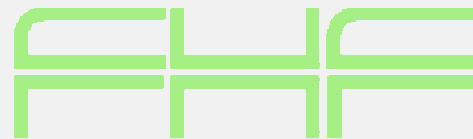


# Software-Architektur-Modellierung mit UML

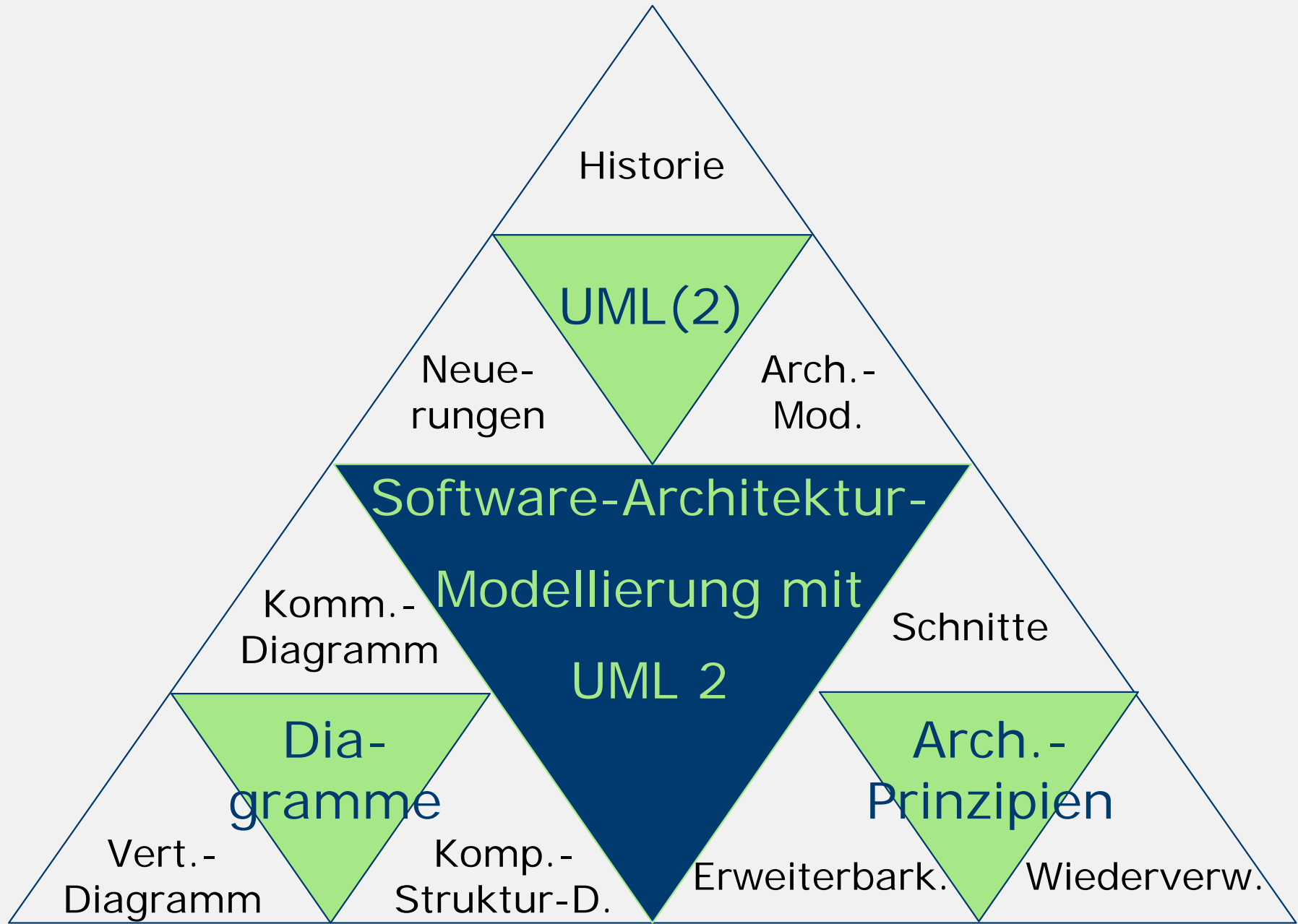


Prof. Mario Jeckle

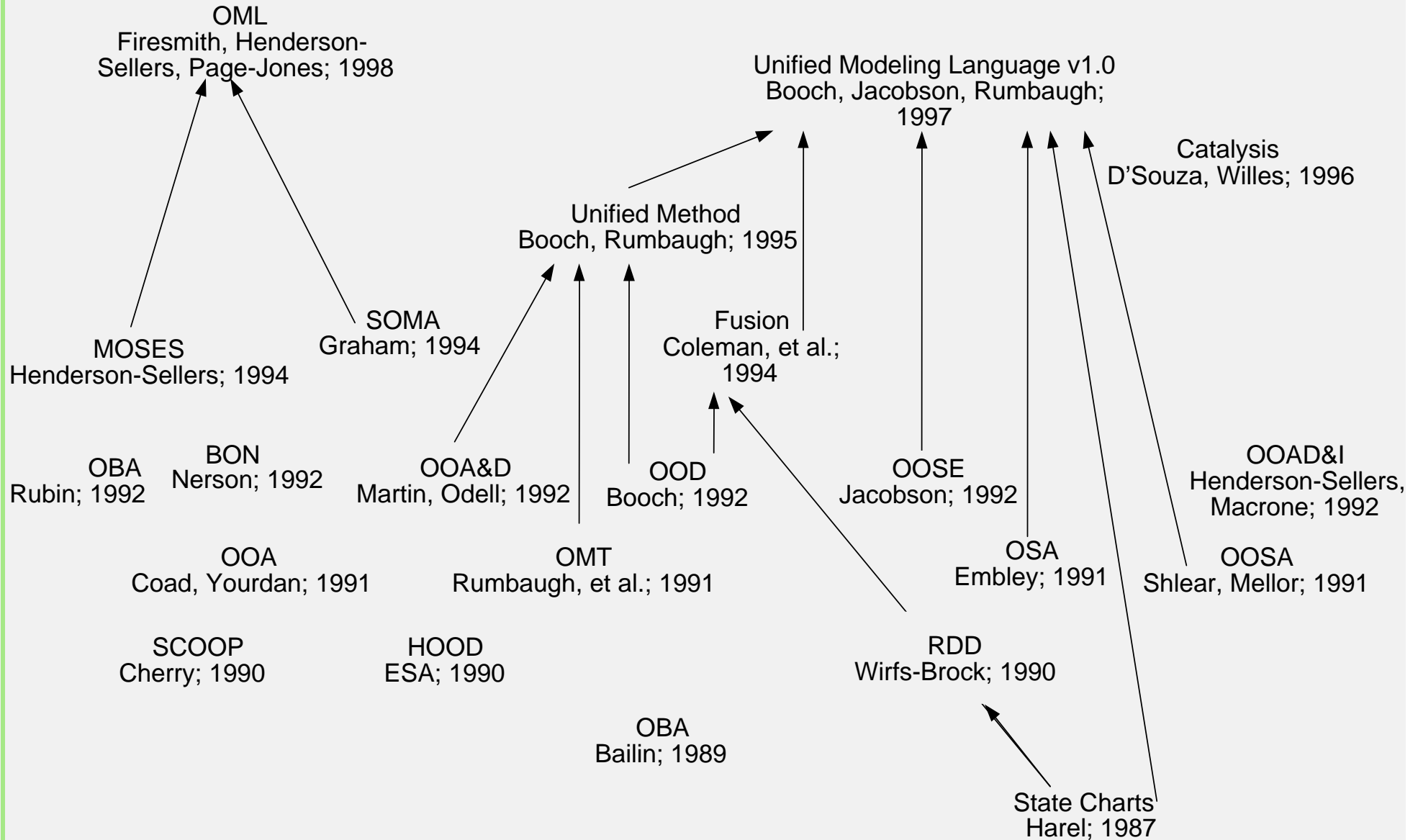
Fachhochschule Furtwangen

[mario@jeckle.de](mailto:mario@jeckle.de)

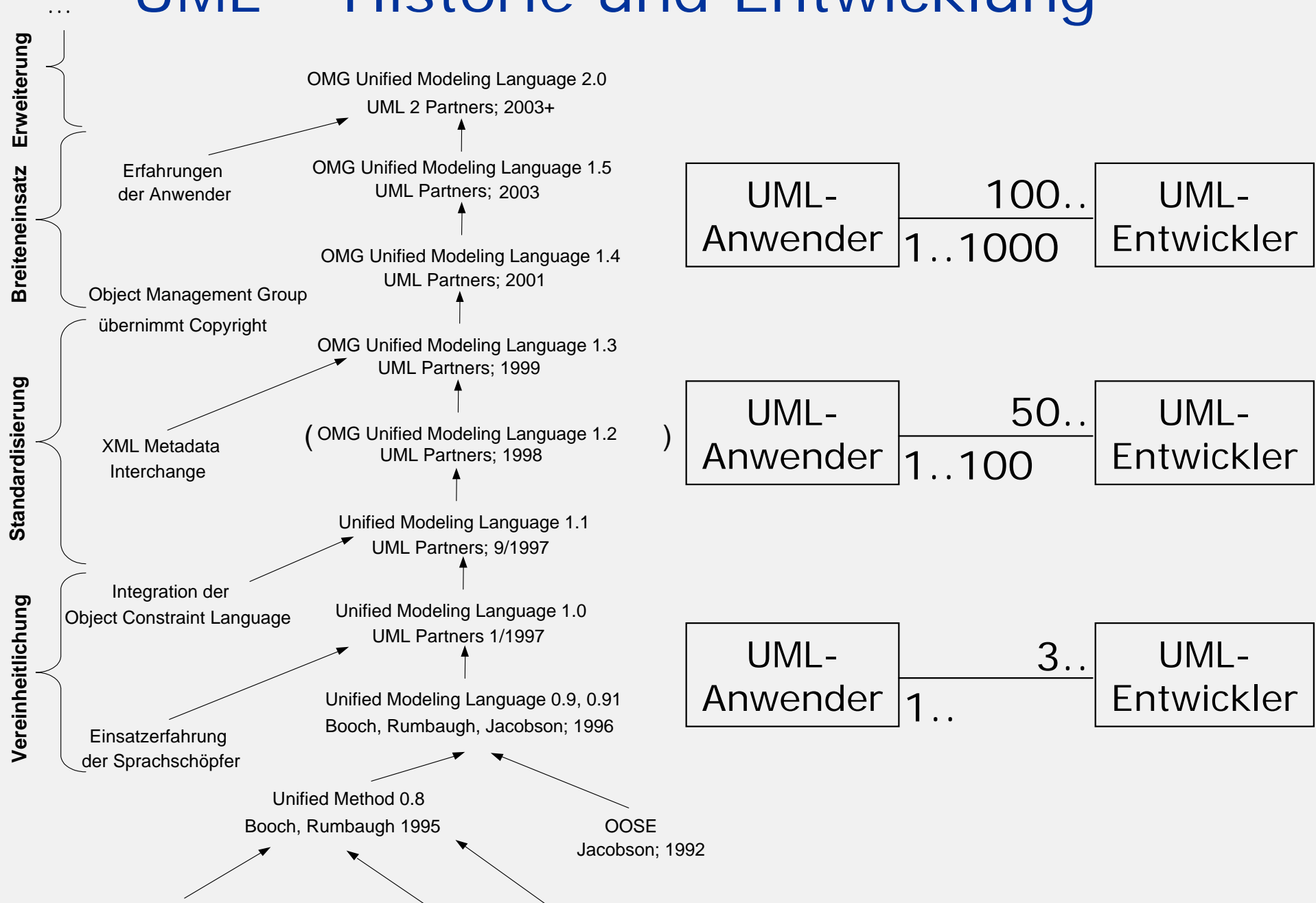
<http://www.jeckle.de>



# UML – Historie



# UML – Historie und Entwicklung



# UML 2 – Warum eine neue Version?

- **Evolution**

- Der Markt hat sich bewegt ...
  - Neue Programmiersprachen (z.B. C#, Python, PHP)
  - Neue Anwendungsdomänen  
(z.B. Serverprogrammierung, Echtzeitanwendungen, Architekturmodellierung)

- **Erfahrung**

- Für einige Einsatzgebiete bietet UML v1.x ...
  - Manchmal zu wenig Konstrukte
  - Manchmal zu viele
  - Manchmal so viele, dass die sinnvolle Auswahl schwerfällt

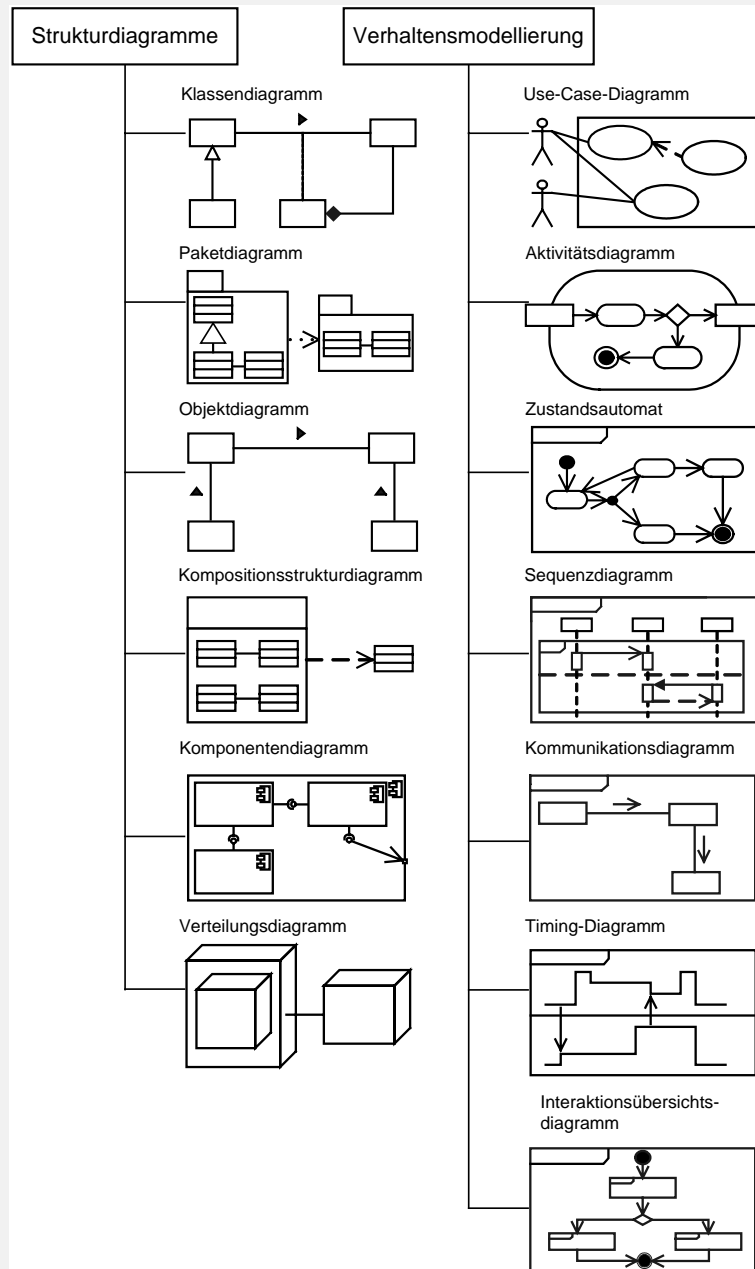
- **Eliminierung**

- Einige Programmiersprachen verschwinden (z. B. C++)
- Einige früher als modellierungsnah eingestufte Konzepte entwickeln sich inzwischen getrennt von UML weiter  
(z. B. Entwicklungsprozesse, Codegenerierung)

# Ziele der UML 2

- **Übersichtlichkeit**
  - Weniger graphische Modellkonstrukte
  - Weniger Basiskonzepte
  - Wiederverwendung von Basiskonzepten
- **Präzisionssteigerung**
  - Reformulierung des Meta-Modells
  - Weitestgehende OCL-Verwendung
  - Unveränderte Wiederverwendung von Basiskonstrukten soweit sinnvoll möglich
- **Ausführbarkeit**
  - Erweiterte Zustandsmaschinen
  - Stärkere Beziehungen zwischen statischen und dynamischen Diagrammen
  - Integration erprobter Konzepte außerhalb der UML

# UML 2 -- Neuerungen



## Nur marginale Änderungen

- Klassendiagramm
- Use-Case-Diagramm
- Objektdiagramm

## Klein(-e Änderungen) aber oho:

- Paketdiagramm
- Verteilungsdiagramm

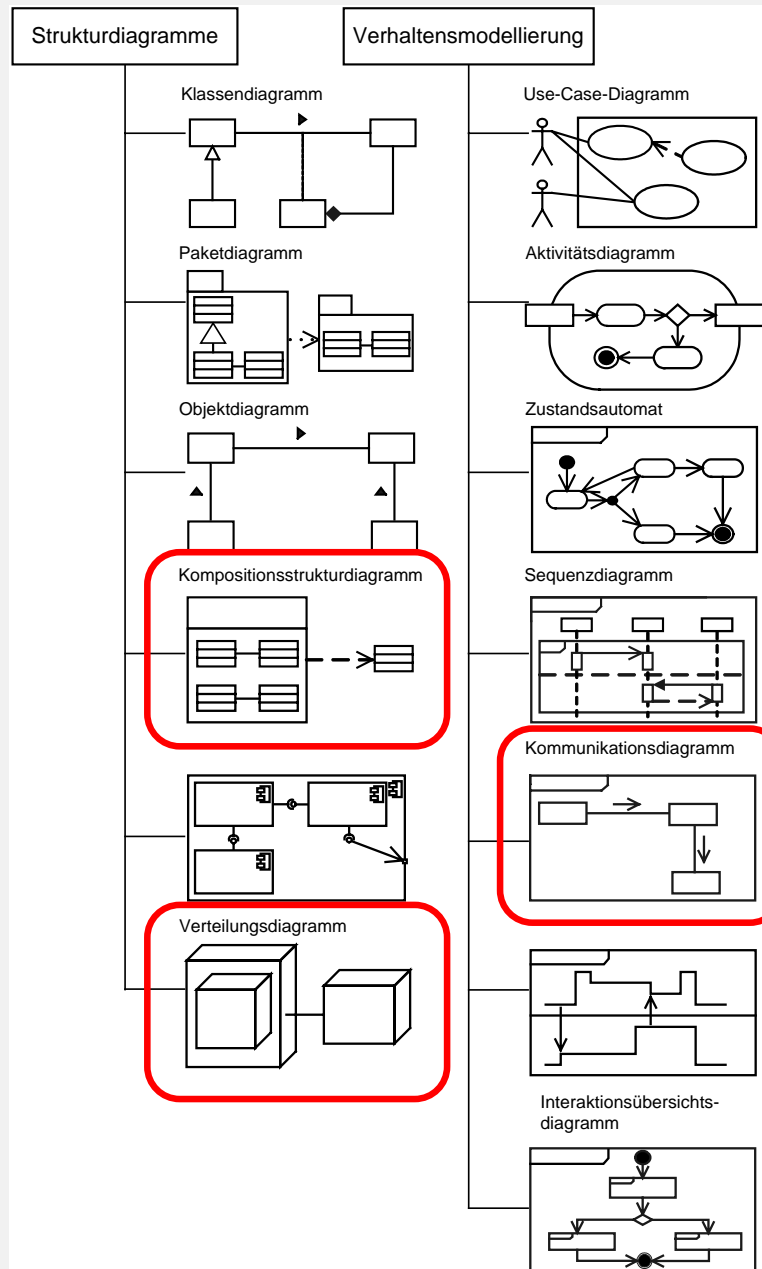
## Massiv und tiefgreifend verändert:

- Aktivitätsdiagramm
- Sequenzdiagramm
- Zustandsautomat

## Vollständig neu:

- Kompositionsstrukturdiagramm
- Interaktionsübersichtsdiagramm
- Timing-Diagramm
- Kommunikationsdiagramm

# UML 2 -- Neuerungen



## Nur marginale Änderungen

- Klassendiagramm
- Use-Case-Diagramm
- Objektdiagramm

## Klein(-e Änderungen) aber oho:

- Paketdiagramm
- **Verteilungsdiagramm**

## Massiv und tiefgreifend verändert:

- Aktivitätsdiagramm
- Sequenzdiagramm
- Zustandsautomat

## Vollständig neu:

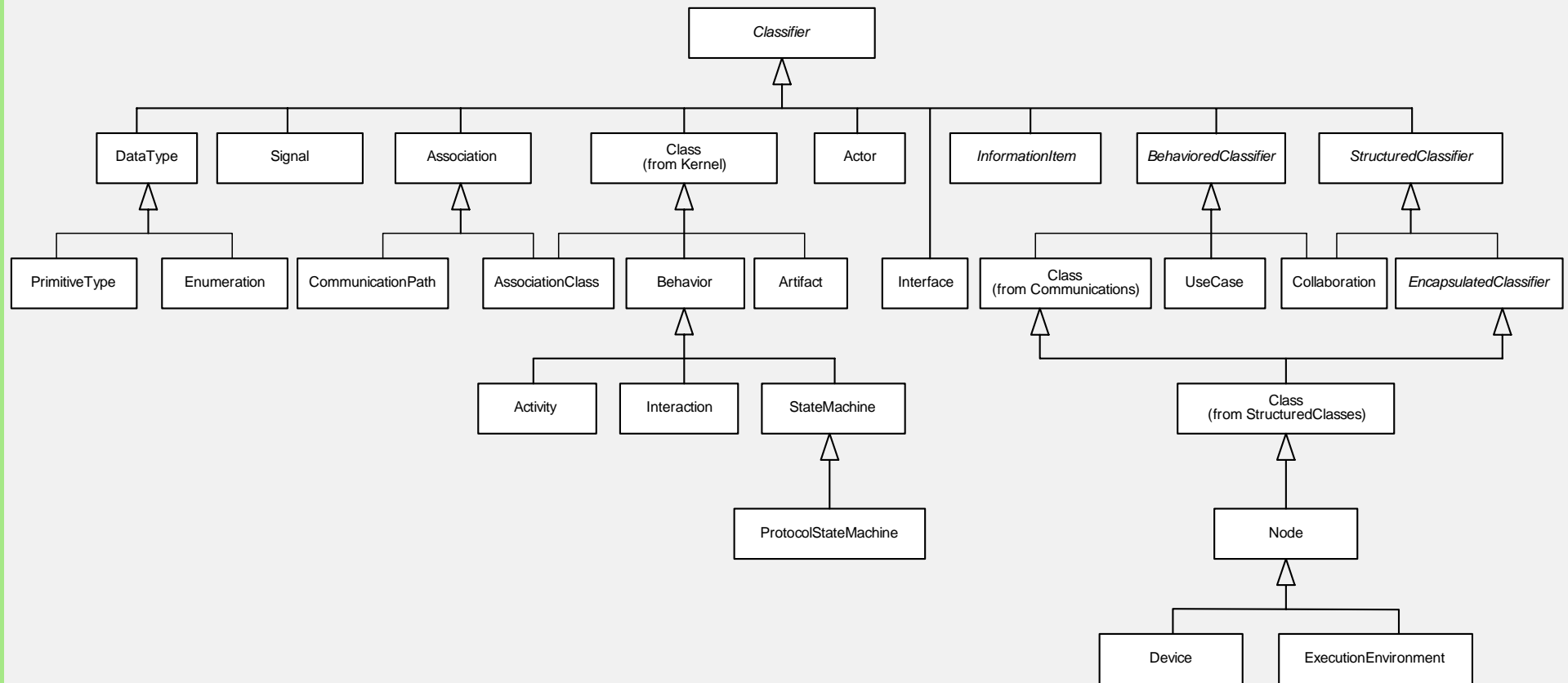
- **Kompositionsstrukturdiagramm**
- Interaktionsübersichtsdiagramm
- Timing-Diagramm
- **Kommunikationsdiagramm**



# Architekturdiagramme in UML 2

- **Strukturelle Architekturaspekte:  
Kompositionsstrukturdiagramm**
  - Zeigt die interne Struktur eines Classifiers sowie seine Interaktionspunkte mit anderen Architekturkomponenten. 
- **Dynamische Architekturaspekte:  
Kommunikationsdiagramm**
  - Zeigt Interaktion zwischen Architekturkomponenten (Lebenslinien).
- **Laufzeit-Architekturaspekte:  
Verteilungsdiagramm**
  - Zeigt über ein Netzwerk verbundene Hardware- oder Softwarekomponenten (Rechner, Ausführungsumgebung), die zur Ausführung benötigt werden.

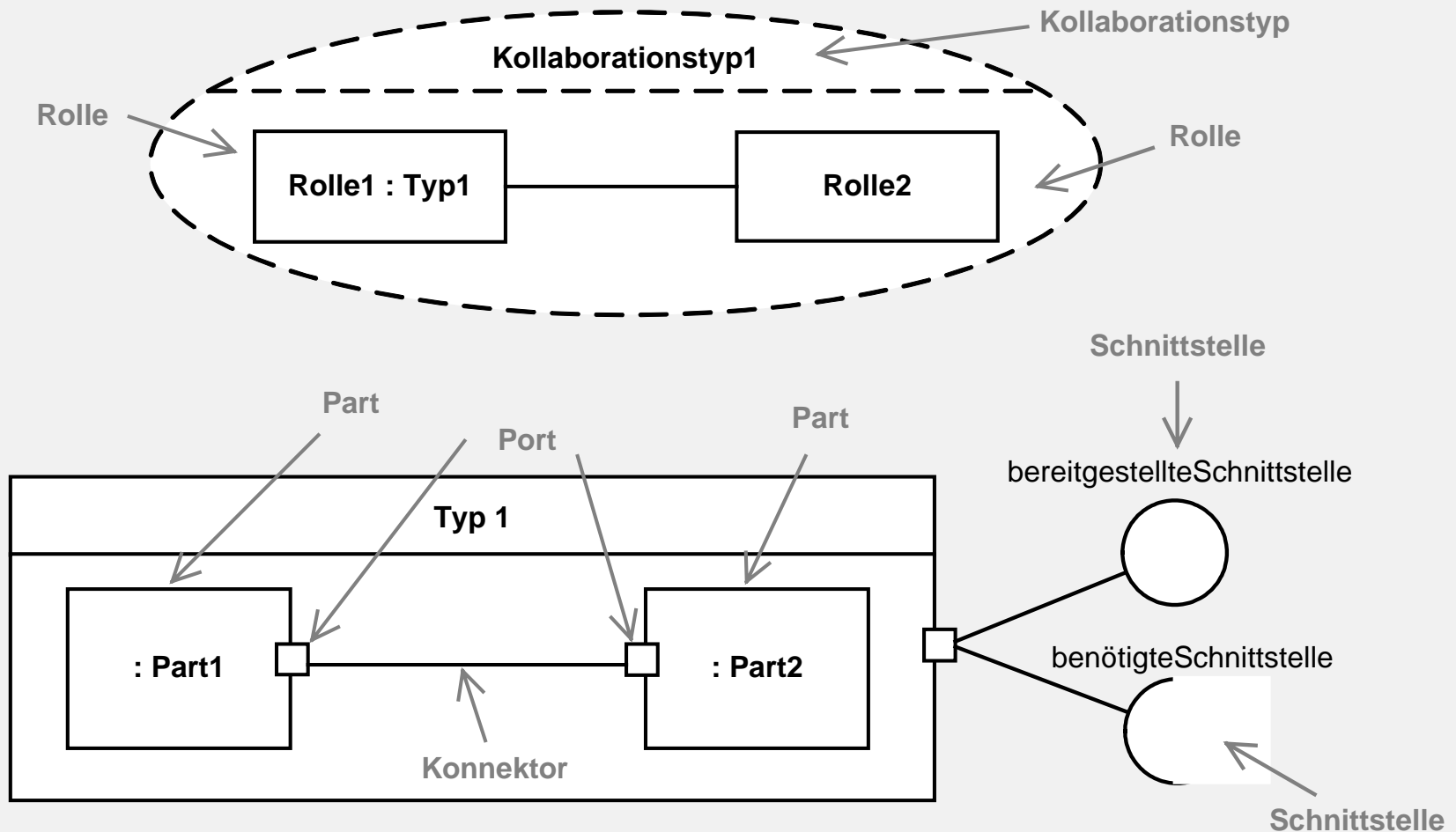
# Classifier in UML 2



## *Classifier:*

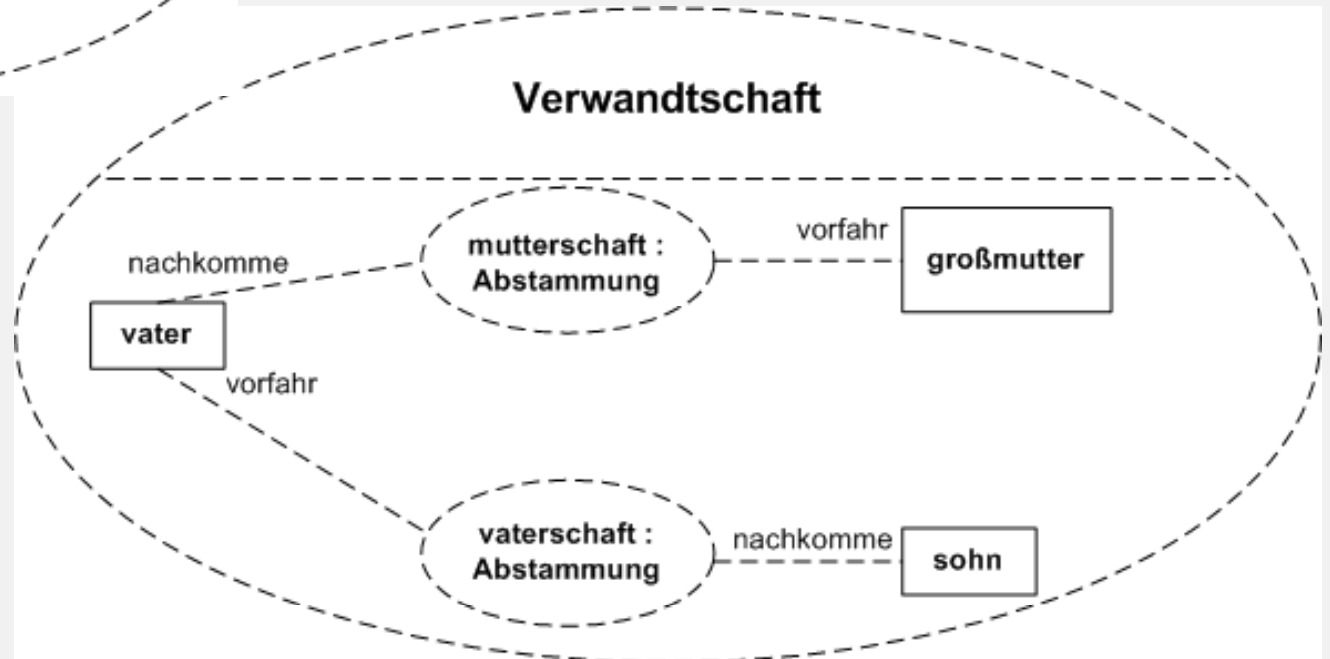
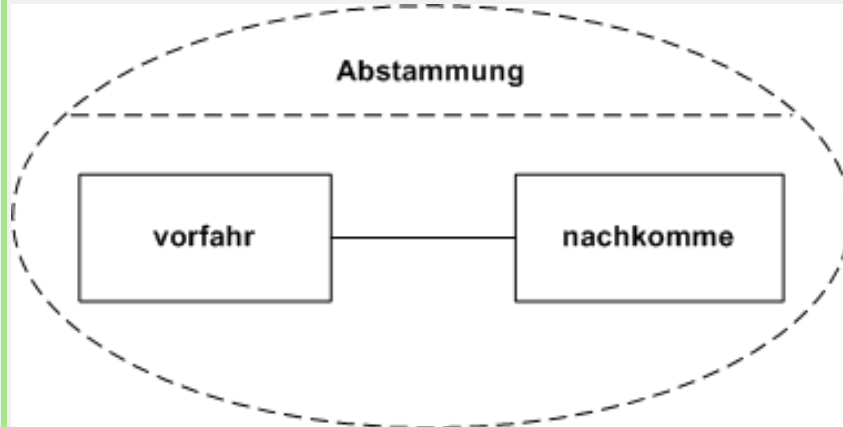
- Kein unmittelbar nutzbares graphisches Konstrukt
- Abstraktion der *Klasse*
- Klassifizierende Instanz einer Reihe konkreter Ausprägungen, unabhängig von deren Detailrealisierung

# Architekturdiagramme der UML 2: Das Kompositionsstrukturdiagramm



Zeigt die interne Struktur eines Classifiers sowie seine Interaktionspunkte mit anderen Architekturkomponenten.

# Architekturdiagramme der UML 2: Das Kompositionsstrukturdiagramm



*Kollaboration:*  
Zeigt das Zusammenspiel  
von Operationen oder Classifiern

# Architekturdiagramme der UML 2: Das Kompositionsstrukturdiagramm

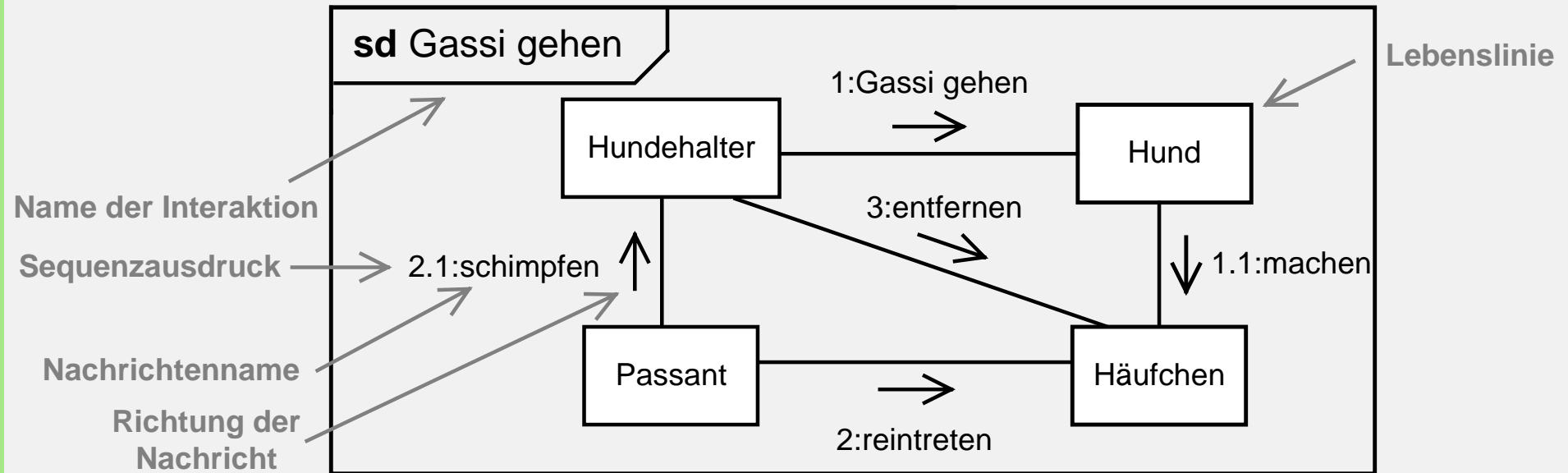


*Kollaboration:*  
Zeigt das Zusammenspiel  
von Operationen oder Classifiern

# Architekturdiagramme der UML 2: Das Kompositionsstrukturdiagramm

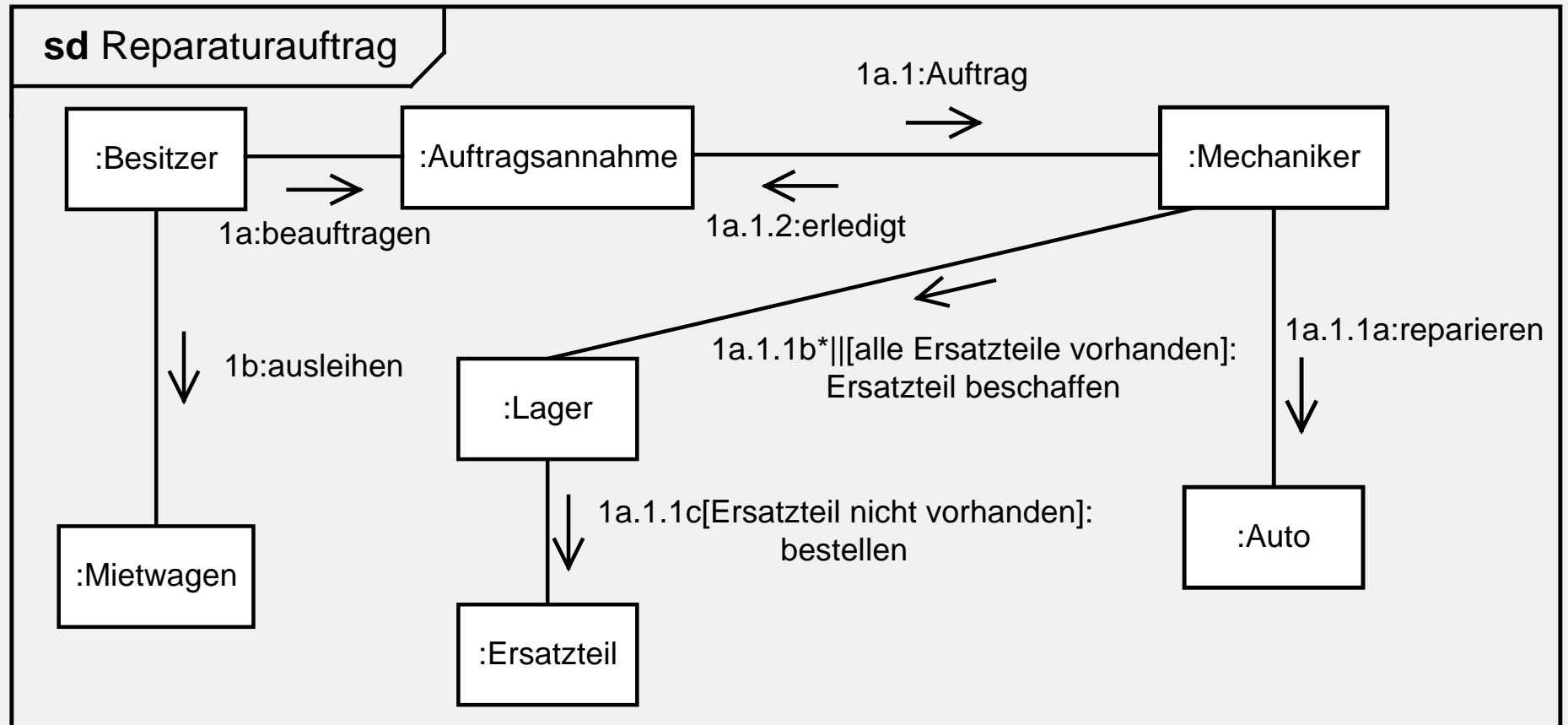
- Verwendung: Veranschaulichung der Architektur.
- Besonders sinnvoll bei der Beschreibung (des Einsatzes von) Entwurfsmustern.
  - Darstellung durch Kollaboration
  - Verwendung durch `occurrence` dokumentieren
- Jeden Architektur-aspekt durch ein Kompositionsstrukturdiagramm abbilden.
- Ausdetaillierung der *Ports* hinsichtlich zugelassener Delegationsmöglichkeiten.

# Architekturdiagramme der UML 2: Das Kommunikationsdiagramm



Zeigt Interaktion zwischen  
Architekturkomponenten (Lebenslinien).

# Architekturdiagramme der UML 2: Das Kommunikationsdiagramm



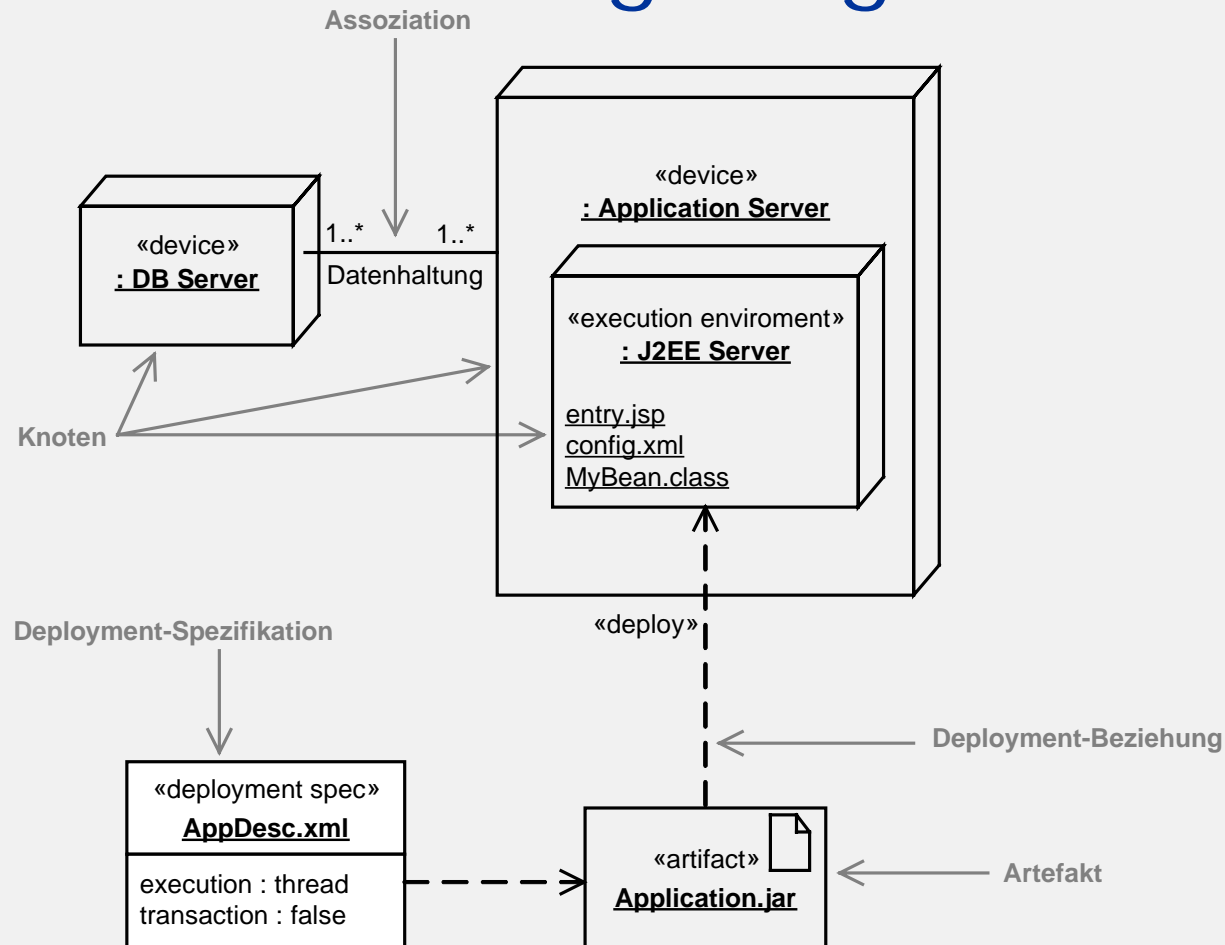
Zeigt Interaktion zwischen  
Architekturkomponenten (Lebenslinien).



# Architekturdiagramme der UML 2: Das Kommunikationsdiagramm

- Gleichberechtigte Sicht auf alle beteiligten Kommunikationspartner.
- Möglichst keine, durch Bedingungen unterschiedene, Ablaufvarianten darstellen, sondern separate Kommunikationsdiagramme erstellen.
- Möglichst Sequenz-agnostisch gestalten.
- Verwendung nebenläufiger und iterativer Nachrichten verwirrt unter Umständen.

# Architekturdiagramme der UML 2: Das Verteilungsdiagramm

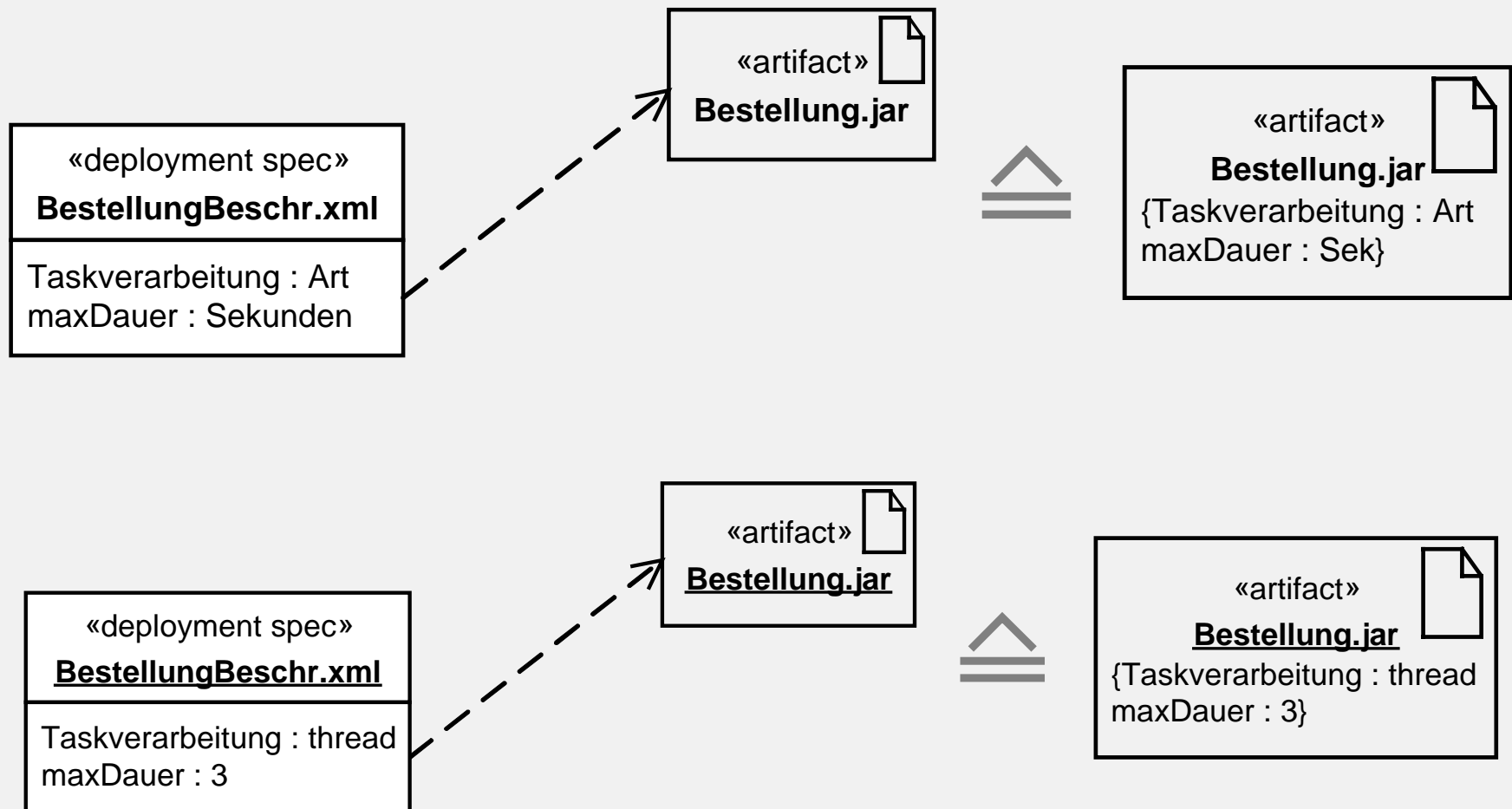


Zeigt über ein Netzwerk verbundene Hardware- oder Softwarekomponenten (Rechner, Ausführungsumgebung), die zur Ausführung benötigt werden.

# Architekturdiagramme der UML 2: Das Verteilungsdiagramm

- Angesichts der offen konkurrierenden Notationen (etwa für Einsatzspezifikationen) muß im Projekt zwingend eine Auswahl getroffen werden.
- Sinnvolle Beschränkung auf den relevanten Ausschnitt.

# Architekturdiagramme der UML 2: Das Verteilungsdiagramm



Alternativnotationen zur Darstellung von Einsatzfallspezifikationen

# Architekturdiagramme der UML 2

- **Kompositionsstrukturdiagramm:**  
„Wie sind die einzelnen Architekturkomponenten strukturiert und wie spielen sie zusammen?“
- **Komponentendiagramm:**  
„Welche Teile einer komplexen Struktur arbeiten wie zusammen, um eine bestimmte Funktion zu erfüllen?“
- **Verteilungsdiagramm:**  
„Wie werden die Komponenten des Systems zur Laufzeit wohin verteilt?“

# Architekturprinzipien der UML 2

- **Modularität**  
Starke Kohäsion und lose Kopplung bei der Bildung der (Metamodell-)Pakete und Organisation der (Meta-)Klassen
- **Schichtung**  
Organisation von Konzepten verschiedener Abstraktionsebenen in Pakete und Schichten
- **Partitionierung**  
Logische Organisation von Konzepten derselben Abstraktionsebene.
- **Erweiterbarkeit**  
Profile und Verwendung des UML-Sprachkerns (tatsächlich MOF) in eigenen Sprachen
- **Wiederverwendbarkeit**  
Konstrukt(wieder-)nutzung im UML-Metamodell, im Metamdetamodell der MOF und anderen Metamodellen (z.B. CWM)

jeckle.de - Mozilla

File Edit View Go Bookmarks Tools Window Help

Back Forward Reload Stop http://www.jeckle.de/ Search Print

Unified Modeling Language (UML)  
eXtensible Markup Language (XML)  
XML Metadata Interchange (XMI)  
Web Services  
XML Acronym Demystifier Project  
XML-Strategie

Vorträge und Publikationen  
Vorlesungen  
Diplomarbeiten  
GOOAL.net  
XML-Arbeitskreis  
Software & Downloads

Web Services Workshop *WS-RSD'02*  
ICWS'03-Europe  
Web Services @ Berliner XML-Tage  
Internet & Search Engines  
Mersennesche Primzahlen  
Feedback  
Rotkreuz Mitgliederverwaltung

Mario Jeckle ...  
Dialog ...  
Über diese Seiten ...  
suchen ...  
SiteMap  
RSS Newsfeed XML  
Was gibt's hier Neues?

**jeckle.de**  
Diese Folien ...  
sowie weitere Informationen zum Thema

Transferring data from www.jeckle.de...