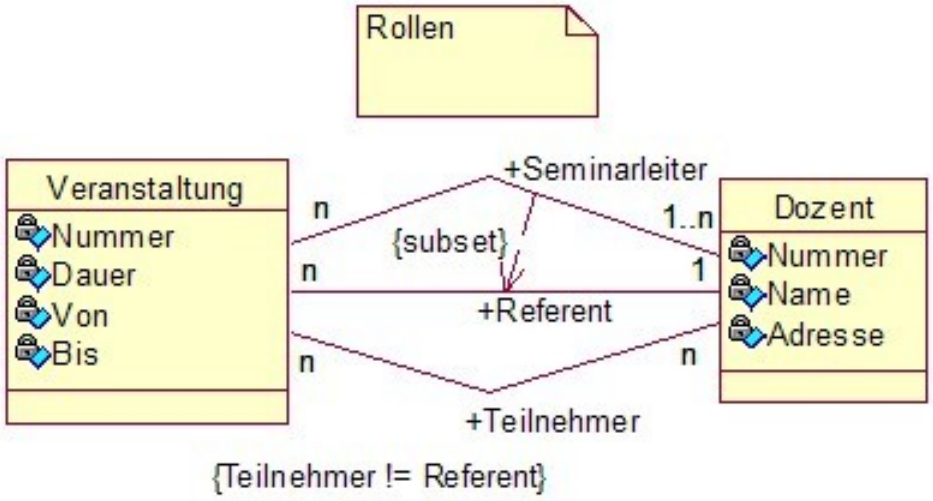
- Eigenschaften
  - Es liegen einfache Assoziationen vor.
  - Die Koordinator-Klasse ersetzt eine n-äre (n ≥2) Assoziation durch binäre Assoziationen mit assoziativer Klasse.
  - Die Koordinator-Klasse besitzt kaum Attribute/Operationen, sondern mehrere Assoziationen zu anderen Klassen, i. A. zu genau einem Objekt jeder Klasse.

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Muster 6: Rollen

- In der Fallstudie Seminarorganisation kann ein Dozent auch Teilnehmer einer Veranstaltung sein, um sich selbst weiterzubilden.
- Der der Dozent kann sowohl Referent, als auch Seminarleiter als auch Teilnehmer von Veranstaltungen sein.
- Dozent spielt – zur selben Zeit – in Bezug auf die Klasse Veranstaltung mehrere Rollen.



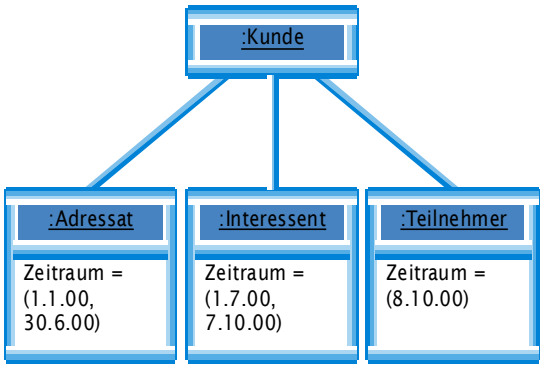


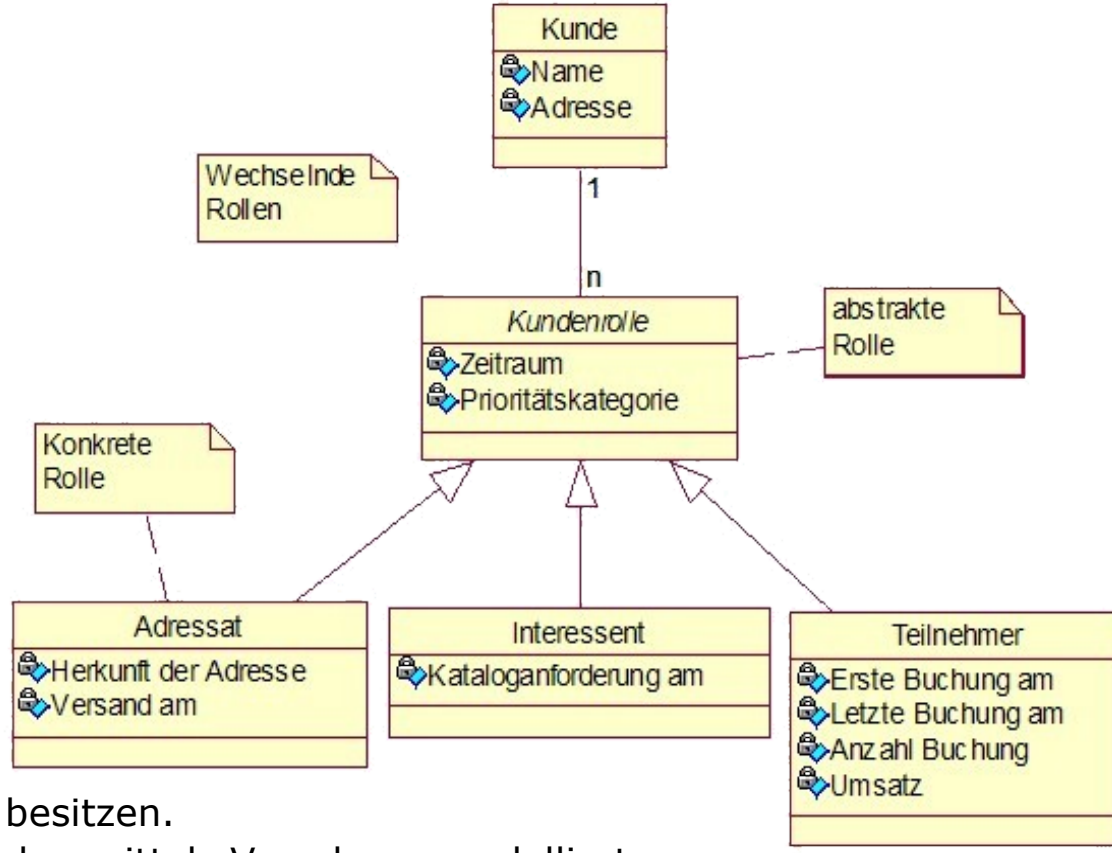
- Eigenschaften
  - Zwischen 2 Klassen existieren zwei oder mehrere einfache Assoziationen.
  - Ein Objekt kann – zu einem Zeitpunkt – in Bezug auf die Objekte der anderen Klasse verschiedene Rollen spielen.
  - Objekte, die verschiedene Rollen spielen können, besitzen unabhängig von der jeweiligen Rolle die gleichen Eigenschaften und ggf. gleiche Operationen.

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Muster 7: Wechselnde Rollen

- Ein Kunde kann wechselnde Rollen in der Seminarorganisation einnehmen. Soll später nachvollzogen werden, welche Werte ein Objekt zu welchem Zeitpunkt besessen hat, dann ist eine Aufzeichnung der Historie notwendig.

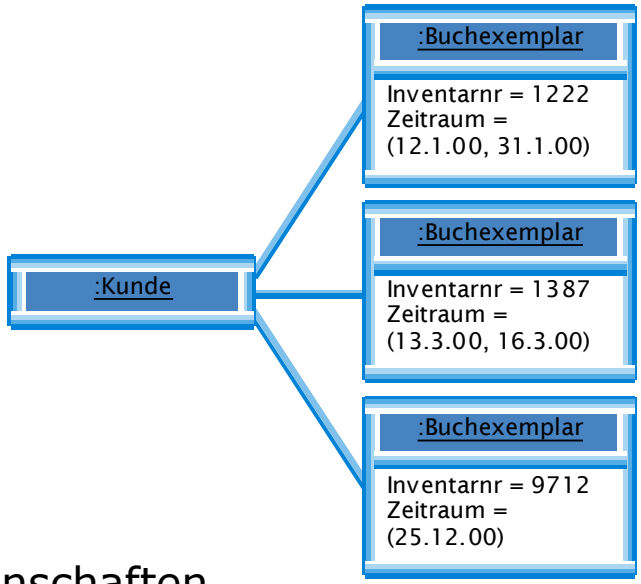


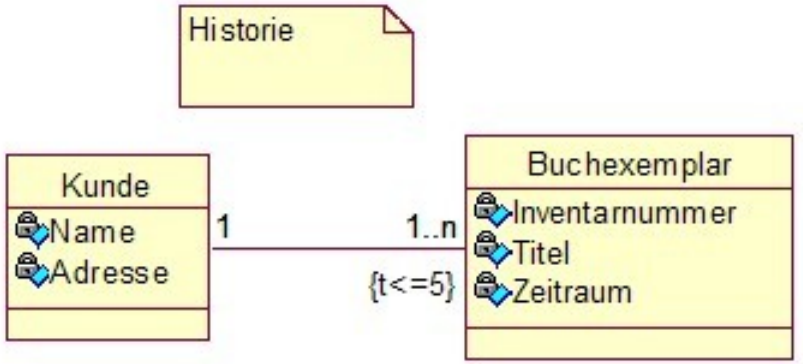


- Eigenschaften
  - Ein Objekt der realen Welt kann zu verschiedenen Zeiten verschiedene Rollen spielen. In jeder Rolle kann es unterschiedliche Eigenschaften (Attribute, Assoziationen) und Operationen besitzen.
  - Die unterschiedlichen Rollen werden mittels Vererbung modelliert.
  - Objektverbindungen zwischen dem Objekt und seinen Rollen werden nur erweitert, d. h. weder gelöscht noch zu anderen Objekten aufgebaut.

Muster 8: Historie

- Eine Bibliothek muss speichern, welche Bücher welcher Kunde ausgeliehen hat. Dabei darf zu jedem Zeitpunkt t der Kunde nur fünf Buchexemplare ausgeliehen haben.



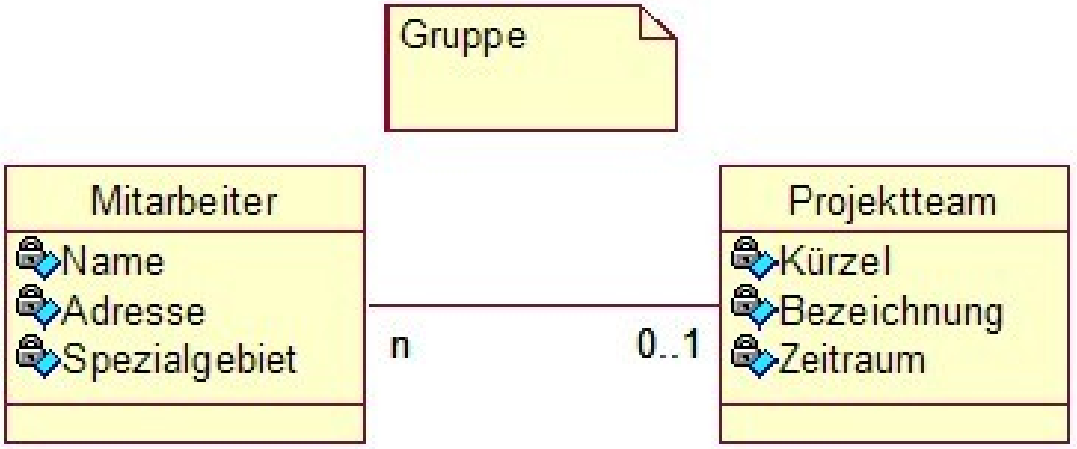


- Eigenschaften
  - Es liegt eine einfache Assoziation vor.
  - Für ein Objekt sind mehrere Vorgänge bzw. Fakten zu dokumentieren.
  - Für jeden Vorgang bzw. jedes Faktum ist der Zeitraum festzuhalten.
  - Aufgebaute Objektverbindungen zu den Vorgängen bzw. Fakten werden nur erweitert.
  - Die zeitliche Restriktion {t#k} (k = gültige Kardinalität) sagt aus, was zu einem Zeitpunkt gelten muss, wobei # für die Vergleichsoperationen =, <, >, ≤, ≥ und ≠ steht.

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Muster 9: Gruppe

- Eine Gruppe bildet sich, wenn mehrere Mitarbeiter zu einem Projektteam gehören.

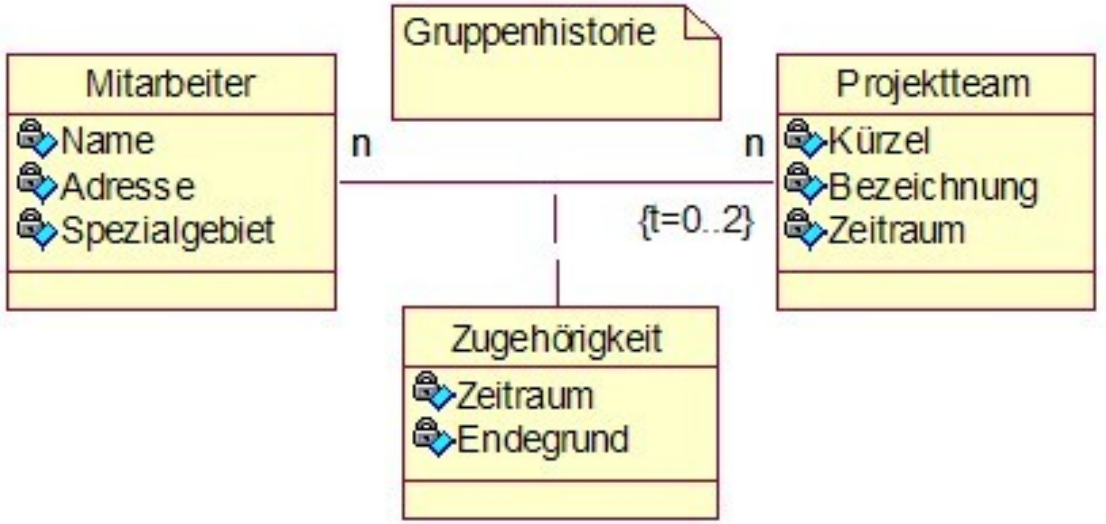


- Eigenschaften
  - Es liegt eine einfache Assoziation vor.
  - Mehrere Einzel-Objekte gehören – zu einem Zeitpunkt – zum selben Gruppen-Objekt.
  - Es ist jeweils zu prüfen, ob die Gruppe – zeitweise – ohne Einzel-Objekte existieren kann oder ob sie immer eine Mindestanzahl von Einzel-Objekten enthalten muss.
  - Objektverbindungen können auf- und abgebaut werden.

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Muster 10: Gruppenhistorie

- Die Zugehörigkeit zu einer Gruppe soll nicht zu einem Zeitpunkt, sondern über einen Zeitraum festgehalten werden.



- Eigenschaften
  - Ein Einzel-Objekt gehört – im Laufe der Zeit – zu unterschiedlichen Gruppen-Objekten.
  - Die Historie wird mittels einer assoziativen Klasse modelliert.
  - Dadurch ist die Zuordnung zwischen Einzel-Objekten und Gruppen deutlich sichtbar.
  - Die zeitliche Restriktion {t=k} (k = gültige Kardinalität) sagt aus, was zu einem Zeitpunkt gelten muss.
  - Da Informationen über einen Zeitraum festzuhalten sind, bleiben erstellte Objektverbindungen bestehen und es werden nur Verbindungen hinzugefügt.

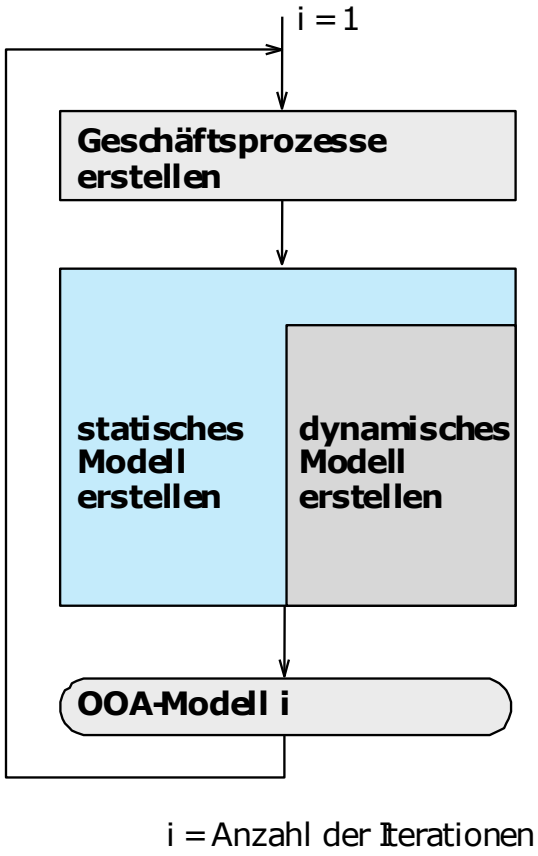
Methodische Vorgehensweise

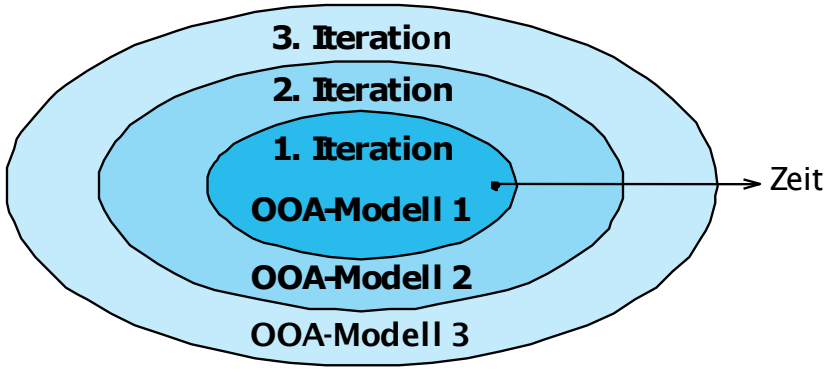
- Bei den objektorientierten Methoden lassen sich prinzipiell drei unterscheiden:
  - Orientierung an den Informationen des Systems (statisches Modell)
  - Orientierung an der Funktionalität des Systems (dynamisches Modell)
  - Synthese von 1. und 2.

- Im folgenden wird die methodische Vorgehensweise von [Heide Balzert 99] bestehend aus: dem **Makroprozess** der die methodischen Schritte festlegt und methodischen Regeln, die in Form von Checklisten zur Verfügung stehen, dargestellt.
- Der Makroprozess beschreibt auf einem hohen Abstraktionsniveau, in welcher Reihenfolge die einzelnen Aufgaben zu Erstellung eines OOA-Modells auszuführen sind:
  - Ermitteln der relevanten Geschäftsprozesse (oft im Pflichtenheft erfolgt),
  - Ableitung von Klassen aus den Geschäftsprozessen,
  - Erstellen des statischen Modells,
  - Parallel dazu Erstellung des dynamischen Modells,
  - Berücksichtigung der Wechselwirkungen beider Modelle.

Modellbildung

- Der Makroprozess berücksichtigt die Gleichgewichtigkeit *(balancing)* von statischem und dynamischem Modell.
- Die Konzentration auf das statische vor dem dynamischen Modell sorgt für eine größere Stabilität.





- Die hier beschriebene Methode realisiert einen evolutionären, iterativen Entwicklungsprozess. Zunächst wird eine OOA für den Produktkern erstellt, der anschließend zu entwerfen und implementieren ist. Der Kern wird in weiteren Zyklen erweitert, bis ein auslieferbares System entsteht.

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Analyse im Großen

- Es handelt sich bei der Analyse im Großen um Aufgaben, die nicht spezifisch für eine objektorientierte Entwicklung sind, während statische und dynamische Modellierung einen objektorientierten Charakter haben.

1. Geschäftsprozesse aufstellen
   - Erstellen der wesentlichen Geschäftsprozesse
   - Beschreibung Geschäftsprozesse
   - Geschäftsprozessdiagramm

3. Pakete bilden
   - Bilden von Teilsystemen, d.h. zusammenfassen von Modellelementen zu Paketen
   - Bei großen Systemen, die i. Allg. durch mehrere Teams bearbeitet werden, muss die Bildung von Paketen am Anfang stehen
   - Paketdiagramm.

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Schritte zum statischen Modell I

1. Klassen identifizieren
   - Für jede Klasse nur so viele Attribute und Operationen identifizieren, wie für das Problemverständnis und das einwandfreie Erkennen der Klasse notwendig sind.
     → *Klassendiagramm*
     → *Kurzbeschreibung Klassen*
2. Assoziationen identifizieren
   - Zunächst nur die reinen Verbindungen eintragen, d. h. noch keine genaueren Angaben, z. B. zur Kardinalität oder zur Art der Assoziation, machen.
     → *Klassendiagramm*
3. Attribute identifizieren
   - Identifizieren aller Attribute des Fachkonzepts.
     → *Klassendiagramm*
4. Vererbungsstrukturen identifizieren
   - Aufgrund der identifizierten Attribute Vererbungsstrukturen erstellen.
     → *Klassendiagramm*

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Schritte zum statischen Modell II

1. Assoziationen vervollständigen
   - Endgültig festlegen, ob eine »normale« Assoziation, Aggregation oder Komposition vorliegt sowie Festlegung der Kardinalitäten, Rollen, Namen und Restriktionen.
     → *Klassendiagramm*
     → *Objektdiagramm*

6. Attribute spezifizieren
   - Für alle identifizierten Attribute eine vollständige Spezifikation erstellen.
     → *Attributspezifikation*

7. Muster identifizieren
   - Das Klassendiagramm daraufhin überprüfen, ob Muster enthalten sind und diese richtig modelliert wurden.

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Schritte zum dynamischen Modell

1. Szenarios erstellen
   - Jeden Geschäftsprozess durch eine Menge von Szenarios präzisieren.
     → *Sequenzdiagramm*
     → *Kollaborationsdiagramm*
     → Alternativ oder ergänzend können auch *Aktivitätsdiagramme* verwendet werden.
2. Zustandsautomat erstellen
   - Für jede Klasse prüfen, ob ein nicht-trivialer Lebenszyklus erstellt werden kann und muss.
     → *Zustandsdiagramm*
3. Operationen beschreiben
   - Überlegen, ob eine Beschreibung notwendig ist. Wenn ja, dann ist je nach Komplexitätsgrad die entsprechende Form zu wählen.
     → *Klassendiagramm*
     → *fachliche Beschreibung der Operationen*
     → *Zustandsautomaten*
     → *Aktivitätsdiagramme*

Alternative Makroprozesse

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Mögliche Alternativen zum beschriebenen balancierten Makroprozess:

- **Szenario-basierte Makroprozess**
  - bei umfangreichen funktionalen Anforderungen
  - es existieren keine alten Datenbestände
- **Daten-basierter Makroprozess**
  - bei umfangreichen existierenden Datenbeständen
  - der Umfang der funktionalen Anforderungen ist zunächst unbekannt.

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Checklisten

- Bei der praktischen Anwendung zeigt sich, dass eine grob beschriebene Vorgehensweise – wie der dargestellte Makroprozess – nicht ausreicht.
- Erfahrene Systemanalytiker wenden hunderte von Regeln an. Diese stehen in Form von Checklisten zur Verfügung.
- Für Erstellen der entsprechenden Diagramme und Spezifikationen sollten alle Punkte der Checkliste durchgegangen werden.
- Aufbau von Checklisten

Konstruktive Schritte	Wie findet man ein Modellelement
Analytische Schritte	Ist es ein „gutes" Modellelement? Konsistenzprüfung Fehlerquellen
Für klassische Entwickler	Welche methodischen Regeln helfen beim Übergang von der Datenmodellierung zur objektorientierten Analyse

- Für die Qualitätssicherung können die unter „Analytische Schritte" angegebenen Punkte benutzt werden.

→ *VL Qualitätsmanagement*

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Checkliste: Geschäftsprozess

- Die methodische Vorgehensweise und die Checkliste für Geschäfts-prozesse wurde bereits behandelt, da sie nicht spezifisch für eine objektorientierte Entwicklung sind.

- Ergebnisse:
  → *Geschäftsprozessdiagramm*
  → *Beschreibung der Geschäftsprozesse*

- Konstruktive Schritte:
  1. Akteure ermitteln
  2. Geschäftsprozesse für Standardverarbeitung ermitteln
     - mittels Akteuren
     - mittels Ereignissen
     - mittels Aufgabenbeschreibung
  3. Geschäftsprozesse für Sonderfälle ermitteln
  4. Aufteilen komplexer Geschäftsprozesse
  5. Gemeinsamkeiten der Geschäftsprozesse ermitteln

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Checkliste: Paket I

- Ergebnisse
  → Paketdiagramm
  - Erstellen eines Paketdiagramms
    - Jedem Paket werden Modellelemente zugeordnet.
    - Die Abhängigkeiten zwischen den Paketen werden spezifiziert.

- Konstruktive Schritte
  1. Welche Pakete ergeben sich durch top-down-Vorgehen?
     Bei großen Anwendungen:
     - Noch vor der Formulierung von Geschäftsprozessen: Unterteilen des Gesamtsystems in Teilsysteme (Pakete).
     - Umfangreiche Pakete in weitere Pakete zerlegen.
  2. Welche Pakete ergeben sich durch bottom-up-Vorgehen?
     Bei kleinen und mittleren Anwendungen:
     - Klassen unter einem Oberbegriff zusammenfassen.

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Checkliste: Paket II

- Analytische Schritte
  3. Bildet das Paket eine abgeschlossene Einheit?
     - Enthält eigenständigen Themenbereich.
     - Enthält Klassen, die logisch zusammengehören.
     - Es erlaubt eine Betrachtung des Systems auf einer höheren Abstraktionsebene.
     - Vererbungsstrukturen liegen möglichst innerhalb eines Pakets.
     - Aggregationen sind nicht durchtrennt.
     - Möglichst wenig Assoziationen sind durchtrennt.
  4. Ist der Paketname geeignet?
     - Der Inhalt eines Pakets muss mit 25 Worten oder weniger beschreibbar sein.
     - Aus der Beschreibung den Namen ableiten.
     - Paketnamen dürfen keine Verben enthalten.
  5. Fehlerquellen
     - Zu kleine Pakete.

Checkliste: Klasse I

- Ergebnisse
  - → Klassendiagramm
    - ° Jede Klasse – entweder nur mit Namen oder mit wenigen wichtigen Attributen/Operationen – in das Klassendiagramm eintragen.
  - → Kurzbeschreibung der Klassen

- Konstruktive Schritte
  1. Welche Klassen lassen sich mittels Dokumentanalyse identifizieren (bottom-up)?
     - ° Aus Formularen und Listen Attribute entnehmen und zu Klassen zusammenfassen.
     - ° Reengineering von Software-Altsystemen:
       - – Arbeitsabläufe anhand von Benutzerhandbüchern, Bildschirmmasken, Dateibeschreibungen ermitteln.
       - – Anhand des laufenden Systems die Funktionen der Geschäftsprozesse ausführen.
       - – Klassen mithilfe der Bildschirmmasken ermitteln.
     - ° Bei technischen Systemen bieten sich die realen Objekte als Ausgangsbasis an, z.B. Lagermodul.

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Checkliste: Klasse II

- Konstruktive Schritte
  2. Welche Klassen lassen sich aus der Beschreibung der Geschäftsprozesse identifizieren (top-down)?
     - Beschreibung nach Klassen durchsuchen
       - Oft sind Substantive potenzielle Klassen.
     - Potenzielle Klassen auf Attribute überprüfen
     - Akteure, über die man sich etwas »merken« muss, sind potenzielle Klassen.
  3. Zu welchen Kategorien gehören die Klassen?
     - Konkrete Objekte bzw. Dinge, z.B. PKW
     - Personen und deren Rollen, z.B. Kunde, Dozent
     - Informationen über Aktionen, z.B. Banküberweisung durchführen
     - Orte, z.B. Wartezimmer
     - Organisationen, z.B. Filiale
     - Behälter, z.B. Lagerplatz
     - Dinge in einem Behälter, z.B. Reifen in einem Lagerplatz
     - Ereignisse, z.B. Eheschließung
     - Kataloge, z.B. Produktkatalog
     - Verträge, z.B. Autokaufvertrag

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Checkliste: Klasse III

- Analytische Schritte
  4. Liegt ein aussagefähiger Klassenname vor?
     - Der Klassenname soll
       - der Fachterminologie entsprechen,
       - ein Substantiv im Singular sein,
       - dasselbe ausdrücken wie die Gesamtheit der Attribute,
       - nicht die Rolle dieser Klasse in einer Beziehung zu einer anderen Klasse beschreiben,
       - eindeutig im Paket bzw. im System sein,
       - nicht dasselbe ausdrücken wie der Name einer anderen Klasse.
  5. Ist das gewählte Abstraktionsniveau richtig?
     - Die Ziele sind ***nicht***
       - möglichst viele Klassen oder
       - Klassen geringer Komplexität zu identifizieren.
  6. Wann liegt keine Klasse vor?
     - Keine Klassen bilden, um Mengen von Objekten zu verwalten.

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Checkliste: Klasse IV

- Analytische Schritte
  7. Fehlerquellen
     - Zu kleine Klassen
     - Aus jedem Report eine Klasse modellieren
     - Klasse modelliert Benutzungsoberfläche
     - Klasse modelliert Entwurfs- oder Implementierungsdetails

- Für klassische Entwickler
  8. Identifizieren von Klassen für Datenmodellierer
     - Eine potenzielle Klasse ist jeder Entitätstyp und jeder Beziehungstyp mit Attributen.

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Checkliste: Assoziationen I

- Ergebnis
  → Klassendiagramm
    - Assoziationen im ersten Schritt nur als Linie eintragen.
    - Noch keine Kardinalitäten, Aggregationen, Kompositionen, Rollen, Namen, Restriktionen.

- Konstruktive Schritte
  1. Welche Assoziationen lassen sich mittels Dokumentanalyse ableiten?
     - Aus Primär- und Fremdschlüsseln ermitteln.
  2. Welche Assoziationen lassen sich aus Beschreibungen der Geschäftsprozesse ermitteln?
     - Nach Verben suchen, insbesondere:
       - räumliche Nähe (in der Nähe von)
       - Besitz (hat)
       - Aktionen (fährt)
       - Kommunikation (redet mit), (verheiratet mit)
       - allgemeine Beziehungen

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Checkliste: Assoziationen II

- Konstruktive Schritte
  3. Liegt eine Assoziation folgender Kategorien vor?
     - A ist physische Komponente von B
     - A ist logische Komponente von B
     - A ist eine Beschreibung für B
     - A ist eine Zeile einer Liste B
     - A ist ein Mitglied von B.
     - A ist eine organisatorische Einheit von B
     - A benutzt B
     - A kommuniziert mit B
     - A besitzt B
  4. Welche Restriktionen muss die Assoziation erfüllen?
     - Eine Assoziation: {ordered}
     - Mehrere Assoziationen: {or}, {subset}
  5. Welche Rollen spielen die beteiligten Klassen?
     - Rollennamen angeben, wenn
       - Assoziation zwischen derselben Klasse existiert
       - eine Klasse in verschiedenen Assoziationen verschiedene Rollen spielt
       - durch den Rollennamen die Bedeutung der Klasse in der Assoziation genauer spezifiziert werden kann.

Checkliste: Assoziationen III

- Analytische Schritte
  6. Ist eine Benennung notwendig oder sinnvoll?
     - Notwendig, wenn zwischen zwei Klassen mehrere Assoziationen bestehen.
     - Rollennamen (Substantive) bevorzugen.
     - Rollennamen sind bei reflexiven Assoziationen immer notwendig.
  7. Liegt eine 1:1-Assoziation vor?
     - Zwei Klassen sind zu modellieren, wenn...
       - die Verbindung in einer oder beiden Richtungen optional ist und sich die Verbindung
       - zwischen beiden Objekten ändern kann,
       - es sich um zwei umfangreiche Klassen handelt,
       - die beiden Klassen eine unterschiedliche Semantik besitzen.
  8. Gibt es zwischen zwei Klassen mehrere Assoziationen?
     - Prüfen, ob die Assoziationen
       - eine unterschiedliche Bedeutung besitzen oder/und
       - unterschiedliche Kardinalitäten haben.

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Checkliste: Assoziationen IV

- Analytische Schritte

9. Sind abgeleitete Assoziationen korrekt verwendet?
   - Fügen keine neue Information zum Modell hinzu.
   - Lassen sich mittels Objektdiagrammen erkennen.
10. Assoziative Klasse oder eigenständige Klasse?
   - Assoziative Klasse betont die Assoziation zwischen den beteiligten Klassen.
   - Assoziative Klassen lassen sich in eigenständige Klassen wandeln.
11. Fehlerquellen
   - Verwechseln von Assoziation mit Vererbung.

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Checkliste: Attribut I

- Ergebnisse
  - → Klassendiagramm
    - Für jedes Attribut den Namen in das Klassendiagramm eintragen
    - Klassenattribute und abgeleitete Attribute kennzeichnen
  - → Attributspezifikation
    - Jedes Attribut spezifizieren
    - Für komplexe Attribute sind ggf. entsprechende Typen zu definieren.

- Konstruktive Schritte
  1. Attribute durch Dokumentanalyse identifizieren
     - Einfache Attribute sind ggf. zu Datenstrukturen zusammenzufassen.
     - Prüfen, ob alle Attribute wirklich notwendig sind.
     - Für jedes Attribut prüfen, ob es »im Laufe seines Lebens« einen Wert annehmen kann und ob diese Werte an der Benutzungs-oberfläche sichtbar sind.
  2. Attribute durch Geschäftsprozesse identifizieren
     - Benötigte Daten zur Ausführung der Aufgaben eines Geschäftsprozesses.
     - Benötigte Daten für Listenfunktionalität.

Checkliste: Attribut II

- Konstruktive Schritte
  3. Wurden geeignete Attributtypen gewählt und u. U. als elementare Klasse beschrieben?
     - Vorgegebene Typen nur verwenden, wenn problemadäquat.
     - Attribute beliebigen Typs definieren, um problemadäquate Modellierung auf ausreichendem Abstraktionsniveau zu erreichen.
     - Für komplexe Attribute zur Konstruktion der Typen das Klassenkonzept verwenden (elementare Klassen).

- Analytische Schritte
  4. Ist der Attributname geeignet?
     - Der Attributname soll
       - kurz, eindeutig und verständlich sein
       - ein Substantiv oder Adjektiv-Substantiv sein
       - den Namen der Klasse nicht wiederholen
       - bei komplexen (strukturierten) Attributen der Gesamtheit der Komponenten entsprechen
       - nur fachspezifische oder allgemein übliche Abkürzungen enthalten, z.B. PLZ

Checkliste: Attribut III

- Analytische Schritte
  5. Klasse oder komplexes Attribut?
     - Klasse:
       - Objektidentität, gleichgewichtige Bedeutung im System, Existenz unabhängig von der Existenz anderer Objekte, Zugriff in beiden Richtungen grundsätzlich möglich.
     - Attribut
       - keine Objektidentität, Existenz abhängig von Existenz anderer Objekte, Zugriff immer über das Objekt, untergeordnete Bedeutung
  6. Wurde das richtige Abstraktionsniveau gewählt?
     - Wurden komplexe Attribute gebildet?
     - Bilden Attribute geeignete Datenstrukturen?
     - Anzahl der Attribute pro Klasse angemessen?
  7. Gehört Attribut zur Klasse oder einer Assoziation?
     - Test:
       - Muss Attribut auch dann zur Klasse gehören, wenn die Klasse isoliert von anderen Klassen betrachtet wird?
         - » Wenn ja, dann gehört das Attribut zu dieser Klasse.
         - » Wenn nein, dann ist zu prüfen, ob es sich einer Assoziation zuordnen lässt .
         - » Sonst: Klasse oder Assoziation vergessen.

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Checkliste: Attribut IV

- Analytische Schritte
  8. Liegen Klassenattribute vor?
     - Ein Klassenattribut liegt vor, wenn gilt:
       - Alle Objekte der Klasse besitzen für dieses Attribut denselben Attributwert.
       - Es sollen Informationen über die Gesamtheit der Objekte modelliert werden.
  9. Sind Schlüsselattribute fachlich notwendig?
     - Schlüsselattribute werden nur dann eingetragen, wenn sie Bestandteil des Fachkonzepts sind.
  10. Werden abgeleitete Attribute korrekt verwendet?
     - Information ist für den Benutzer sichtbar.
     - Lesbarkeit wird verbessert.
  11. Wann wird ein Attribut nicht eingetragen?
     - Dient nur zum Identifizieren der Objekte.
     - Dient lediglich dazu, eine andere Klasse zu referenzieren, d. h. es realisiert eine Assoziation.
     - Beschreibt den internen Zustand eines Lebenszyklus und ist außerhalb des Objekts nicht sichtbar.
     - Beschreibt Entwurfs- / Implementierungsdetails.
     - Ist abgeleitetes Attribut, das nur eingefügt wurde, um Berechnungszeit zu sparen.
  12. Fehlerquellen
     - Verwenden atomarer Attribute anstelle von komplexen Datenstrukturen.
     - Formulieren von Assoziationen als Attribute.

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Checkliste: Vererbung

- Ergebnis
  → Klassendiagramm
  - Alle Vererbungsstrukturen eintragen und abstrakte Klassen bilden

- Konstruktive Schritte
  1. Ergibt Generalisierung eine Einfachvererbung?
     - Neue Oberklasse aus gleichartigen Klassen?
     - Vorhandene Klasse als Oberklasse geeignet?
  2. Ergibt Spezialisierung eine Einfachvererbung?
     - Kann jedes Objekt der Klasse für jedes Attribut einen Wert annehmen?
     - Kann jede Operation auf jedes Objekt der Klasse angewendet werden?

- Analytische Schritte
  3. Liegt eine »gute« Vererbungsstruktur vor?
     - Verbessert die Vererbungsstruktur das Verständnis des Modells?
     - Benötigt jede Unterklasse alle geerbten Attribute, Operationen und Assoziationen?
     - Liegt eine Ist-ein-Beziehung vor?
     - Entspricht die Vererbungsstruktur den »natürlichen« Strukturen des Problembereichs?
     - Besitzt sie maximal 3 bis 5 Hierarchiestufen?
  4. Wann liegt keine Vererbung vor?

Checkliste: Kardinalitäten

- Ergebnis
  - → Klassendiagramm
  - Alle Kardinalitäten eintragen

- Konstruktive/analytische Schritte
  1. Schnappschuss oder Historie modellieren?
     - Aus den Anfragen an das System ergibt sich, ob ein Schnappschuss (1- bzw. 0..1-Kardinalität) oder die Historie (many-Kardinalität) zu modellieren ist
       - **Schnappschuss**: eine alte Verbindung wird gelöst wird, wenn eine neue aufgebaut wird
       - **Historie**: eine neue Verbindung wird zwischen den jeweiligen Objekten hinzugefügt.
  2. Liegt eine Muss- oder Kann-Assoziation vor?
  3. Enthält die Kardinalität feste Werte?
     - Ist eine Obergrenze vom Problembereich her (oft bei technischen Systemen) zwingend vorgegeben?
       - Im Zweifelsfall mit variablen Obergrenzen arbeiten.
     - Ist die Untergrenze vom Problembereich her zwingend vorgegeben?
       - Im Zweifelsfall mit »0« arbeiten.
     - Gelten besondere Restriktionen für die Kardinalitäten?
  4. Fehlerquelle
     - Oft werden Muss-Assoziationen verwendet, wo sie nicht benötigt werden.

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Checkliste: Assoziation, Aggregation, Komposition

- Ergebnis
  - → Klassendiagramm
  - Alle Aggregationen und Kompositionen eintragen
- Konstruktive/Analytische Schritte
  1. Für eine Komposition gilt:
     - Es liegt eine Ist-Teil-von-Beziehung vor
     - Kardinalität bei der Aggregatklasse: 0..1 oder 1
       - Lebensdauer der Teile ist an die des Ganzen gebunden.
       - Ein Teil darf jedoch zuvor explizit entfernt werden.
     - Funktionen des Ganzen werden automatisch auf seine Teile angewendet
     - Muster bilden eine gute Orientierungshilfe
       - Liste (Bestellung – Bestellposition), Baugruppe (Auto – Motor) und Stückliste mit physikalischem Enthaltensein.
  2. Für eine Aggregation gilt:
     - Es liegt eine Ist-Teil-von-Beziehung mit shared aggregation (ein Teilobjekt kann mehreren Aggregatobjekten zugeordnet werden) vor.
     - Sie ist selten.
  3. Im Zweifelsfall: immer einfache Assoziation
     - Bei dem geringsten Zweifel an einer Komposition oder Aggregation immer die Assoziation wählen.
  4. Fehlerquellen
     - Prinzipiell ist es möglich, jedes Attribut als Klasse zu modellieren und mittels einer Komposition mit der ursprünglichen Klasse zu verbinden.
       - Dies führt jedoch zu schlechten Modellen.

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Checkliste: Szenario I

- Ergebnisse
  → Sequenzdiagramm, Kollaborationsdiagramm
  - Für jedes relevante Szenario: Sequenzdiagramm
  - Alternativ: Kollaborationsdiagramme
- Konstruktive Schritte
  1. Aus jedem Geschäftsprozess (GP) mehrere Szenarios entwickeln
     - Variationen von GPs ermitteln
     - Standardausführung und Alternativen
     - Positive und negative Fälle unterscheiden
     - Prüfen, welche Szenarios wichtig sind
     - Diagramme benennen und beschreiben.
  2. Ablauf relevanter Szenarios durch Sequenz- oder Kollaborationsdiagramme beschreiben
     - Beteiligte Klassen
     - Aufgaben in Operationen zerlegen
     - Reihenfolge der Operationen prüfen
     - UML erlaubt es, im Sequenzdiagramm Bedingungen anzugeben
       - Damit können mehrere Variationen durch ein einziges Sequenzdiagramm beschrieben werden.
  3. Operationen den Klassen zuordnen
     - Werden nur Attribute einer Klasse benötigt, dann ist die Operation dieser Klasse zuzuordnen.
     - Konstruktoren sind der jeweiligen Klasse selbst zuzuordnen.

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Checkliste: Szenario II

- Analytische Schritte
  4. Sind die Empfänger-Objekte erreichbar?
     - Assoziation existiert (permanente Verbindung)
     - Identität kann dynamisch ermittelt werden (temporäre Verbindung)
  5. Sequenzdiagramm konsistent mit Klassendiagramm?
     - Alle Klassen auch im Klassendiagramm enthalten
     - Mit Ausnahme von impliziten Operationen werden nur Operationen aus dem Klassendiagramm eingetragen
  6. Fehlerquellen
     - Benutzungsoberfläche wird beschrieben
     - Zu viele Details sind beschrieben.

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Checkliste: Zustandsautomaten I

- Ergebnis
  → Zustandsdiagramm
- Konstruktive Schritte
  1. Existiert ein nicht-trivialer Lebenszyklus?
  2. Welche Zustände enthält der Automat?
     - Ausgangsbasis ist der Anfangszustand
     - Durch welche Ereignisse wird ein Zustand verlassen?
     - Ereignisse umgangssprachlich beschreiben, was »von außen auf das Objekt einwirkt«
     - Welche Folgezustände treten auf?
     - Wodurch wird der Zustand definiert (Attributwerte, Objektverbindungen)?
  3. Existieren Endzustände?
     - Nur bei linearen Lebenszyklen:
       - Das Objekt hört auf zu existieren.
       - Das Objekt existiert weiterhin, aber sein dynamisches Verhalten ist nicht länger von Interesse (schlafendes Objekt).
  4. Welche Operationen besitzt das Objekt?
     - Jede zustandsabhängige Operation aus dem Klassendiagramm eintragen.
     - Operationen, die in jedem Zustand ausgeführt werden können, nicht eintragen.
     - Prüfen, ob weitere Operationen notwendig sind.

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Checkliste: Zustandsautomaten II

- Konstruktive Schritte
  4. Sind Operationen als Aktivitäten oder als Aktionen zu modellieren?
  6. Welche Ereignisse sind zu modellieren?
     - Externe Ereignisse:
       - vom Benutzer
       - von anderen Objekten
     - Zeitliche Ereignisse:
       - Zeitdauer
       - Zeitpunkt
     - Intern generierte Ereignisse des betrachteten Objekts
     - Welche Fehlersituationen können auftreten und wie soll das Objekt darauf reagieren (Ausnahmebehandlung)?
       - Zuerst immer das Normalverhalten und erst im zweiten Schritt das Fehlerverhalten betrachten.

Checkliste: Zustandsautomaten III

- Analytische Schritte
  7. Geeigneter Zustandsname
     - Beschreibt eine bestimmte Zeitspanne.
     - Kein Verb
     - Kann entfallen, wenn er keine zusätzliche Information enthält.
  8. Ist der Objekt-Lebenszyklus konsistent mit der Liste der Operationen?
     - Gibt es für jede Operation mindestens einen Zustand, in dem das Objekt auf die entsprechende Botschaft reagieren kann?
     - Sind alle Aktivitäten und Aktionen auch Operationen des Klassendiagramms?
  9. Sind alle Zustandsübergänge korrekt eingetragen?
     - Ist jeder Zustand erreichbar?
     - Kann jeder Zustand – mit Ausnahme der Endzustände – verlassen werden?
     - Sind die Zustandsübergänge eindeutig?
  10. Fehlerquellen
     - Modellierung der Benutzungsoberfläche im Lebenszyklus
     - Gedankengut aus den Programmablaufplänen übernommen
       - »Schleifen« dürfen nicht vorkommen
       - Bedingungen sind immer mit einem Ereignis verknüpft.

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Checkliste: Operationen

- Ergebnisse
  - → Klassendiagramm: Operationen eingetragen
  - Beschreibung der Operationen
- Konstruktive Schritte
  1. Operationen ins Klassendiagramm eintragen
     - Aus Szenarios & Zustandsautomat übernehmen
     - Listenoperationen hinzufügen
     - Keine Verwaltungsoperationen eintragen
  2. Vererbung von Operationen berücksichtigen
     - Operationen so hoch wie möglich in der Hierarchie eintragen.
  3. Beschreibungen erstellen (nur wenn nötig).
- Analytische Schritte
  4. Besitzt die Operation einen geeigneten Namen?
     - Beginnt mit einem Verb.
     - Beschreibt, was die Operation »tut«.
  5. Erfüllt jede Operation die Qualitätskriterien?
     - Angemessener Umfang
     - Funktionale Bindung, d.h . jede Operation realisiert eine in sich abgeschlossene Funktion.
  6. Ist das balancing erfüllt?
     - Alle Attribute werden von den Operationen benötigt.
  7. Fehlerquellen
     - Keine Benutzungsoberfläche

Aufwandsschätzung

- Object Point-Methode [Sneed 96]
  - Geschäftsprozess-Tabelle bzw. Prozess-Tabelle
  - Klassen-Tabelle
  - Botschaften-Tabelle bzw. Nachrichten-Tabelle
  - Objekt-Punkte = Prozess-Punkte + Klassen-Punkte + Botschaften-Punkte
  - Die Objekt-Punkte werden mit einem Qualitätsfaktor QF multipliziert
    - Dazu werden 12 Qualitätsmaße durch ihre prozentuale Erfüllung definiert und nach festgelegten Regeln in eine Multiplikationsfaktor konvertiert.

Zusammenfassung

Die OOA verwendet die Basiskonzepte der Objektorientierung - Objekt, Klasse, Attribute, Operationen – ergänzt um die statischen Konzepte Assoziation, Vererbung, Paket und die dynamischen Konzepte Botschaft, Geschäftsprozess, Zustandsautomat und Szenario um die **fachliche Problemlösung** zu modellieren. Dabei entstehen ein **statisches** und ein **dynamisches Modell**, die sich gegenseitig beeinflussen und ausbalanciert sein sollen.

Muster (patterns) sind bewährte generische Lösungen für immer wiederkehrende Probleme, die in bestimmten Situationen auftreten. **OOA-Muster** werden für die Analyse und Konstruktion von OOA-Modellen benutzt. Wichtige OOA-Muster sind: Liste, Exemplartyp, Baugruppe, Stückliste, Koordinator, Rollen, wechselnde Rollen, Historie, Gruppe und Gruppenhistorie.

Das Erstellen eines OOA-Modells auf der Grundlage schriftlicher Informationen, z. B. eines Pflichtenheftes, oder mündlicher Informationen, z. B. durch Interviews, gehört zu den schwierigsten Aufgaben der Software-Technik.

Es handelt sich um einen iterativen Vorgang. Beschrieben wurde ein **Makroprozess**, der die Gleichgewichtigkeit von statischem und dynamischem Modell berücksichtigt. Für jedes objektorientierte Konzept stehen einheitlich aufgebaute **Checklisten** zur Verfügung, die die Konstruktion und Analyse unterstützen.

Literatur

- Balzert, Helmut, Lehrbuch der Softwaretechnik, 2. Auflage, Spektrum Akademischer Verlag,Heidelberg, 2000

- Coad P. mit North D., Mayfield M., Object Models Strategies, Patterns and Applications, Englewood Cliffs: Yourdon Press, Prentice Hall, 1995.

- Balzert, Heide, Lehrbuch der Objektmodellierung – Analyse und Entwurf, Spektrum Akademischer Verlag, Heidelberg, 1999.

- Sneed H.M., Schätzung der Entwicklungskosten von objektorientierter Software, Informatik-Spektrum 19, S. 133-140, 1996.