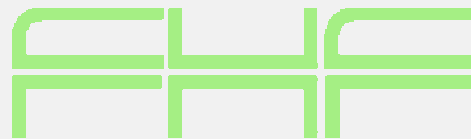


UML

Unified Modeling Language



Prof. Mario Jeckle

Fachhochschule Furtwangen

mario@jeckle.de

<http://www.jeckle.de>

Vorlesungsinhalte

1 Einführung

- 1.1 Herkunft und Historie der UML
- 1.2 System- und Softwareentwicklungsprozesse

2 Strukturdiagramme

- 2.1 Klassendiagramm
- 2.2 Paketdiagramm
- 2.3 Objektdiagramm
- 2.4 Kompositionsstrukturdiagramm
- 2.5 Komponentendiagramm
- 2.6 Verteilungsdiagramm

Vorlesungsinhalte

3 Verhaltensdiagramme

- 3.1 Basiskonzepte
- 3.2 Use-Case-Diagramm
- 3.3 Aktivitätsdiagramm
- 3.4 Zustandsautomat
- 3.5 Sequenzdiagramm
- 3.6 Kommunikationsdiagramm
- 3.7 Timing-Diagramm
- 3.8 Interaktionsübersichtsdiagramm

4 Weiterführende Themen

- 4.1 UML-Profile
- 4.2 Diagrammaustausch
- 4.3 XML Metadata Interchange (XMI)

Vorlesungsziele

- Der Student soll nach der Veranstaltung ...
 - die zentralen Grundkonzepte der UML kennen und beherrschen
 - in der Lage sein UML-Modelle erstellen und verstehen zu können
 - die Grenzen der UML im praktischen Einsatz kennen
- Ergänzend zur Veranstaltung wird auf freiwilliger Basis die Teilnahme an der offiziellen OMG-Zertifizierung zum *UML Professional Fundamental* angeboten

OMG-Certified UML Professional Fundamental

Wofür?	Sicherstellung, dass UML Anwender, Trainer, Berater und Tool- Hersteller einheitliche Grundlagen haben	
Wie?	Der Test wird in Englisch in einem Prometric Test Center durchgeführt und ist ein reiner Sprachtest- <u>kein</u> Methodentest. Der Test dauert 1:45 h.	
Themen?	Allgemeine UML Inhalte, zusätzlich aber auch Bereiche des UML- Metamodells.	
Einteilung?	Klassendiagramme	30%
	Aktivitätsdiagramme	20%
	Interaktionsdiagramme	20%
	Anwendungsfalldiagramme	20%
	Allgemeine Grundlagen	10%
		100%

Was ist eigentlich UML?

Die Unified Modeling Language ist

- eine standardisierte Notationssprache
- um Softwaresysteme und –projekte besser
 - entwerfen
 - analysieren
 - dokumentieren zu können

Aber?

Die Unified Modeling Language ist NICHT

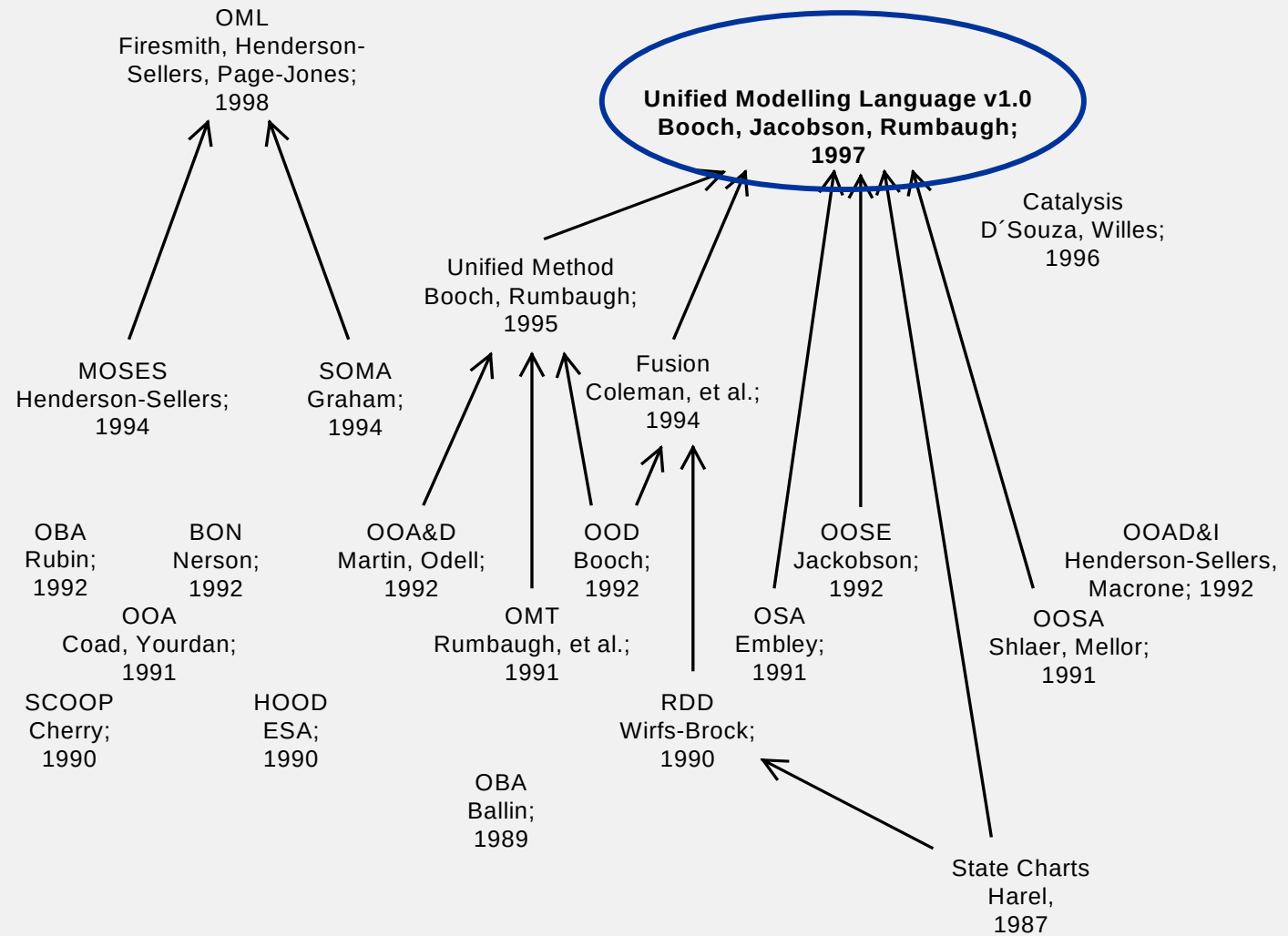
- perfekt
- vollständig
- ohne präzisierende Erweiterungen als Programmiersprache geeignet
- spezialisiert auf genau ein Anwendungsgebiet
- Vollständiger Ersatz für eigene Textbeschreibungen
- Methode oder Vorgehensmodell

Die OMG

Der Standard der UML wird durch die OMG betrieben:

- Object Management Group
- 1989 gegründet
- überwacht herstellerneutrale Industriestandards
- Gremium mit ca. 800 Mitgliedern, z.B.:
 - IBM
 - HP
 - DaimlerChrysler
 - Microsoft
 - Oracle
 - ...

„Die Methodenkriege der Softwareentwicklung“



Entwicklung der UML durch „Methodenkriege der Softwareentwicklung“
Anfang 1990

Ergebnisse

- **Unified Method (v0.8):**
Versuch Methode und Notation zu vereinheitlichen durch J. Rumbaugh und G. Booch.
- **Unified Modeling Language (v0.9ff):**
Reduktion der Vereinheitlichungsbemühungen auf Notation.
I. Jacobson unterstützt Bemühungen.
Integration von statischer und dynamischer Modellierung.
- **UML 1.0:**
Beginn der Standardisierungsaktivitäten
- **UML 2.0:**
Erste große Überarbeitung

Probleme der 1.x Versionen

- Zu groß
- Zu komplex
- Nicht schnell erlernbar
- Schnelle Änderungen bei Softwareentwicklung
- Neue Programmiersprachen entstanden, alte verschwanden
- Lösungsansatz: Weiterentwicklung zur UML 2

Der „Second System Effect“

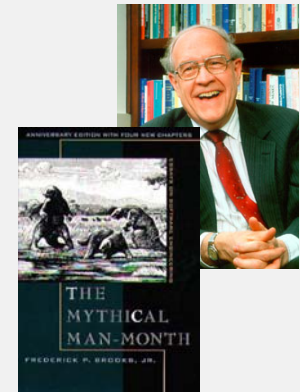
- Probleme bei der Entwicklung von UML 2:
Vermeidung des *Second System Effect*

Gefahr der Umfangssteigerung durch

- Verwendung bisheriger Komponenten
- Hinzunahme neuer Ideen und Entwicklungen

Lösung:

- Komplexitätsreduktion des bestehenden Sprachumfanges
- Weiterentwicklung gemäß Anwendervorgaben



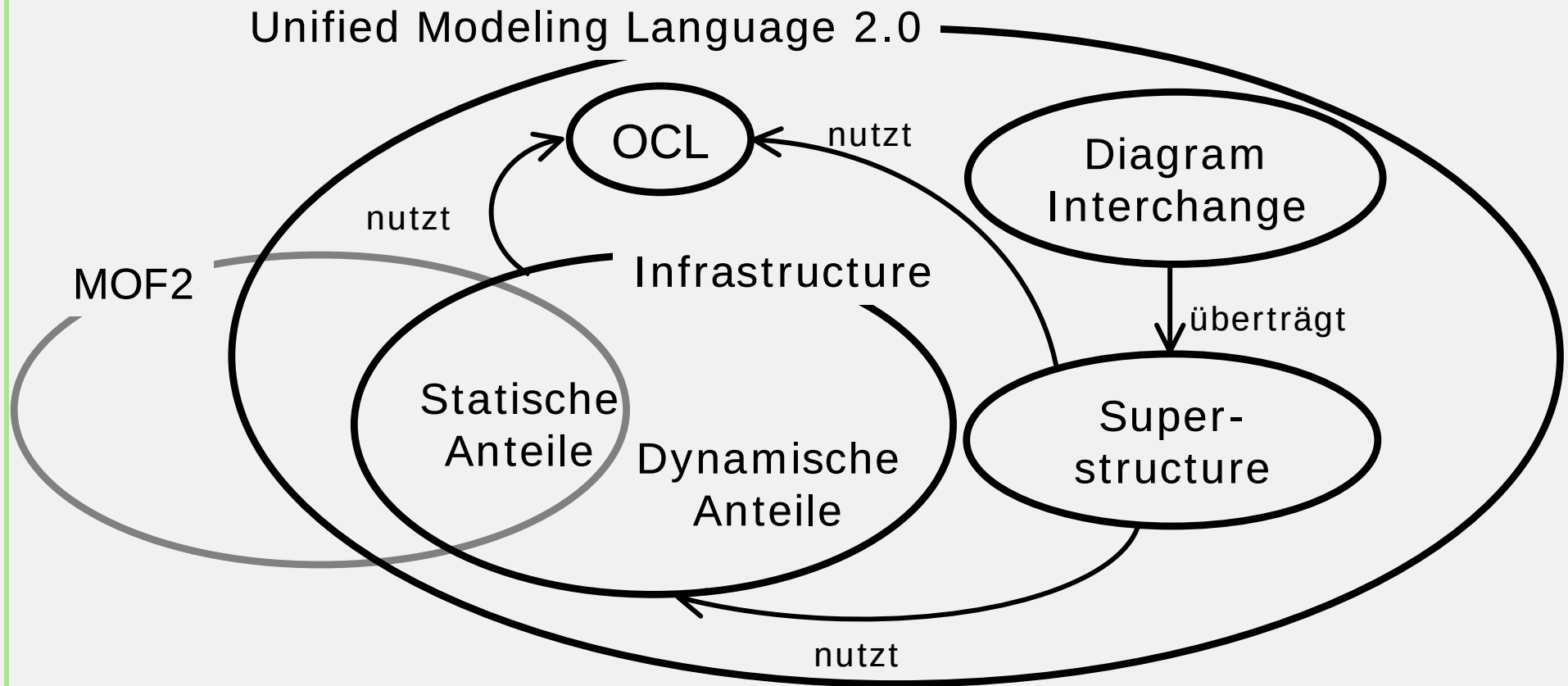
Vorschläge für UML 2

U2 Partners: Konsortium gebildet durch Hersteller und Anwender.
Darunter: Alcatel, Ericsson, HP, IONA, Oracle, SOFTEAM, Unisys, Rational, IBM, Together u.v.m.

Ergebnisse: UML 2-Sprachstruktur

- **Infrastructure:**
Grundlegende Sprachkonstrukte und Basisarchitektur
- **Superstructure:**
Diagrammnotation und Semantik
- **Object Constraint Language:**
In UML formalisiertes Metamodell der OCL.
OCL-Beschreibung vieler UML-Konstrukte.
- **Diagram Interchange:**
Metamodell zur Darstellung von visuellen Modellrepräsentationen.
Grundlage für Werkzeugübergreifenden Austausch.

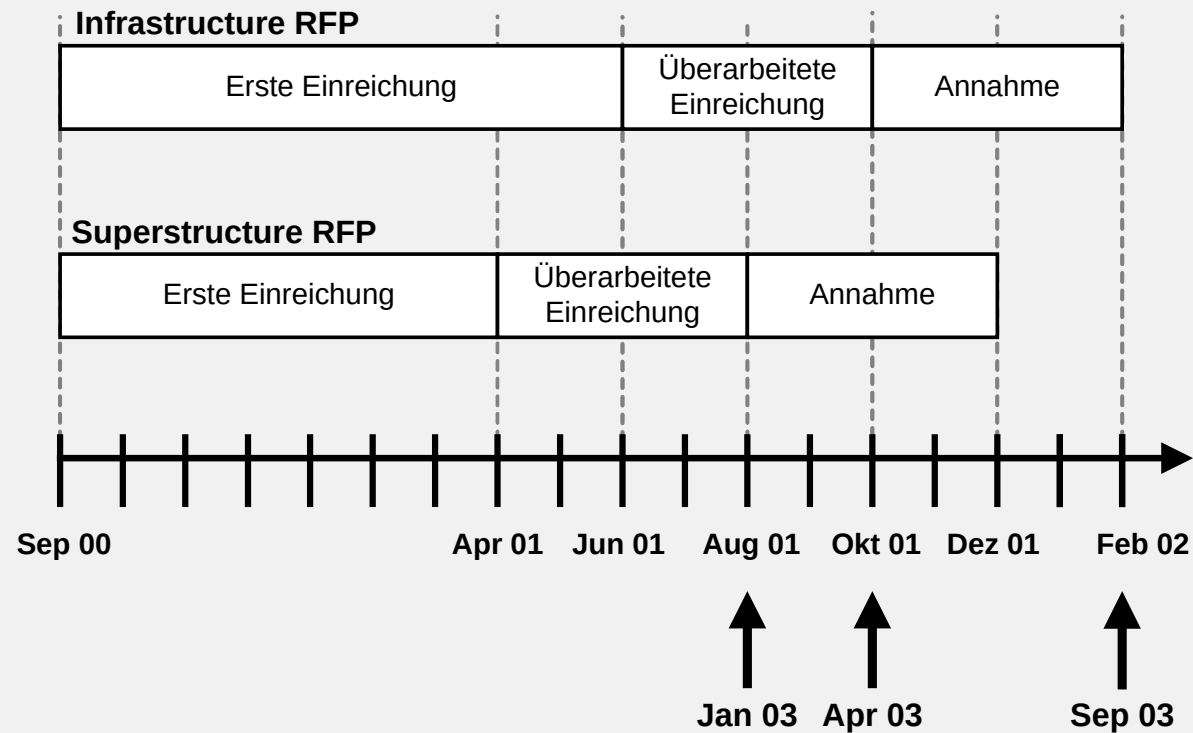
Sprachstruktur der UML 2



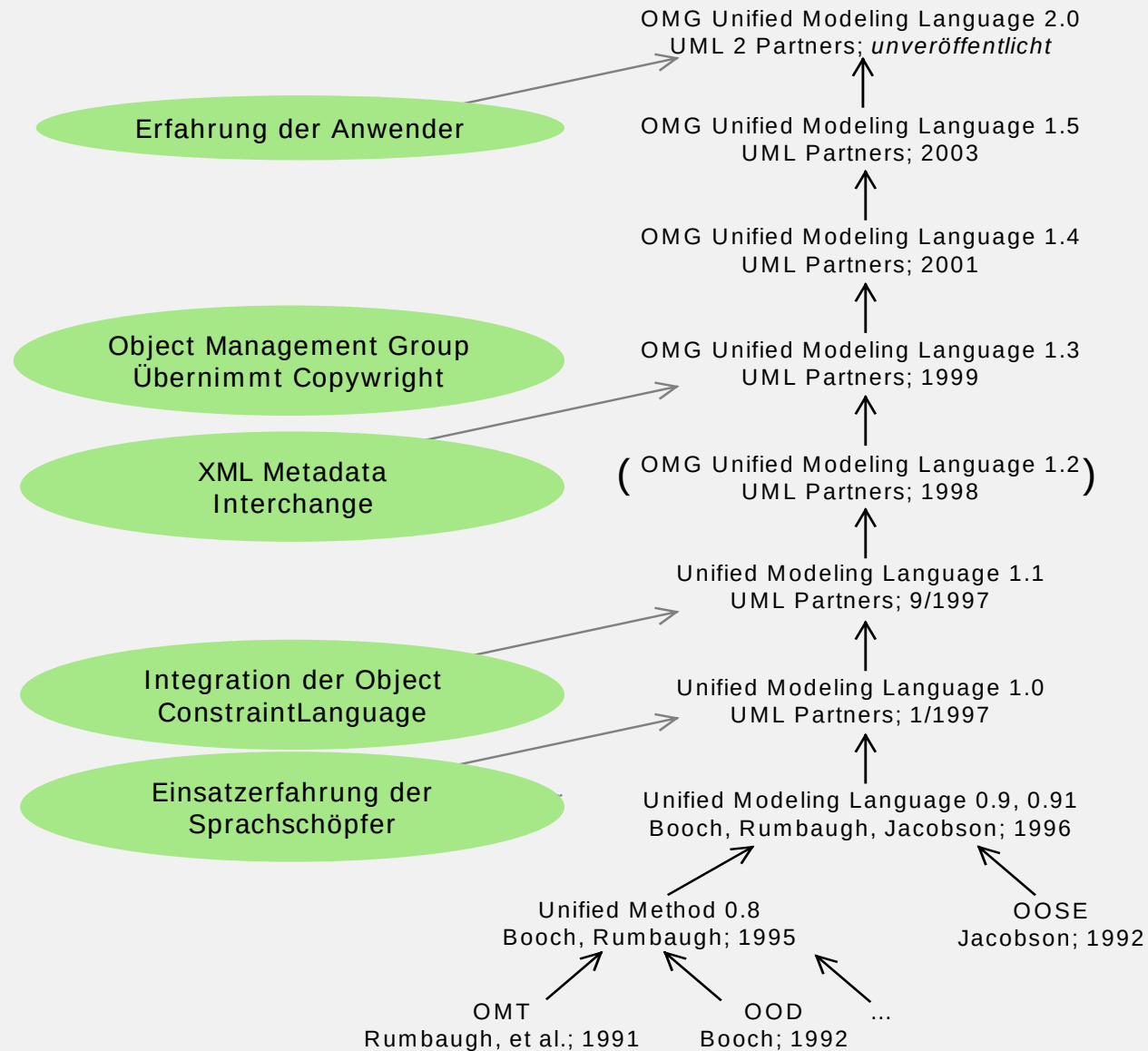
Anmerkung:
Metaobject Facility ist das Meta-Metamodell der UML

Der zeitliche Ablauf

UML 2 Strukturdokumente im Ablaufplan



Entstehungsübersicht UML 2



UML 2 -- Ja oder Nein

Es gibt mehrere Anhaltspunkte, ob ein Umstieg auf UML 2.0 lohnt

- Häufige Verwendung von Zustandsautomaten und Sequenzdiagrammen zur Systemverhaltensmodellierung
- Darstellung von Kommunikation der Systemkomponenten und des Zeitverhaltens ist gewünscht
- Aktivitätsdiagramme der bisherigen UML sind zu eingeschränkt
- Code oder Architekturen werden mit MDA-Ansatz generiert

UML 2 Diagrammtypen

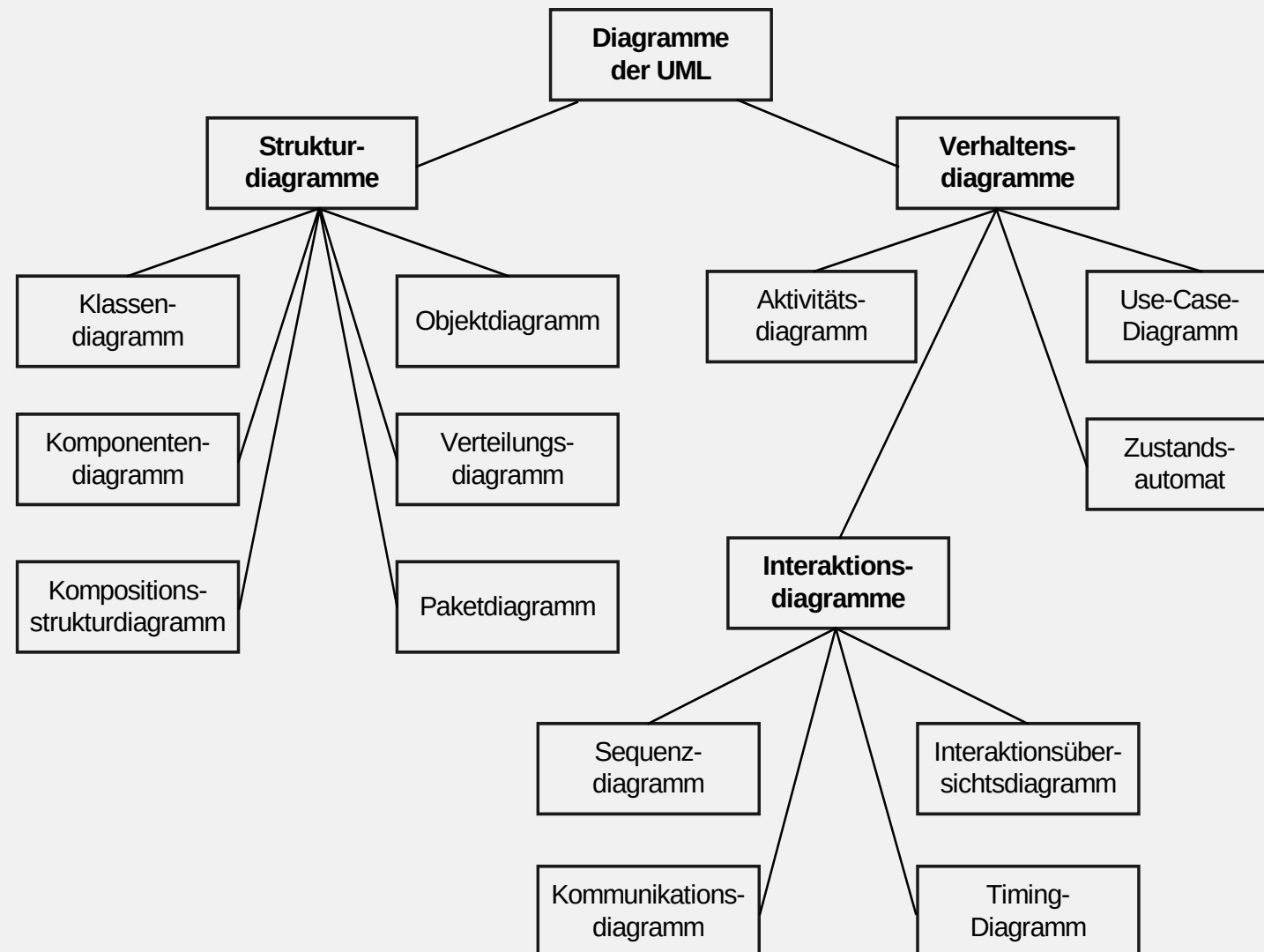
UML 2 bietet dem Benutzer 13 Diagrammtypen:

- Sechs *Strukturdiagramme* für die Modellierung der Statik des Systems
- Sieben *Verhaltensdiagramme*, von denen wiederum vier als *Interaktionsdiagramme* zusammengefasst werden

Neu in UML 2:

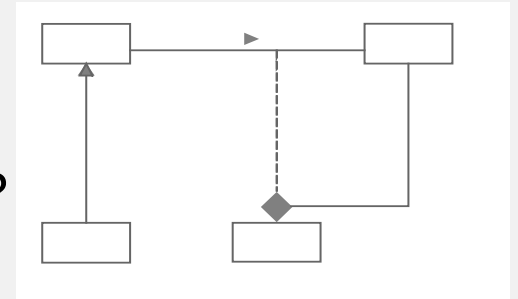
- Timing-Diagramm
- Interaktionsübersichtsdiagramm
- Kompositionsstrukturdiagramm

Übersicht

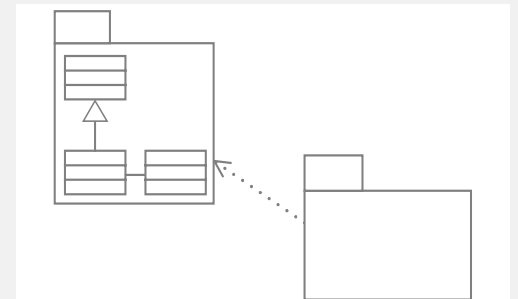


Kurzüberblick Funktionen

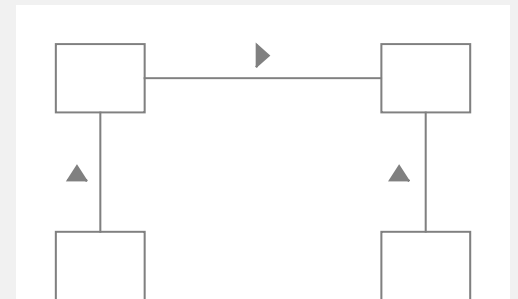
Klassendiagramm: Aus welchen Klassen besteht mein System und wie stehen diese in Beziehung?



Paketdiagramm: Wie kann ich mein Modell so schneiden, dass ich den Überblick bewahre?

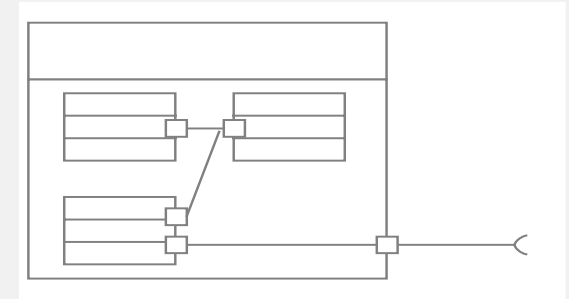


Objektdiagramm: Welche innere Struktur hat mein System zu einem festgelegten Zeitpunkt der Laufzeit?

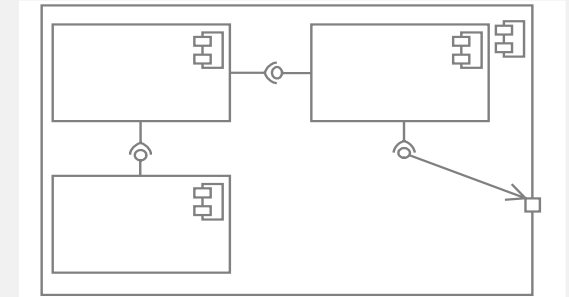


Kurzüberblick Funktionen

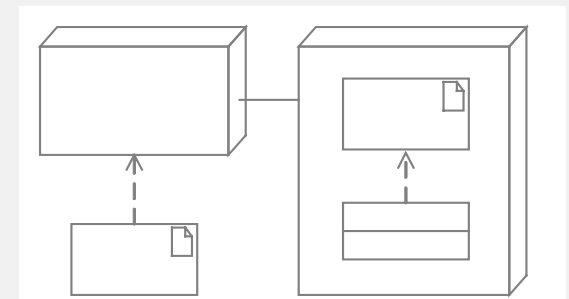
Kompositions- : Wie sieht es in der Klasse,
Strukturdiagramm in der Komponente aus?



Komponenten- : Wie werden Klassen zu
diagramm wiederverwendbaren,
verwaltbaren
Komponenten?

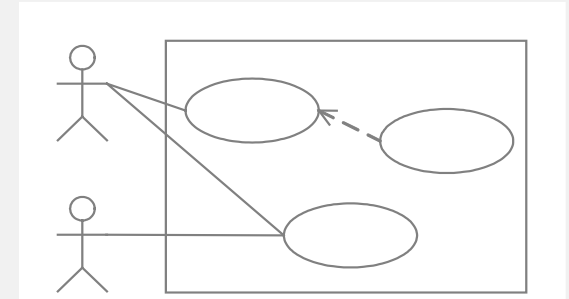


Verteilungsdiagramm: Wie werden Komponenten
wohin verteilt?

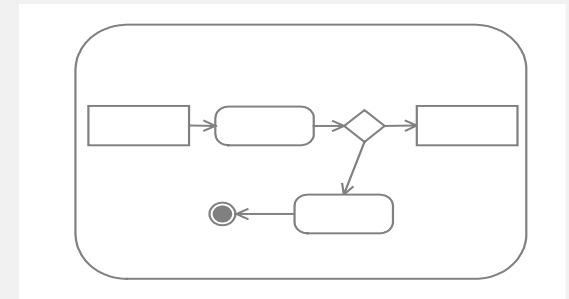


Kurzüberblick Funktionen

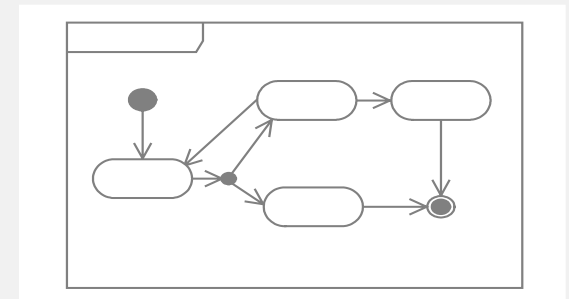
Use-Case-Diagramm: Was leistet mein System für meine Umwelt?



Aktivitätsdiagramm: Wie läuft ein bestimmter flussorientierter Prozess ab?

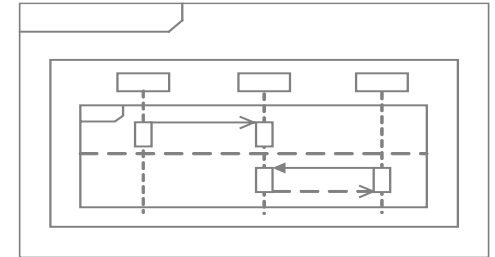


Zustandsautomat: Welche Zustände kann ein Objekt, Interface... bei welchen Ereignissen annehmen?

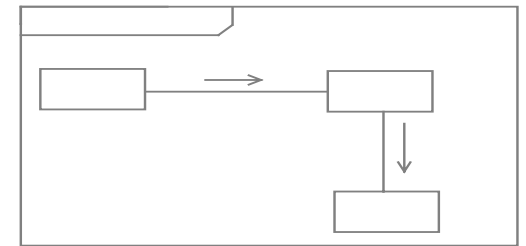


Kurzüberblick Funktionen

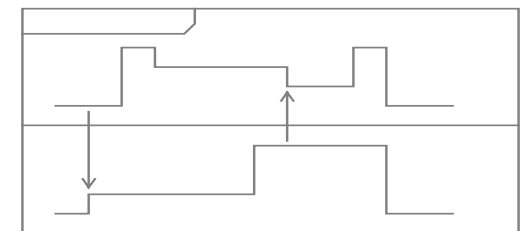
Sequenzdiagramm: Wer tauscht wann mit wem welche Info aus?



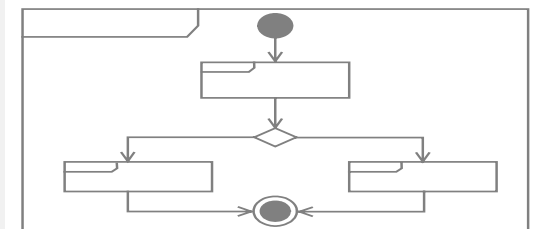
Kommunikations-: Wer kommuniziert mit wem?



Timing-Diagramm: Wann befindet sich wer in welchem Zustand?



Interaktions- : Wann läuft welche übersichtsdiagramm Interaktion ab?



Compliance Levels

Die UML wird in drei Erfüllungsebenen eingeteilt:

- Basic
- Intermediate
- Complete

Ziel:

- Keine Überforderung des Benutzers durch vollen Sprachumfang
- Werkzeughersteller können schrittweise und den Bedürfnissen entsprechend selektiv UML einsetzen
- Einteilung analog zu SQL vorgenommen

Compliance Levels

Die Erfüllungsebenen werden zusätzlich in Erfüllungspunkte unterteilt.

Ein Erfüllungspunkt ist eine inhaltlich gruppierte Sammlung von Notationselementen.

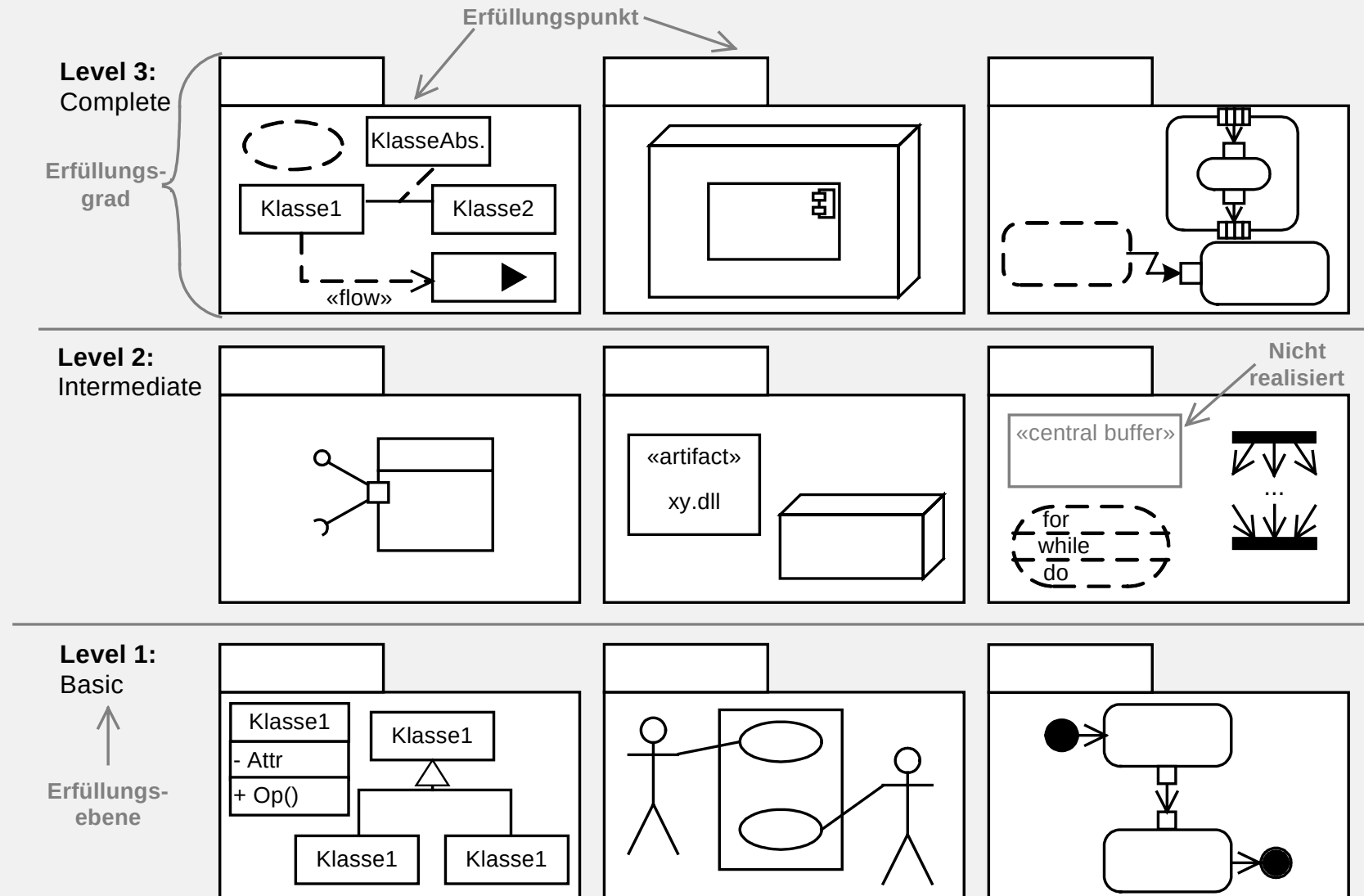
- Werkzeughersteller kann Umfang der Unterstützung einzelner Diagrammtypen auf Kundenwünsche abstimmen.
- Werkzeughersteller muss zusätzlich zu Erfüllungspunkten auch den Grad der Erfüllung angeben (*Compliance Option*)
- UML-Elemente können so in unterschiedlich komplexer Darstellung verwendet werden

Compliance Levels

Erfüllungsgrade werden in 4 Stufen eingeteilt:

- **Nicht erfüllt:** Notationselemente des Erfüllungspunktes werden nicht unterstützt
- **Teilweise erfüllt:** nur ein Teil der Notationselemente wird unterstützt
- **Vollständig erfüllt:** alle Notationselemente werden standardkonform unterstützt
- **Austauschbar:** zusätzlich erfüllt das Tool das Standardaustauschformat für Notationselemente

Compliance Levels -- Überblick



Aber bis dahin...

...ist es noch ein langer Weg!

