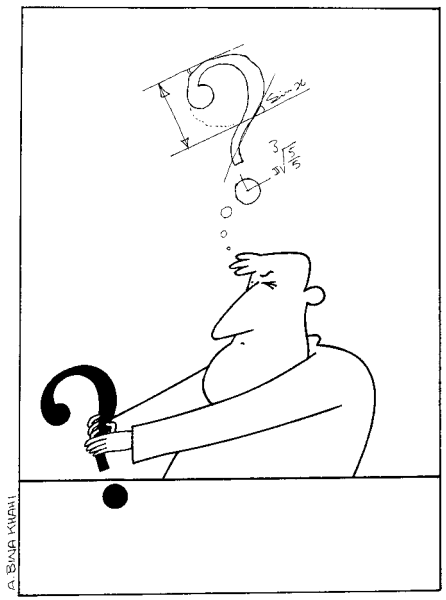
open minded...
SOPHIST GROUP

UML 2 – Ballast oder Befreiung?

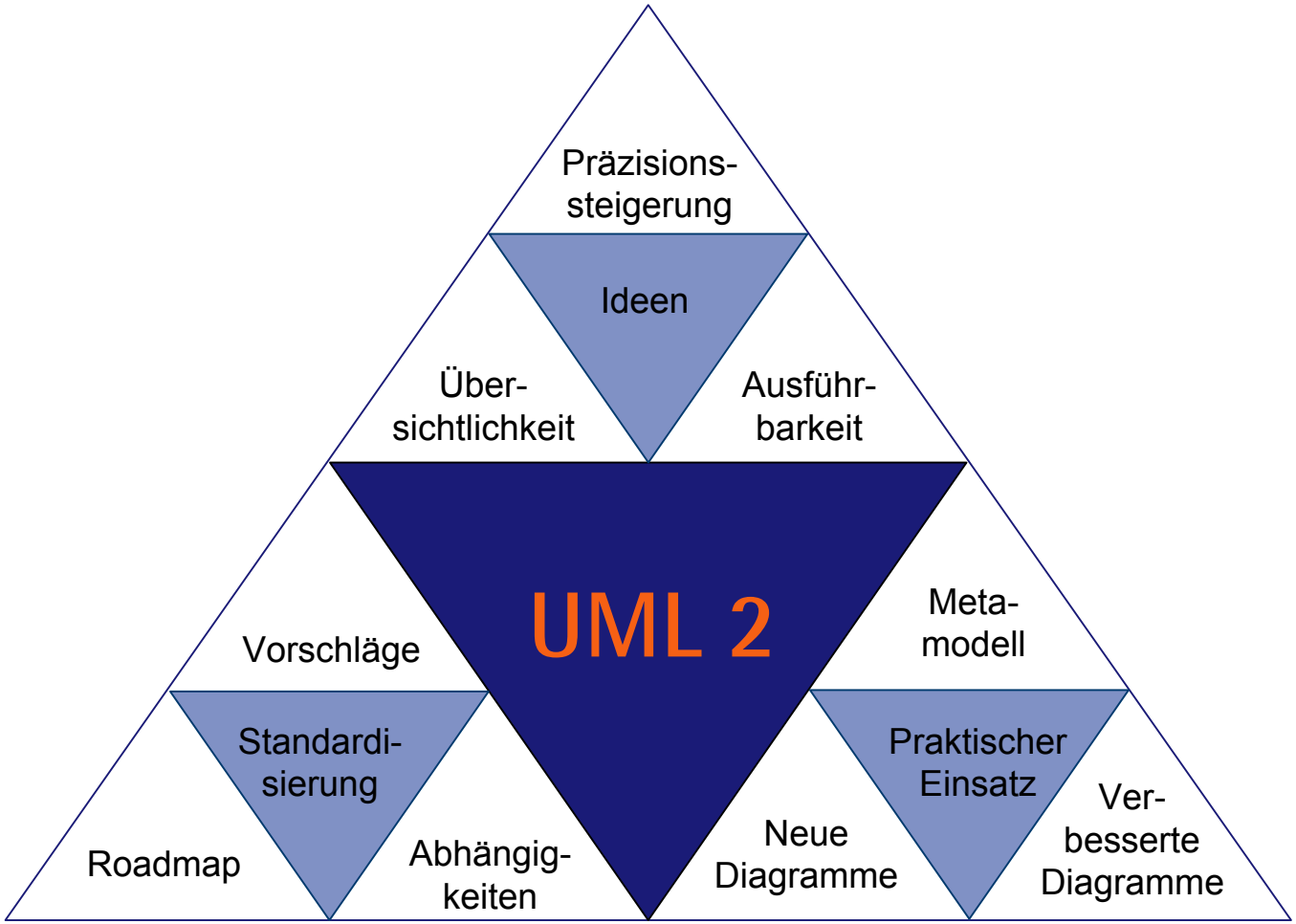
von Chris Rupp, SOPHIST GROUP

chris.rupp@sophist.de



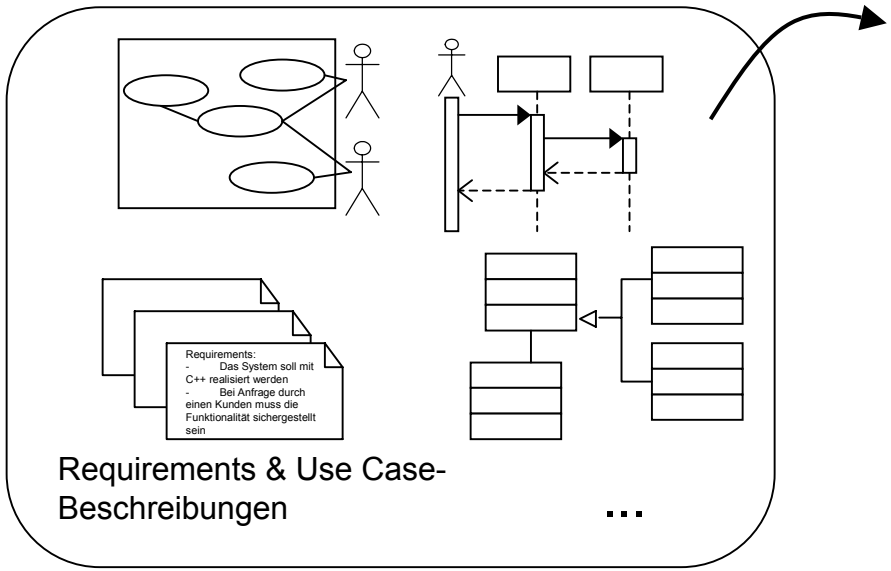
Es gibt Menschen, die, wenn sie das Licht am Ende des Tunnels sehen, ein neues Stück Tunnel kaufen.

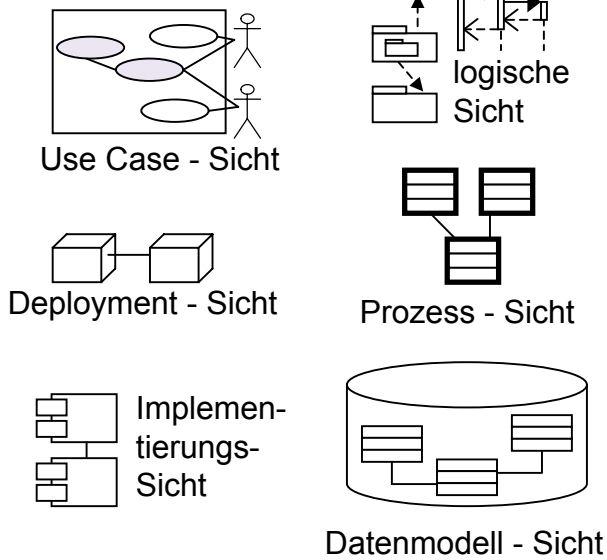
Johannes Rau


Präzisions-
steigerung
Ideen
Über-
sichtlichkeit
Ausführ-
barkeit
UML 2
Vorschläge
Meta-
modell
Standardi-
sierung
Praktischer
Einsatz
Roadmap
Abhängig-
keiten
Neue
Diagramme
Ver-
besserte
Diagramme

Was ist die UML?

Wunsch des Kunden

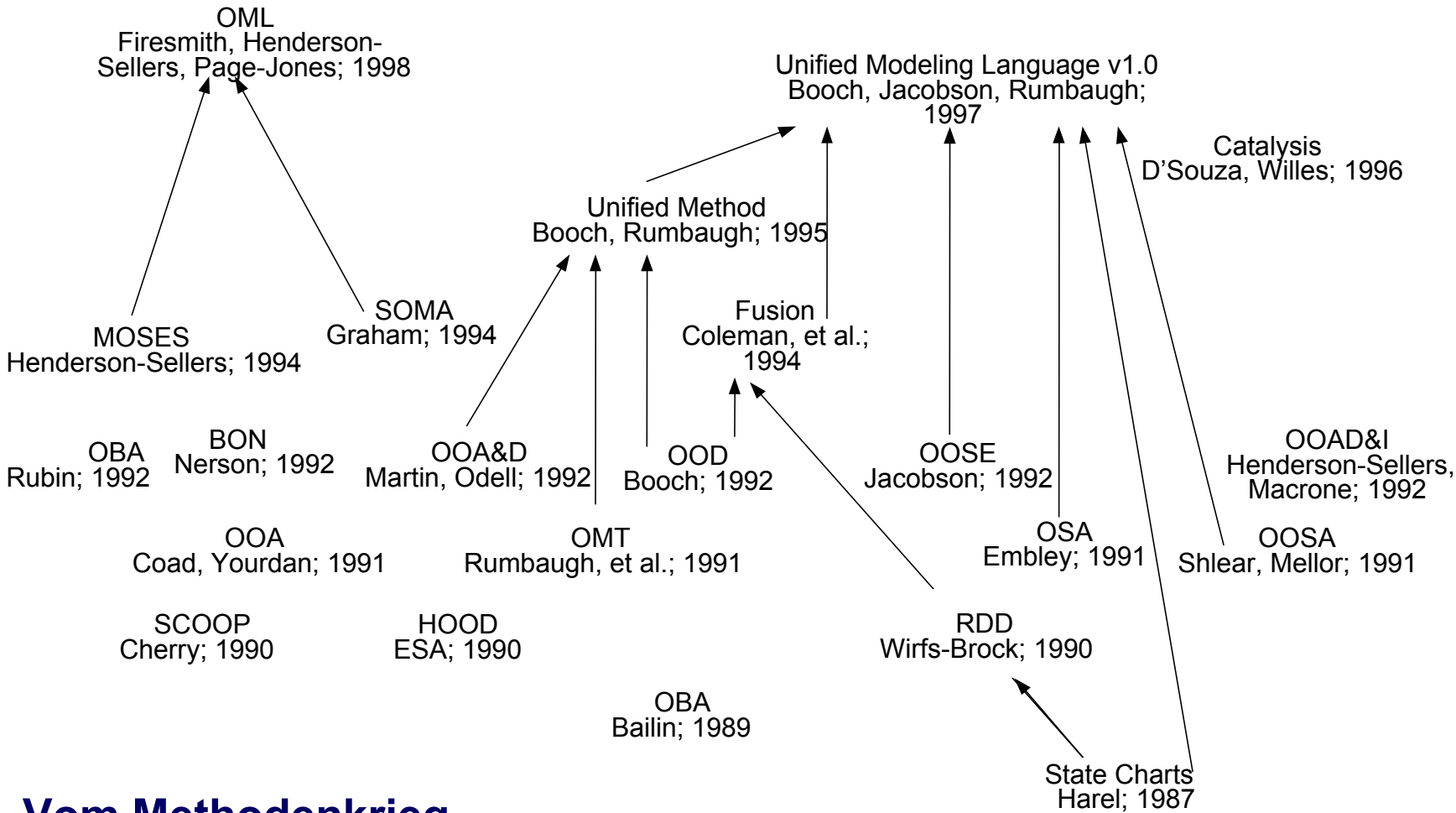




- Notation für Modell des Systems
- zeigt statische und dynamische Aspekte
- Nicht jede Sicht in jedem System sinnvoll

UML ...

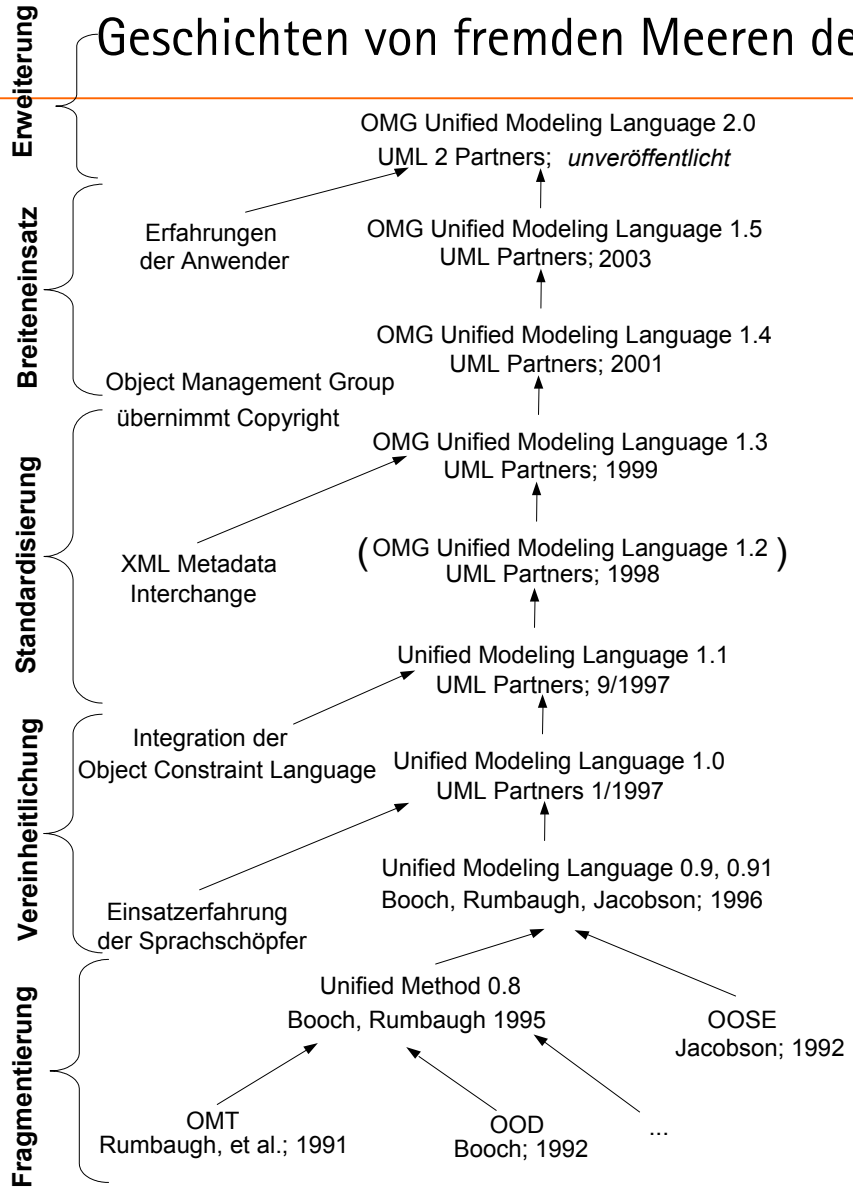
Geschichten von fremden Meeren der Standardisierung



Vom Methodenkrieg ...

UML ...

Geschichten von fremden Meeren der Standardisierung



Viele arbeiten für alle

Einige arbeiten für andere

Wenige arbeiten für einige

... zum Standardisierungskrieg

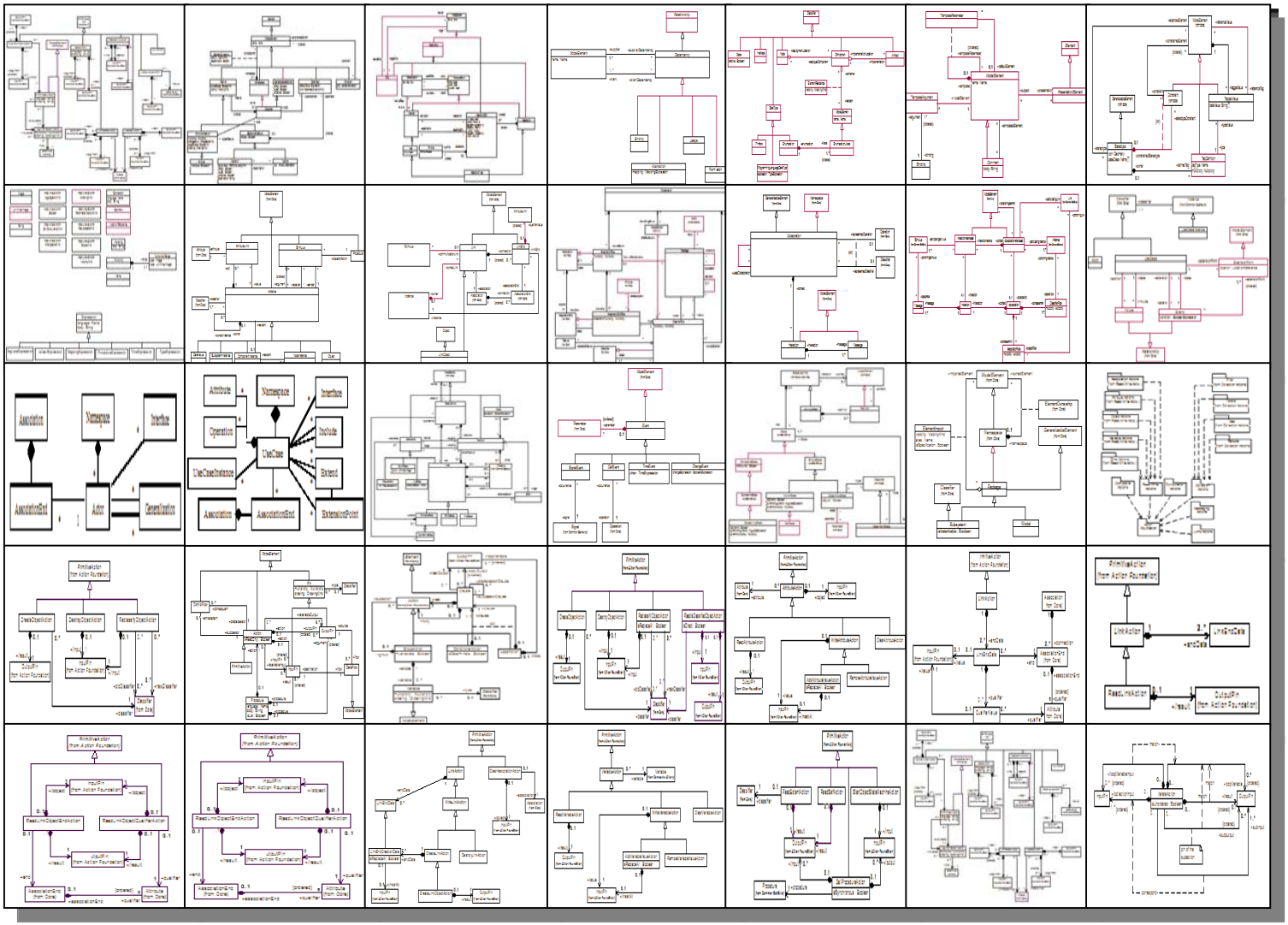
UML 2

... Warum eine neue Version?

- Evolution
  - Der Markt hat sich bewegt...
    - Neue Programmiersprachen (z.B. C#, Python, PHP)
    - Neue Anwendungsdomänen (z.B. Serverprog., RTE-Systeme)
- Erfahrung
  - Für einige Einsatzgebiete bietet UML v1.x ...
    - Manchmal zu wenig
    - Manchmal zu viele Konstrukte
- Eliminierung
  - Sprachen und Konzepte verschwinden aus der UML
    - Einige Programmiersprachen verschwinden (z. B. C++)
    - Einige früher als modellierungsnah eingestufte Konzepte entwickeln sich inzwischen getrennt von UML weiter (z. B. Entwicklungsprozesse, Codegenerierung)

UML ...

Ein Leiden am Second System Syndrom



UML 2
Die Ziele

- Übersichtlichkeit
  - Weniger graphische Modellkonstrukte
  - Weniger Basiskonzepte + deren Wiederverwendung
- Präzisionssteigerung
  - Reformulierung des Meta-Modells
  - Weitestgehende OCL-Verwendung
- Ausführbarkeit
  - Erweiterte Zustandsmaschinen
  - Stärkere Beziehungen zwischen statischen und dynamischen Diagrammen
  - Integration erprobter Konzepte außerhalb der UML

UML 2

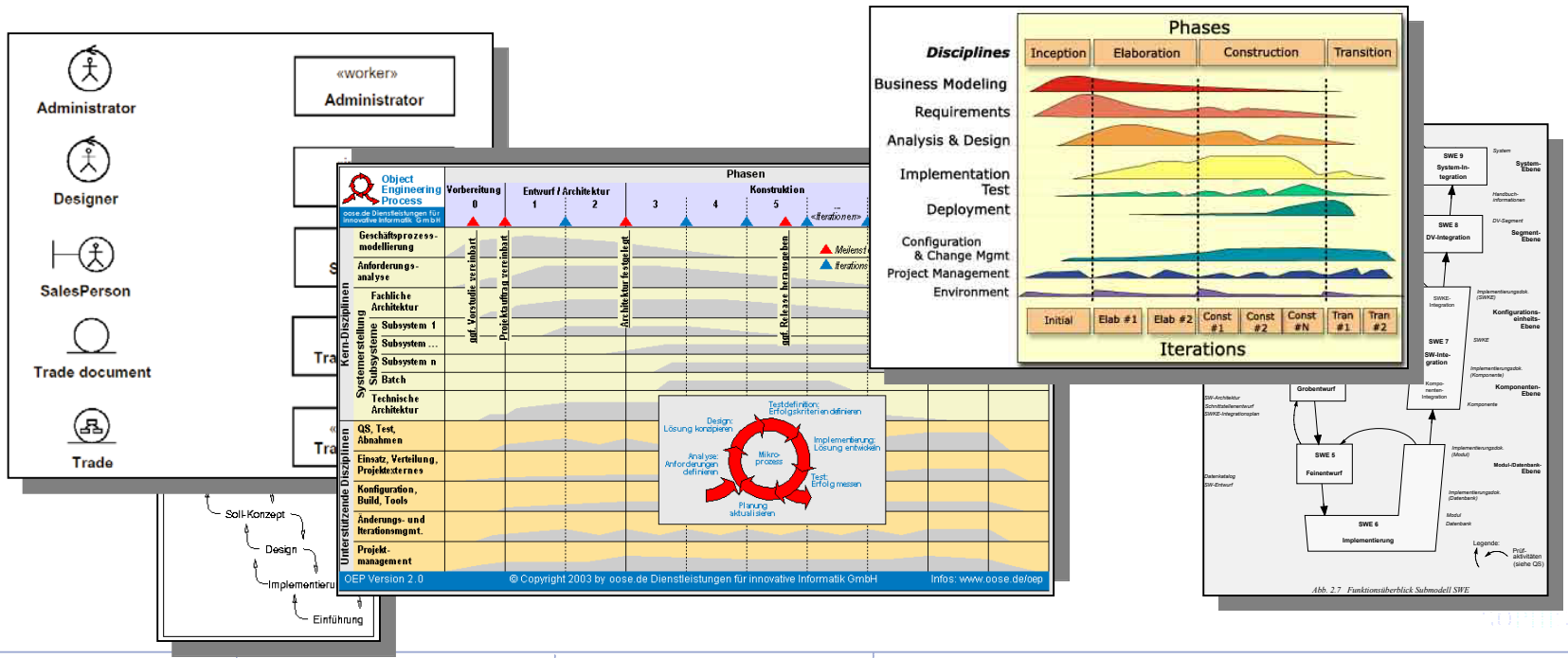
Verrentung existierender Modellelemente

1. Durch UML-Werkzeuge nicht implementierte Sprachanteile
2. Durch OO-Methoden unberücksichtigte Sprachelemente
3. Programmiersprachen-spezifische Sprachelemente
4. Inpräzise UML-Sprachelemente

UML 2

Verrentung existierender Modellelemente

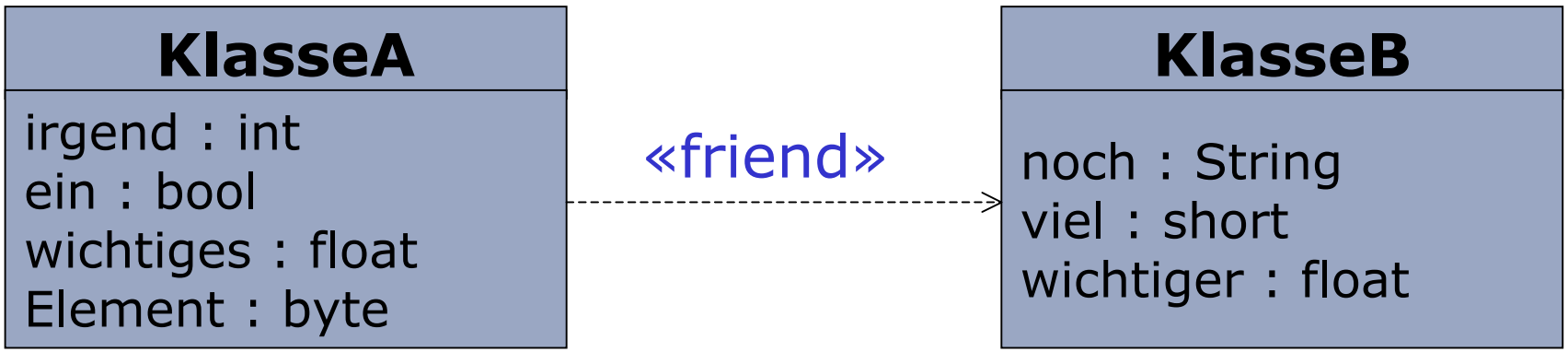
1. Durch UML-Werkzeuge nicht implementierte Sprachanteile
2. Durch OO-Methoden unberücksichtigte Sprachelemente
3. Programmiersprachen-spezifische Sprachelemente
4. Inpräzise UML-Sprachelemente



UML 2

Verrentung existierender Modellelemente

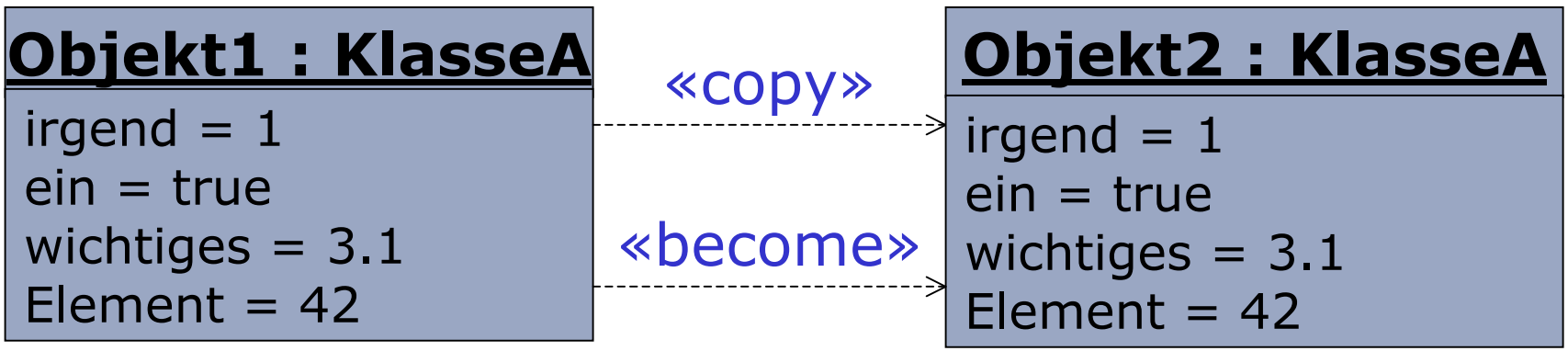
1. Durch UML-Werkzeuge nicht implementierte Sprachanteile
2. Durch OO-Methoden unberücksichtigte Sprachelemente
3. Programmiersprachen-spezifische Sprachelemente
4. Inpräzise UML-Sprachelemente



UML 2

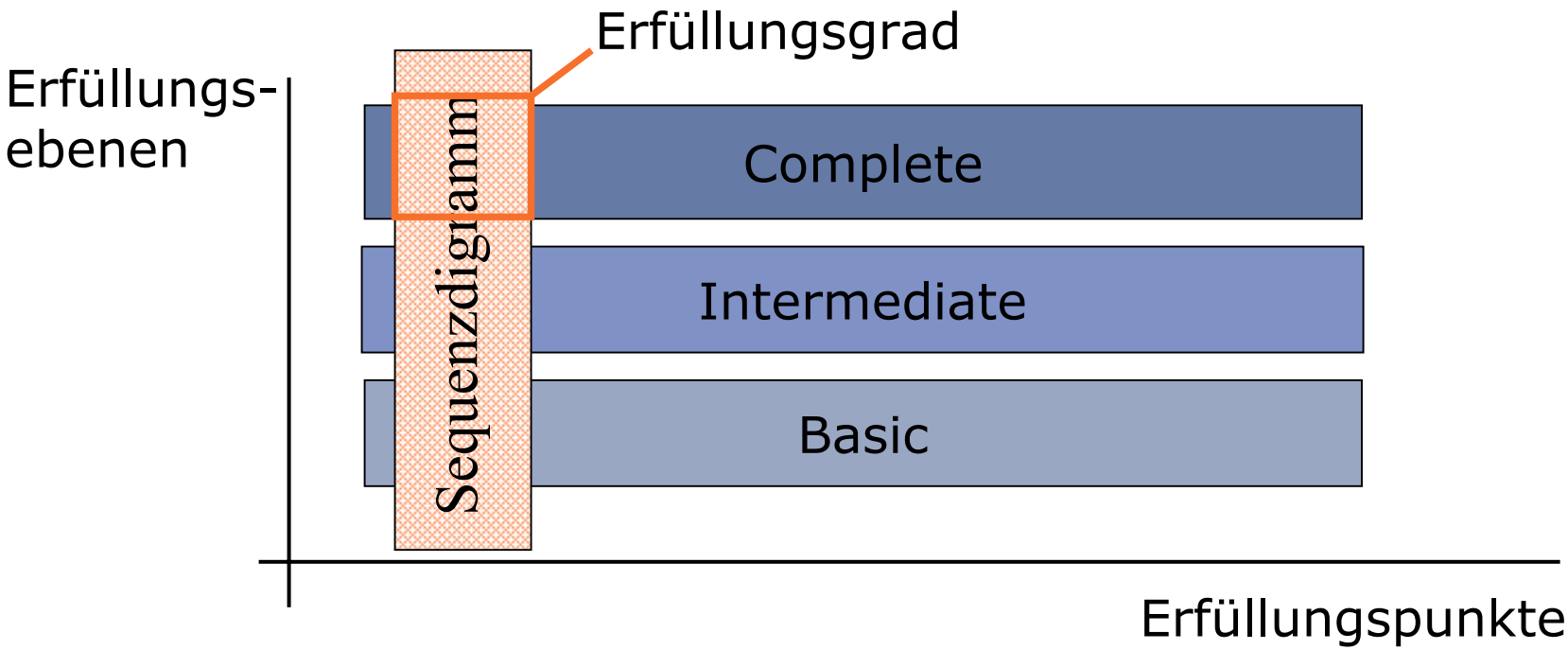
Verrentung existierender Modellelemente

1. Durch UML-Werkzeuge nicht implementierte Sprachanteile
2. Durch OO-Methoden unberücksichtigte Sprachelemente
3. Programmiersprachen-spezifische Sprachelemente
4. Inpräzise UML-Sprachelemente



Schichtenweise UML 2

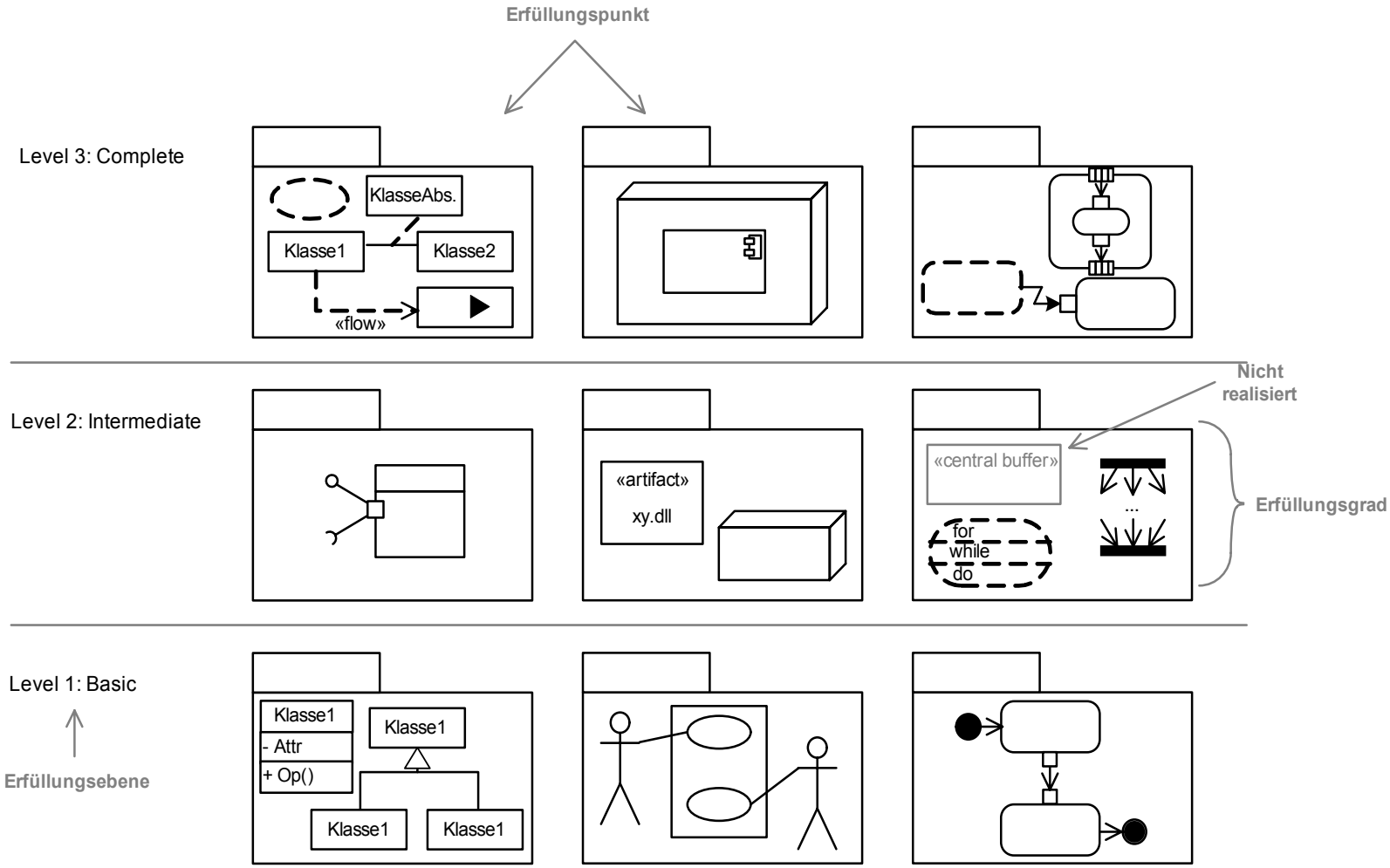
Auch weniger UML ist noch UML!



- Die Idee entstammt der SQL-Standardisierung
- Operationalisiert den Begriff der UML-Unterstützung
- Auch weniger UML ist immer noch UML

Schichtenweise UML 2

Auch weniger UML ist noch UML!



Der Weg zu UML 2

- Verschiedene Weiterentwicklungsvorschläge:
  - **Infrastructure:** 36 Lettern of Intents (LOIs);
    5 Einreichungen durch 28 Firmen
  - **Superstructure:** 37 LOIs;
    5 Einreichungen durch 28 Firmen
  - **OCL:** 30 LOIs; 4 Einreichungen durch 10 Firmen
  - **Diagram Interchange:** 6 LOIs;
    3 Einreichungen durch 6 Firmen

- Eingereicht durch Einzelfirmen und Konsortien

- Bezugnehmend auf einzelne Sprachaspekte der UML v1.x um diese neu zu erweitern; Vorschläge für vollständig neue Diagrammtypen oder die Abschaffung Existierender

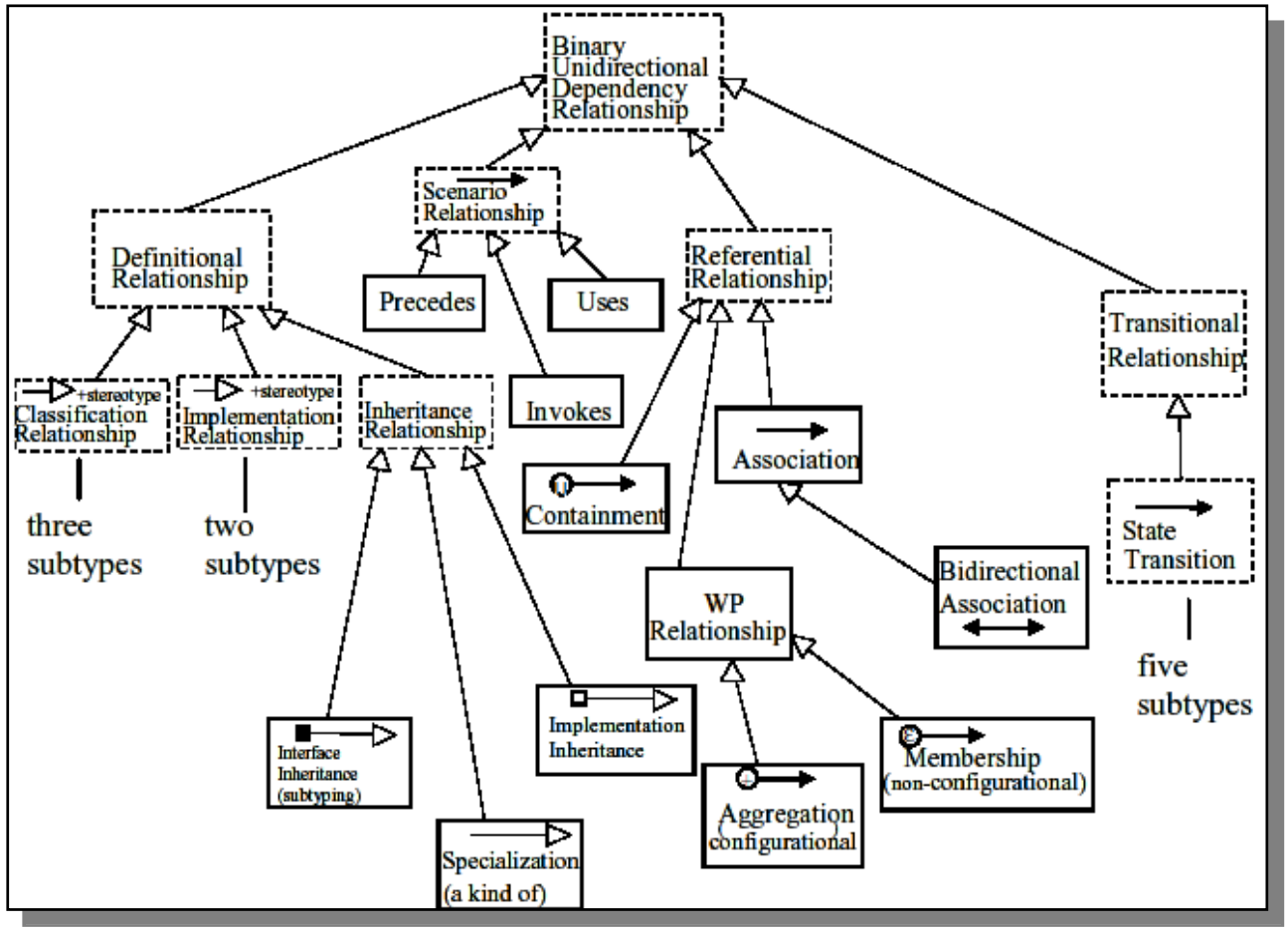
Vorschläge zur UML 2

- ***Superstructure* und *Infrastructure:***
  Ausgereiftester und mit breiter Unterstützung bedachter Vorschlag durch die sog. „UML2 Partners“:

  - Mitglieder:
    Alcatel, Computer Associates, Ericsson, Hewlett-Packard, IONA, Kabira Technologies, Motorola, Oracle, Rational Software, SOFTEAM, Telelogic, and Unisys

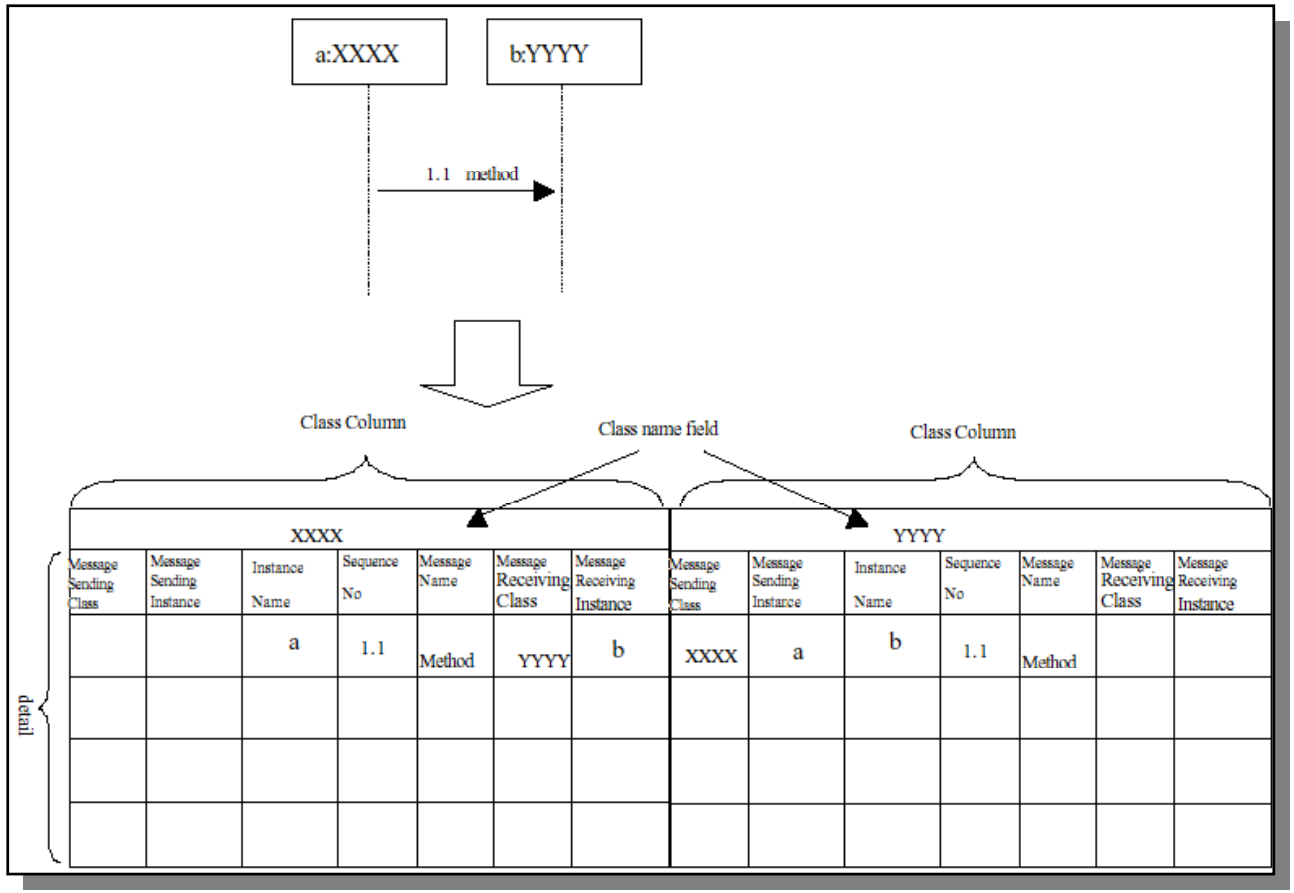
  - Unterstützer:
    Advanced Concepts Center, Ceira Technologies, Compuware, Commisariat à L´Energie Atomique, DaimlerChrysler, Embarcardero Technologies, Enea Business Software, France Telecom, ...

Vorschläge zur UML 2



Einige komplexe Dinge sollten einfacher werden ...

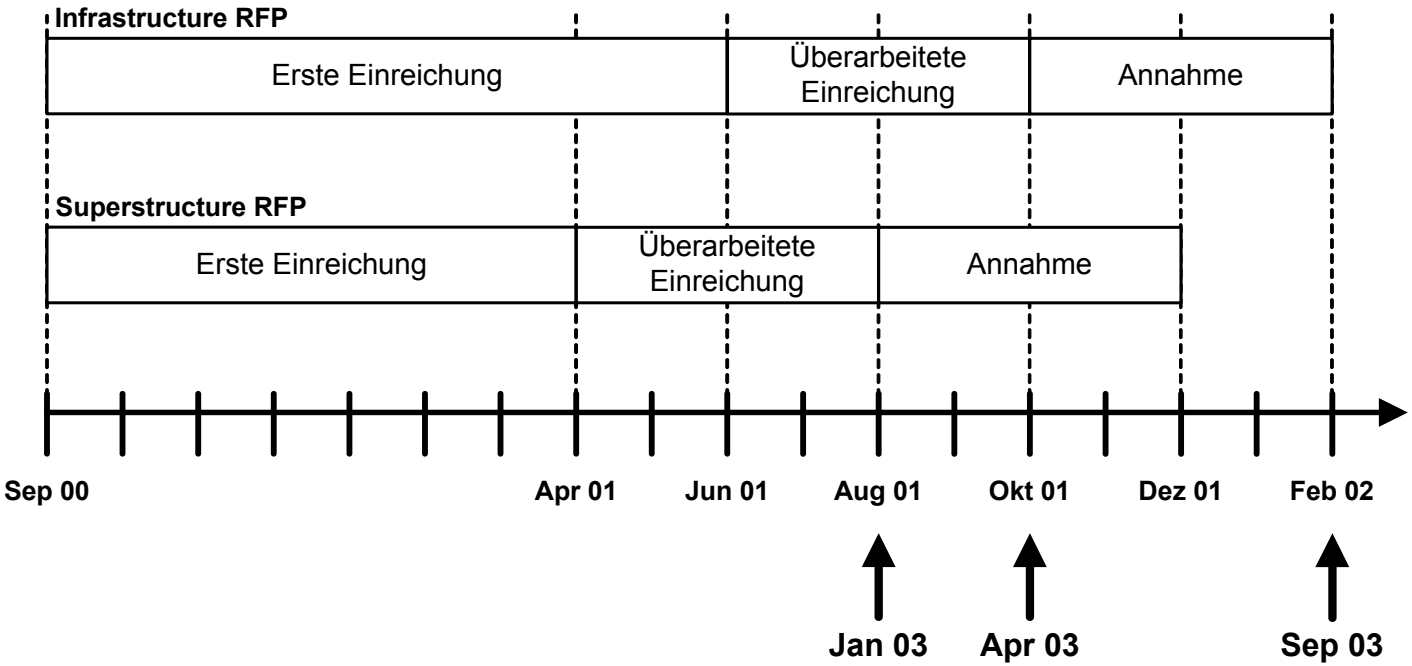
Vorschläge zur UML 2



Manchmal sagen Bilder einfach zu wenig ...

UML 2.0 Ablaufplan

- Ziel: Ein UML 2.0 Standard



UML 2.0 Standardisierungsablauf

> Komplexer Annahmeprozess

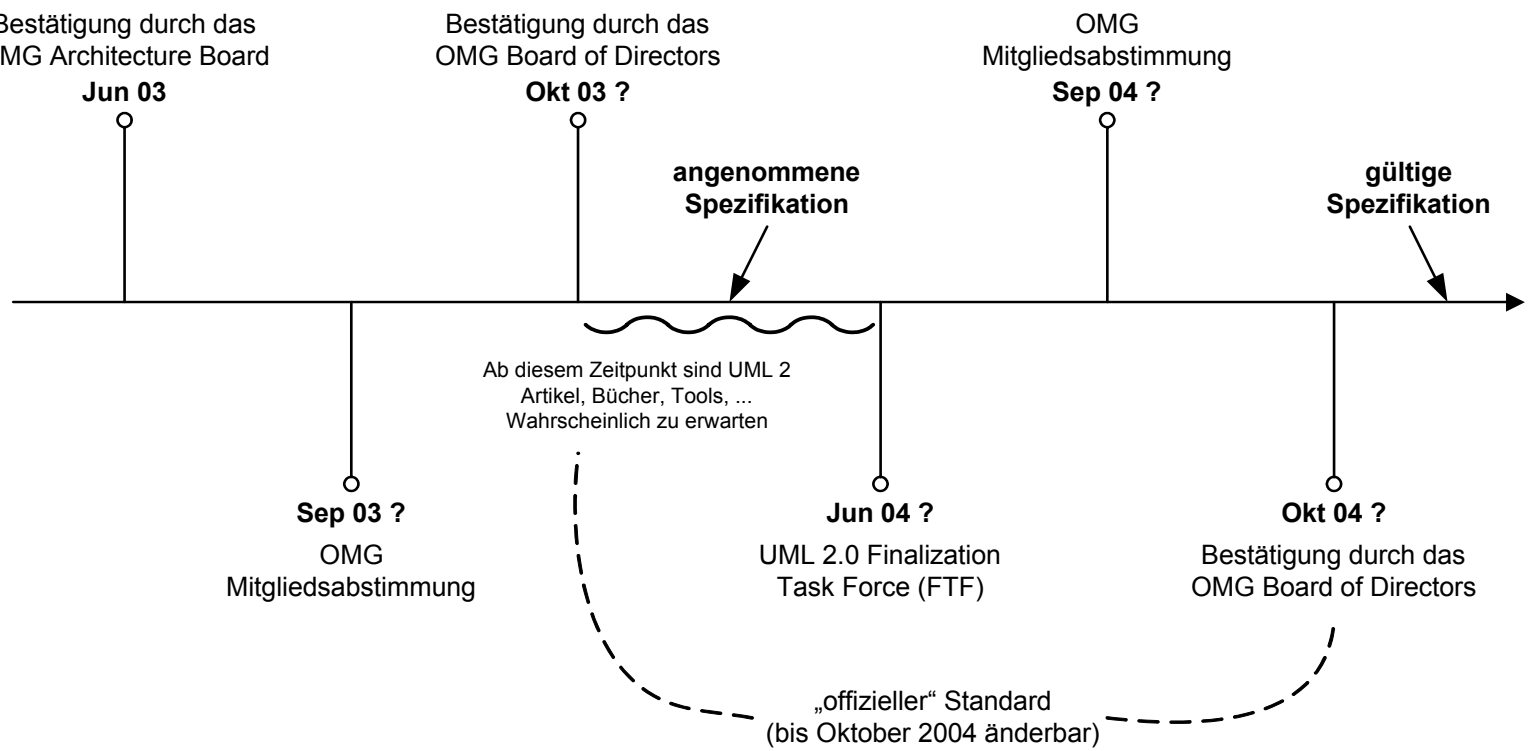


Diagramme der UML 2

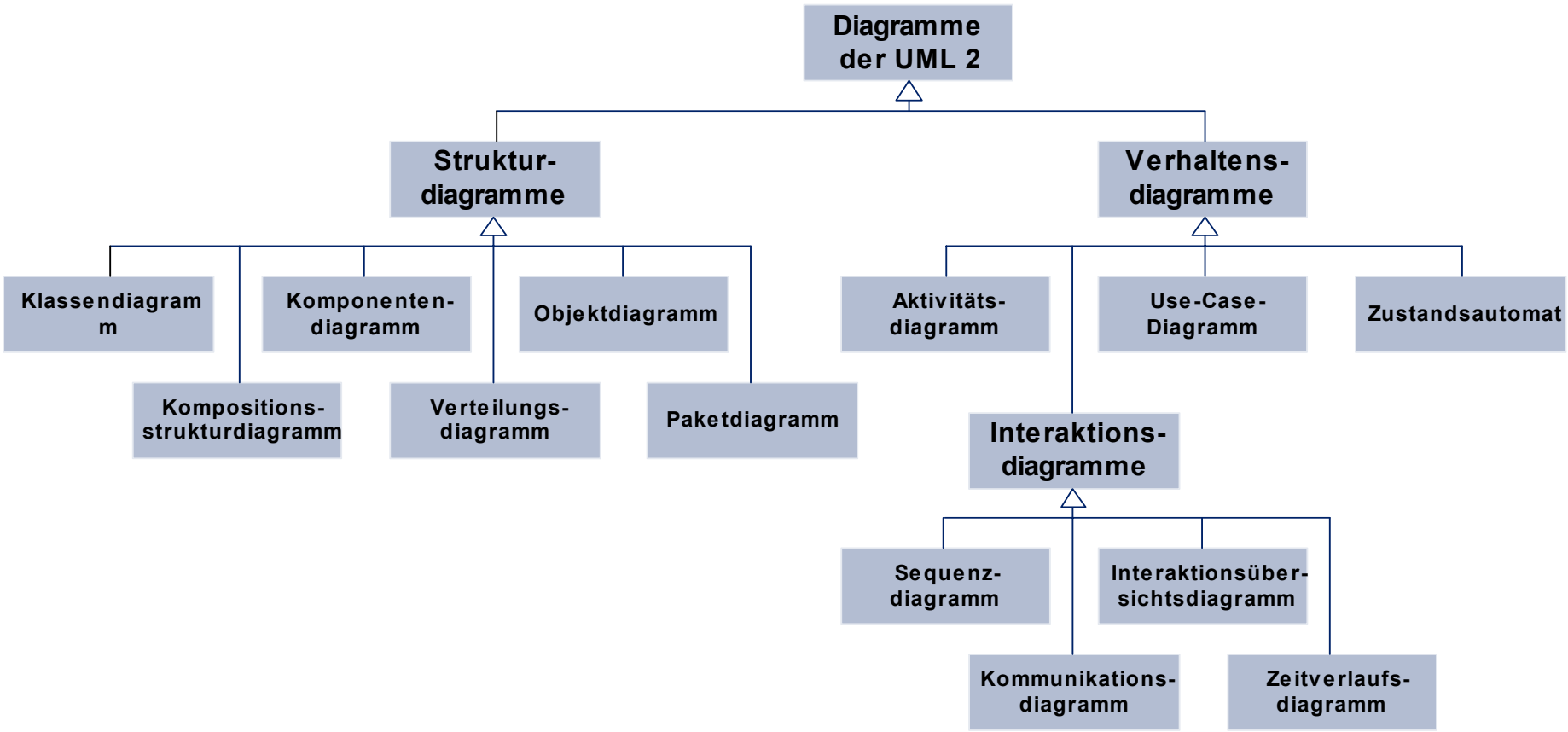


Diagramme der UML - Anwendung I

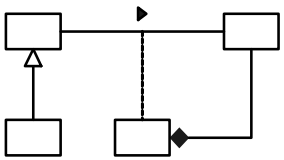
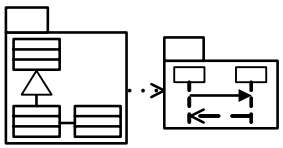
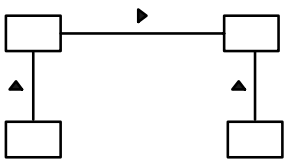
Diagrammtyp	Diese zentrale Frage beantwortet das Diagramm	Stärken
Klassendiagramm 	Aus welchen Klassen besteht mein System und wie stehen diese untereinander in Beziehung?	Beschreibt die statische Struktur des Systems. Enthält alle relevanten Struktur-zusammenhänge/Datentypen. Brücke zu dynamischen Diagrammen. Normalerweise unverzichtbar.
Paketdiagramm 	Wie kann ich mein Modell so schneiden, dass ich den Überblick bewahre?	Logische Zusammenfassung von Modellelementen. Modellierung von Abhängigkeiten/ Inklusion möglich.
Objektdiagramm 	Welche innere Struktur besitzt mein System zu einem bestimmten Zeitpunkt zur Laufzeit (Klassendiagramm-schnappschuss)?	Zeigt Objekte u. Attributbelegungen zu einem bestimmten Zeitpunkt. Verwendung beispielhaft zur Veranschaulichung Detailniveau wie im Klassen-diagramm. Sehr gute Darstellung von Mengenverhältnissen.

Diagramme der UML - Anwendung II

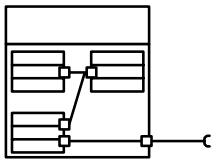
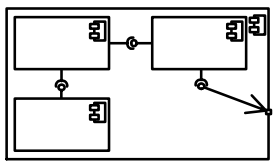
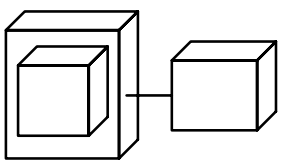
Diagrammtyp	Diese zentrale Frage beantwortet das Diagramm	Stärken
Kompositionsstruktur-diagramm 	Wie sieht das Innenleben einer Klasse, einer Komponente, eines Systemteils aus?	Ideal für die Top-Down-Modellierung des Systems (Ganz-Teil-Hierarchien). Zeigt Teile eines „Gesamtelements“ und deren Mengenverhältnisse. Präzise Modellierung der Teile-Beziehungen über spezielle Schnittstellen (Ports) möglich.
Komponentendiagramm 	Wie werden meine Klassen zu wieder verwendbaren, verwaltbaren Komponenten zusammengefasst und wie stehen diese in Beziehung?	Zeigt Organisation und Abhängig-keiten einzelner technischer Systemkomponenten. Modellierung angebotener und benötigter Schnittstellen möglich.
Verteilungsdiagramm 	Wie sieht das Einsatzumfeld (Hardware, Server, Datenbanken, …) des Systems aus? Wie werden die Komponenten zur Laufzeit wohin verteilt?	Zeigt das Laufzeitumfeld des Systems mit den „greifbaren“ Systemteilen. Darstellung von „Softwareservern“ möglich. Hohes Abstraktionsniveau, kaum Notationselemente.

Diagramme der UML - Anwendung III

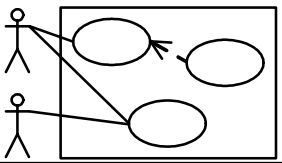
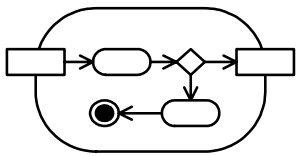
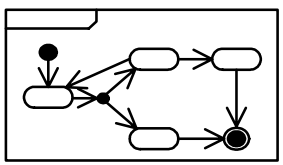
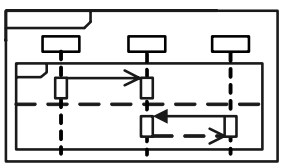
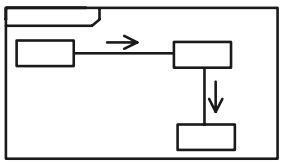
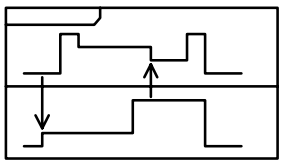
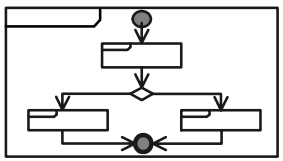
Diagrammtyp	Diese zentrale Frage beantwortet das Diagramm	Stärken
Use-Case-Diagramm 	Was leistet mein System für seine Umwelt (Nachbarsysteme, Stakeholder)?	Außensicht auf das System. Geeignet zur Kontextabgrenzung. Hohes Abstraktionsniveau, einfache Notationsmittel.
Aktivitätsdiagramm 	Wie läuft ein bestimmter fluss-orientierter Prozess oder ein Algorithmus ab?	Sehr detaillierte Visualisierung von Abläufen mit Bedingungen, Schleifen, Verzweigungen. Parallelisierung und Synchronisation. Darstellung von Datenflüssen.
Zustandsautomat 	Welche Zustände kann ein Objekt, eine Schnittstelle, ein Use Case, … bei welchen Ereignissen annehmen?	Präzise Abbildung eines Zustands-modells mit Zuständen, Ereignissen, Nebenläufigkeiten, Bedingungen, Ein- und Austrittsaktionen. Schachtelung möglich.
	Wer tauscht mit wem welche Informationen in welcher Reihenfolge aus?	Darstellung d. Informationsaustauschs zwischen Kommunikationspartnern Sehr präzise Darstellung der zeitlichen Abfolge auch mit Nebenläufigkeiten.

Diagramme der UML und ihre Anwendung IV

Diagrammtyp	Diese zentrale Frage beantwortet das Diagramm	Stärken
Kommunikations-diagramm 	Wer kommuniziert mit wem? Wer „arbeitet“ im System zusammen?	Stellt den Informationsaustausch zwischen Kommunikationspartnern dar. Überblick steht im Vordergrund (Details und zeitliche Abfolge weniger wichtig).
Timingdiagramm 	Wann befinden sich verschiedene Interaktionspartner in welchem Zustand?	Visualisiert das exakte zeitliche Verhalten von Klassen,Schnittstellen,.. Geeignet für die Detailbetrachtungen, bei denen es wichtig ist, dass ein Ereignis zum richtigen Zeitpunkt eintritt.
Interaktionsübersichts-diagramm 	Wann läuft welche Interaktion ab?	Verbindet Interaktionsdiagramme (Sequenz-, Kommunikation- und Timingdiagramme) auf Top-Level-Ebene. Hohes Abstraktionsniveau.

Anforderungen eingehalten?

Die Bewertung

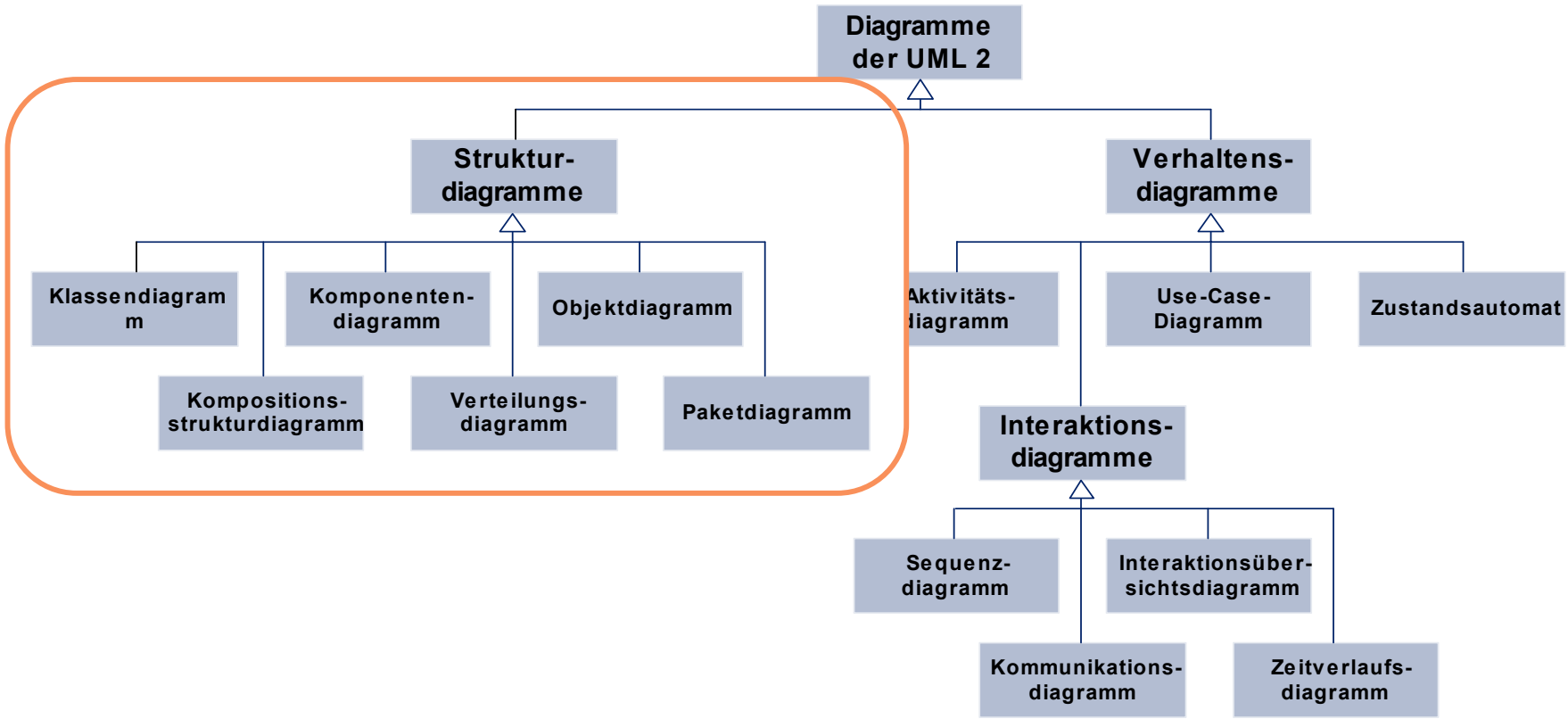
- UML kompakter und in sich schlüssiger durch Anpassung von Konzepten der Infrastruktur
- Angestrebte Abwärtskompatibilität zu älteren UML-Versionen in vielen Bereichen verletzt
- Die aktuelle Version der UML 2 Superstructure beinhaltet noch einige Kinderkrankheiten
- Das grafische Aussehen einiger Notationselemente hat sich verändert, was sicherlich zur Verwirrung führen wird
- Eingeführte Bezeichnungen und Namen wurden verändert

Umsteigen: Ja oder Nein?

Wann sich der Umstieg wirklich lohnt

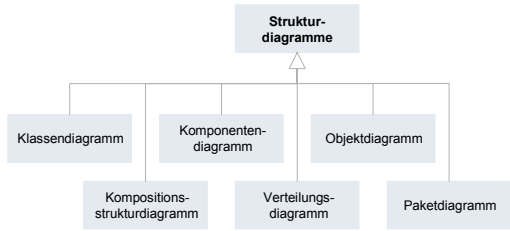
- Die Modellierung des Systemverhaltens steht im Vordergrund: häufige Verwendung von Zustandsautomaten und Sequenz-diagrammen
- Modellierung im Umfeld technischer Systeme: bessere Darstellung z.B. der Kommunikation von Systemkomponenten oder des Zeitverhaltens eines Systems
- Modellierung von Geschäftsprozessen, bzw. von Aktivitäten eines Systems mittels Aktivitätsdiagrammen
- Generierung von Code oder Architekturen basierend auf einem MDA-Ansatz, oder in sich schnell ändernden, architektonischen Umfeldern wie EAI (Enterprise Application Integration)

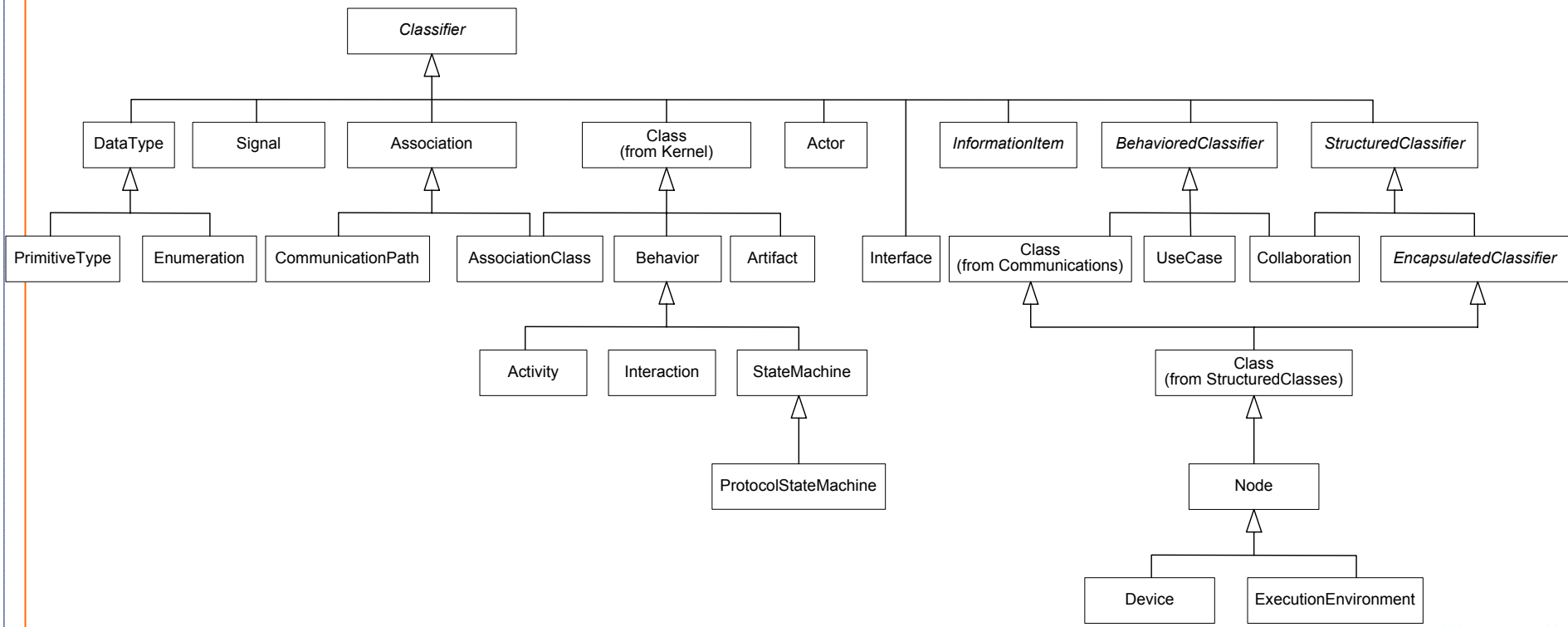
Diagramme der UML 2



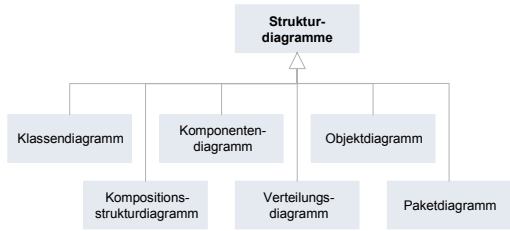
Basiskonzepte

- UML 2 ...
  - Erweitert die Nutzung der *Classifier*

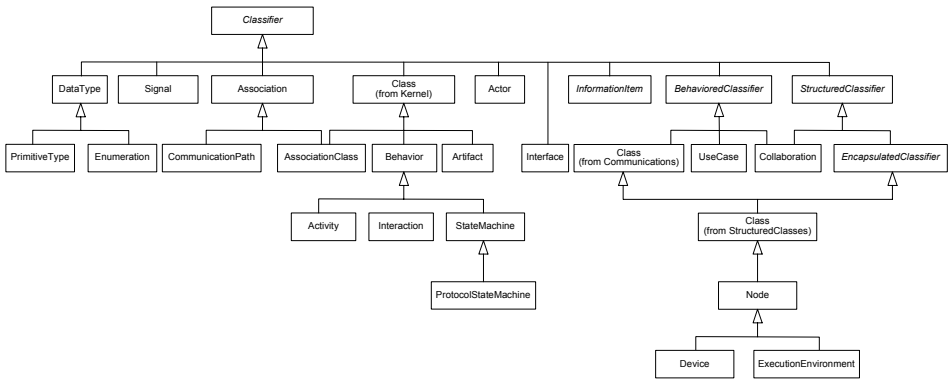




Basiskonzepte -- *Classifier*



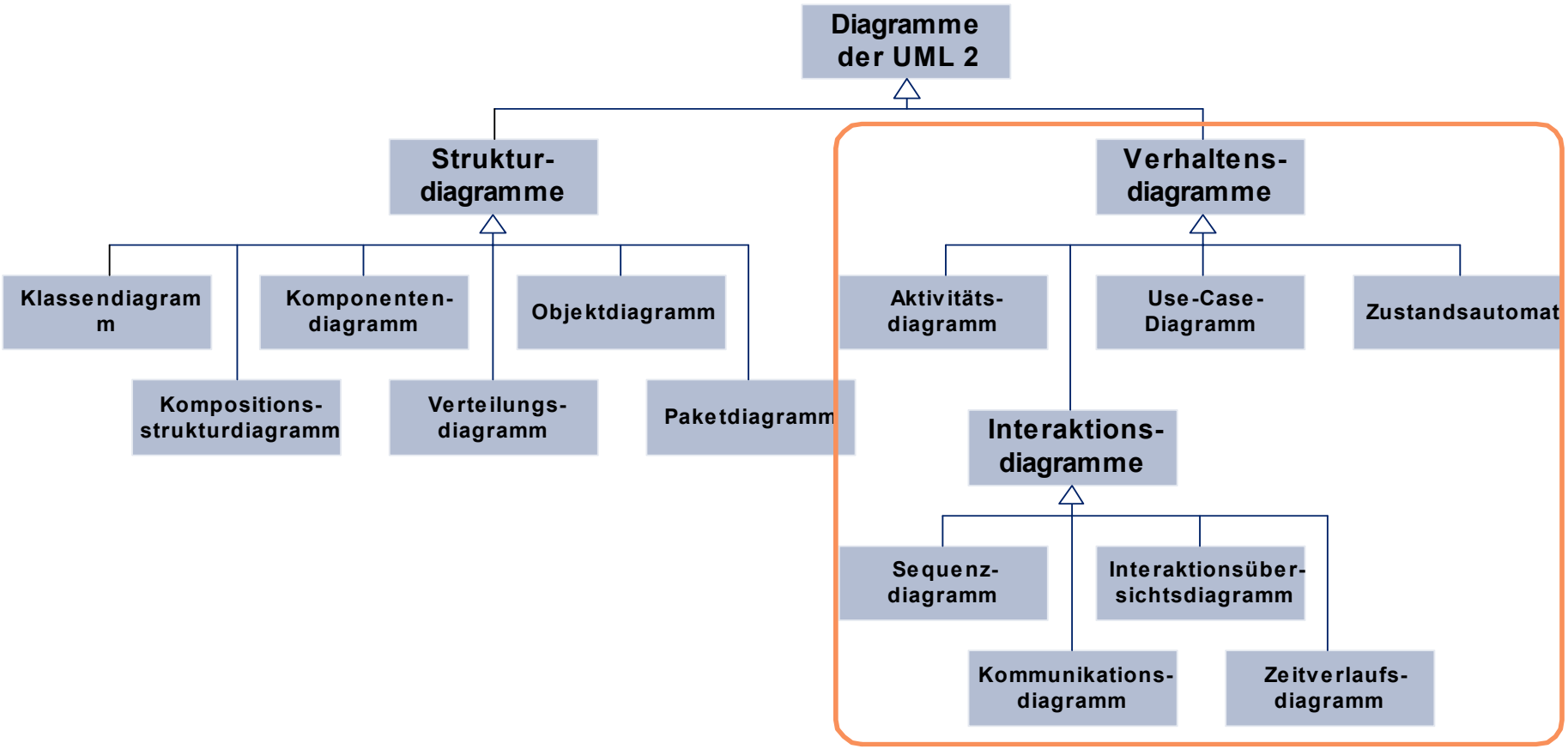
- UML 2 ...
  - Erweitert die Nutzung der *Classifier*
    - Beziehungen (*Assoziationen*) werden zwischen ihnen geknüpft
    - ... können Charakteristika (*Attribute*) besitzen
    - ... können Verhaltensspezifikationen (*Operationen*) besitzen
    - ... können generalisiert werden
    - ... können autonom auf Signale reagieren
    - ... können ausschließlich der Strukturierung dienen (*abstract*)



Fazit ---
statische Diagramme in UML 2

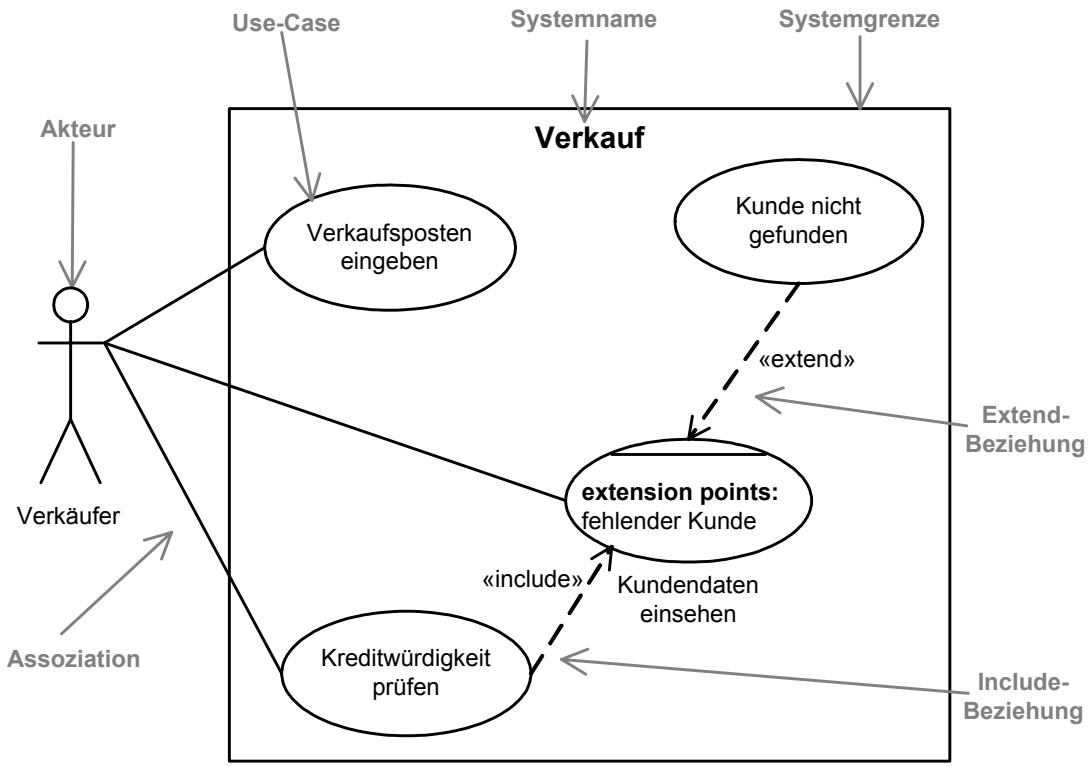
- Absichtsvoll wenig „spürbar“ Neues
  - Im Hintergrund
    - Klarere Basiskonzepte
    - Deutlich mehr Wiederverwendung
    - Fast vollständig neues Metamodell
- Modifizierte Semantik vieler (Meta-)Modellelemente
- Einige neue graphische Primitive
- Reichhaltigeres Angebot an Abhängigkeiten und „eingebauten“ Modellerweiterungen (Stereotypen)
- Behutsame Verrentung „problematischer“ Modellelemente

Diagramme der UML 2

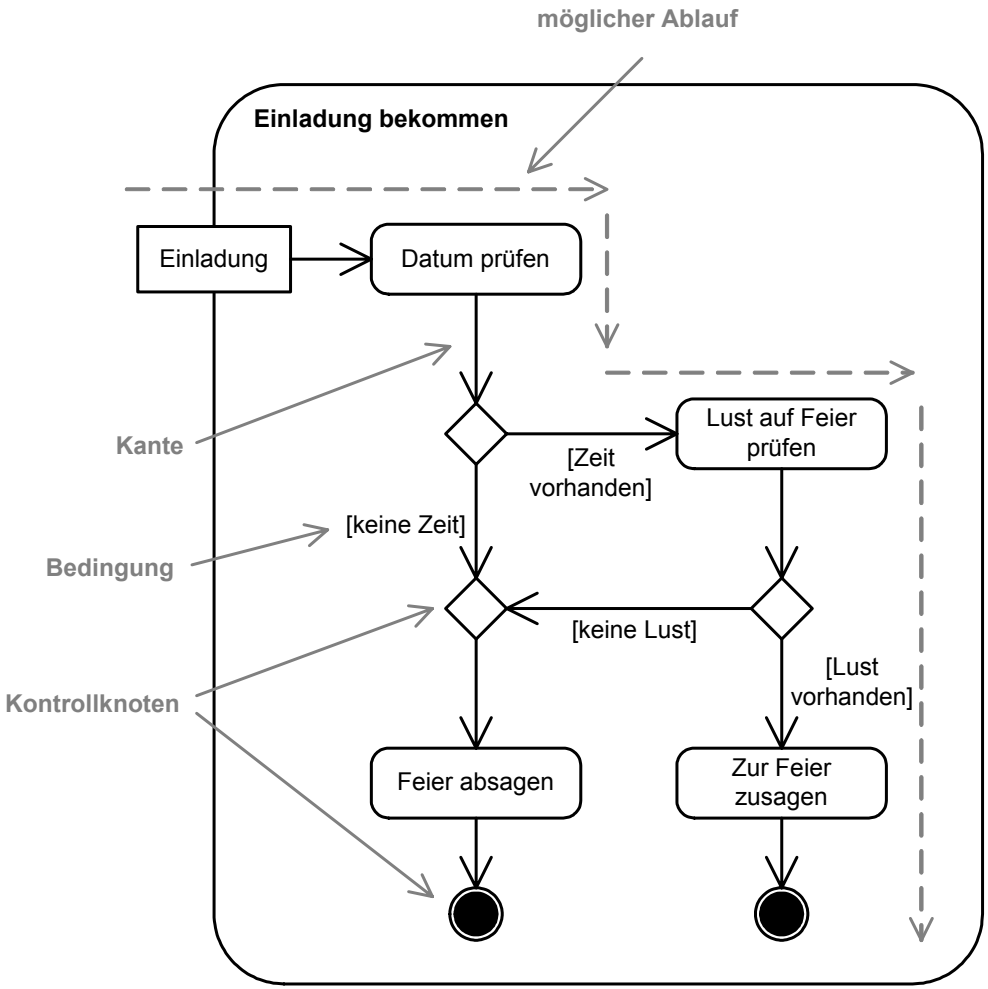


Use-Case-Diagramm

- Geändert in UML 2:
  - Name eines Akteurs nicht mehr optional, sondern verpflichtend
  - Classifier können Use-Cases besitzen, nicht nur Pakete
  - Vorbedingung und extension point werden als Notiz an die Extend-Beziehung angehängt

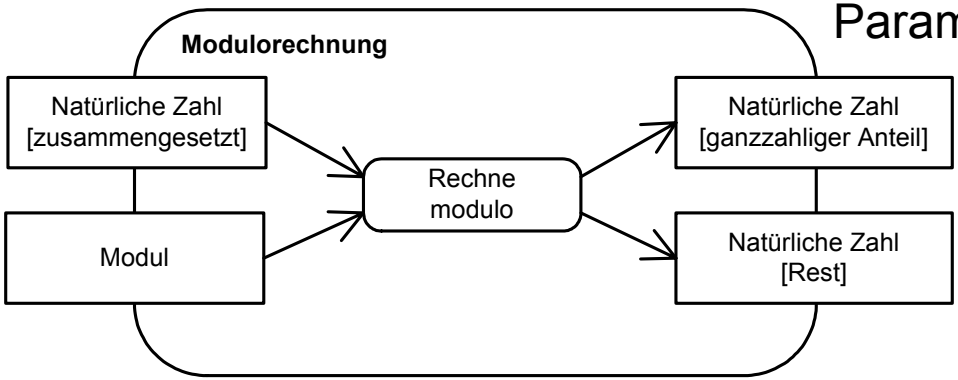


Aktivitätsdiagramm

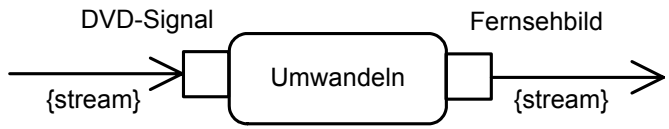


Aktivitätsdiagramm

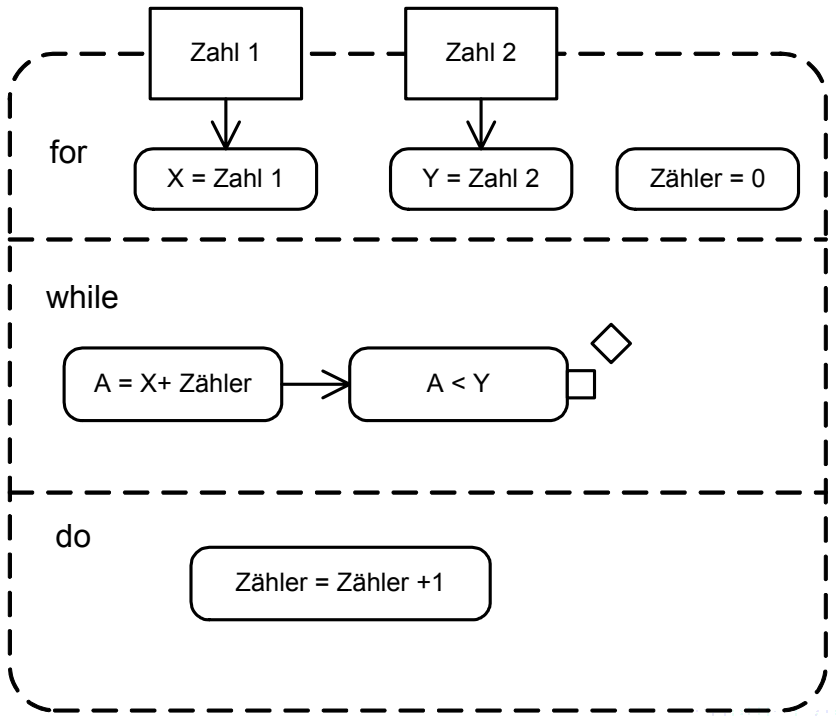
Parameter



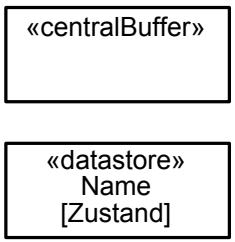
Streaming



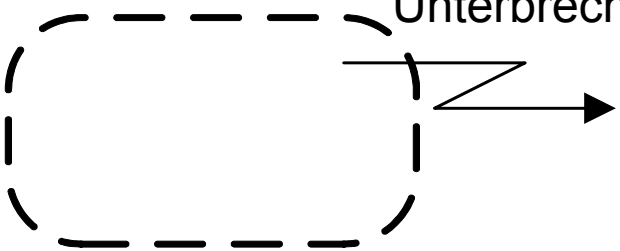
Schleifenknoten



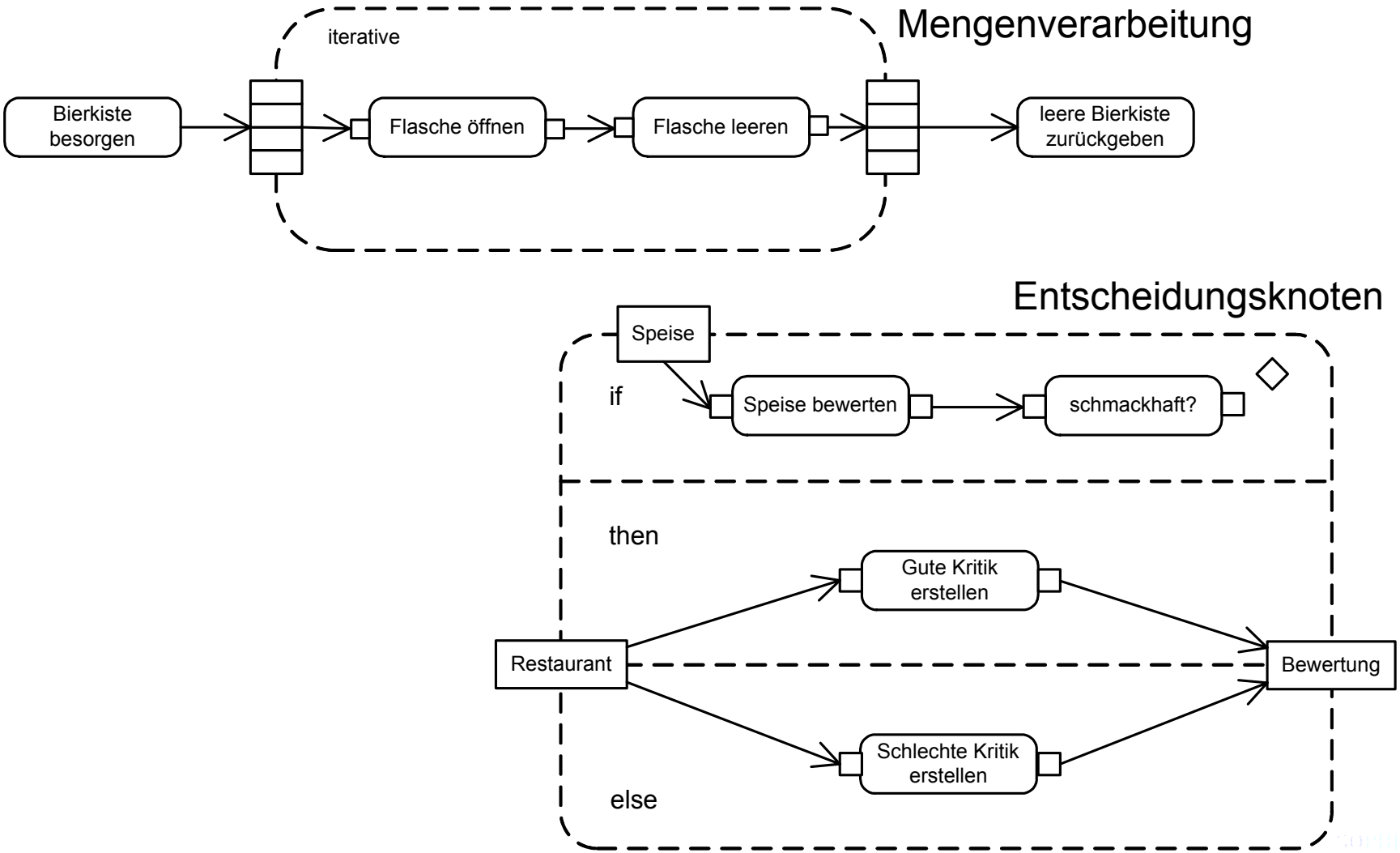
Datenspeicher



Unterbrechungsbereich



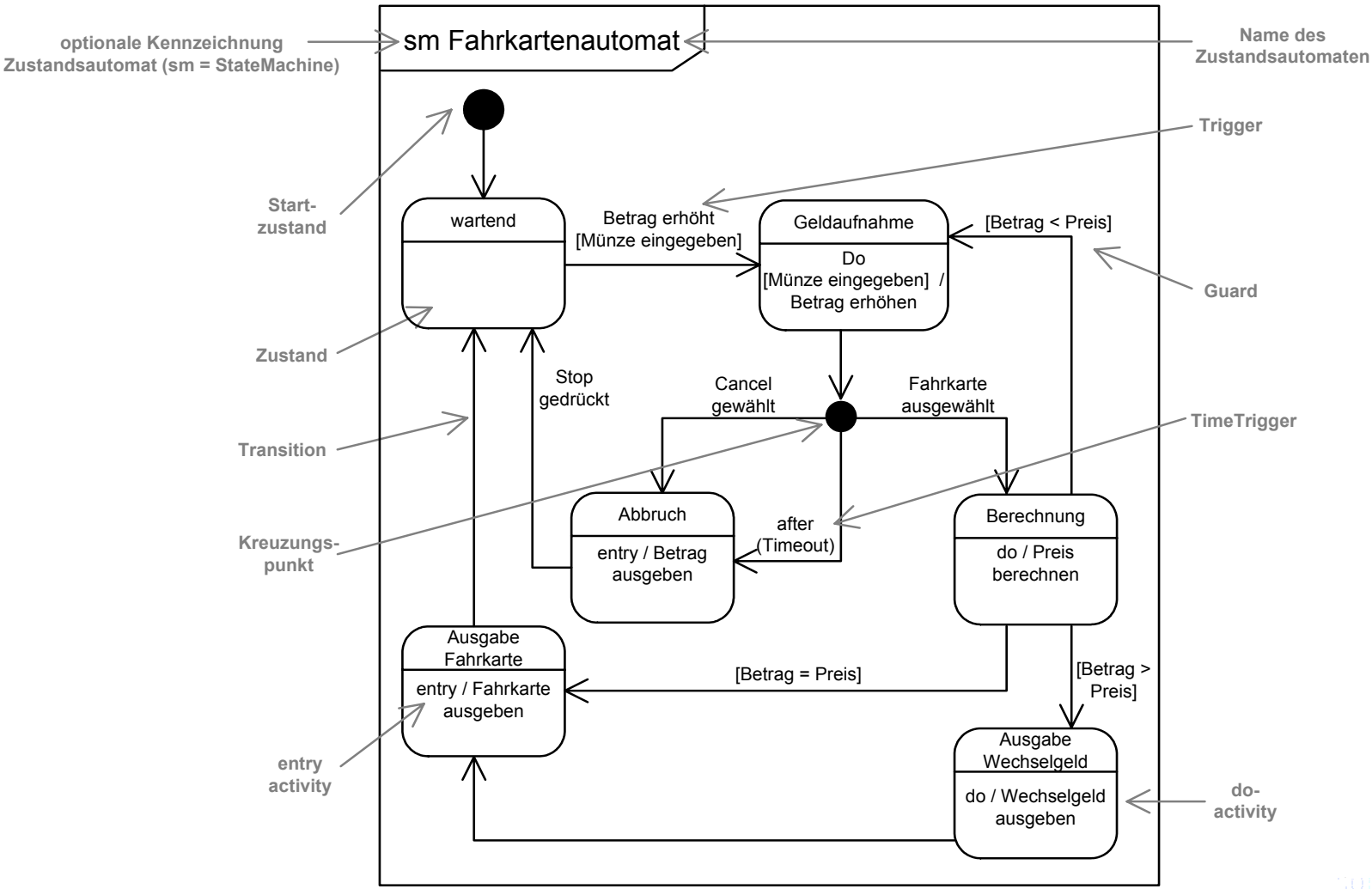
Aktivitätsdiagramm



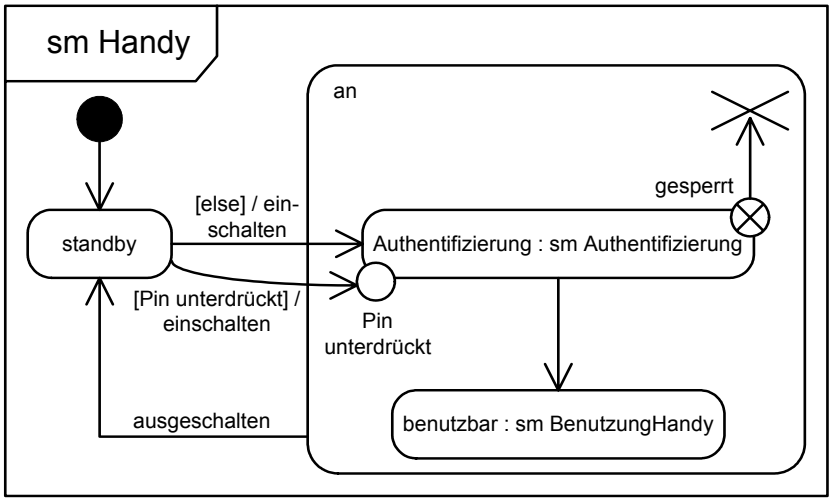
Aktivitätsdiagramm

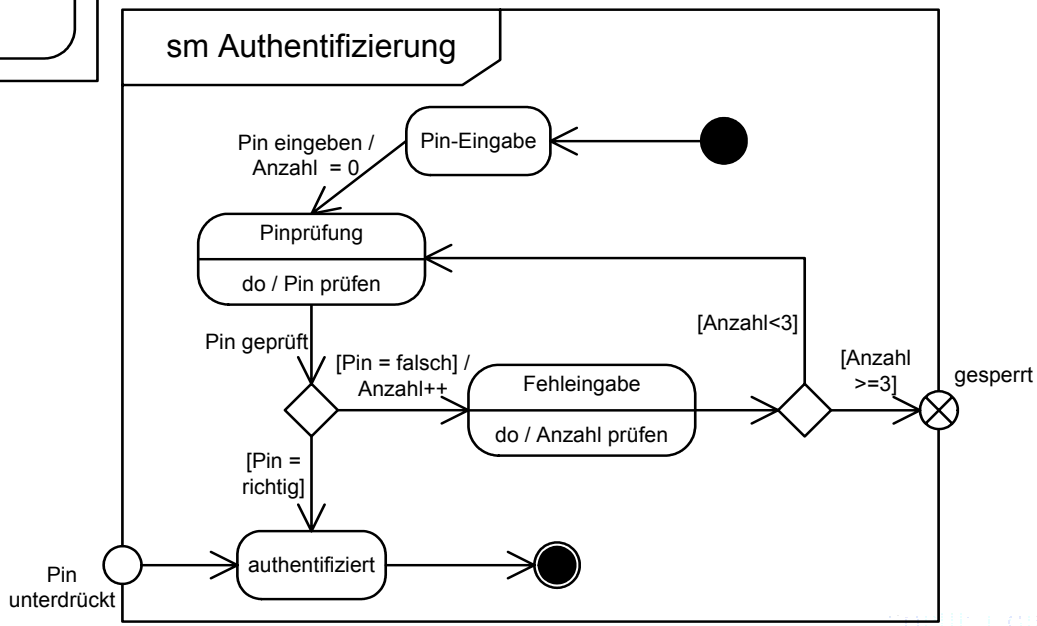
- Neu in UML 2:
  - **Neue Notationselemente:** Strukturierte Knoten, Entscheidungsknoten, Schleifenknoten, Mengenverarbeitungsknoten, Unterbrechungsbereich, Datenspeicher und Bufferknoten, Parametersatz
  - Aktivitäten verwenden eine den Petri-Netzen ähnlich Semantik (Token-Konzept)
  - Aktivitäten können Ein- und Ausgabeparameter enthalten
  - Aktionen können mit Vor- und Nachbedingungen verknüpft werden
- Geändert in UML 2:
  - Aktivitäten sind nun unabhängig von Zustandsautomaten
  - Es sind nun mehrere Startknoten erlaubt
  - Parallele Abläufe müssen nicht wieder zusammengeführt werden
  - Aktivitätsbereiche können hierarchisch oder multidimensional sein
  - Die Notation von Aktionen entspricht der Notation von Zuständen der UML 1.x

Zustandsautomat



Zustandsautomat





Zustandsautomat

- Neu in UML 2:
  - Ports können nun Protokollzustandsautomaten besitzen
  - Ein-, Austrittspunkte und Terminatoren wurden eingeführt
  - Regeln zur Ergänzung und Ersetzung von Transitionen bei vererbten Zustandsautomaten wurde hinzugefügt
- Geändert in UML 2:
  - Tiefe Historien können auch Ziel einer Transition innerhalb des enthaltenen Zustands sein (also nicht nur von außen)
  - Das vererbte Verhalten wird mit einem Zustandsautomaten dargestellt und nicht nur durch eine Notiz

Interaktionsmodellierung

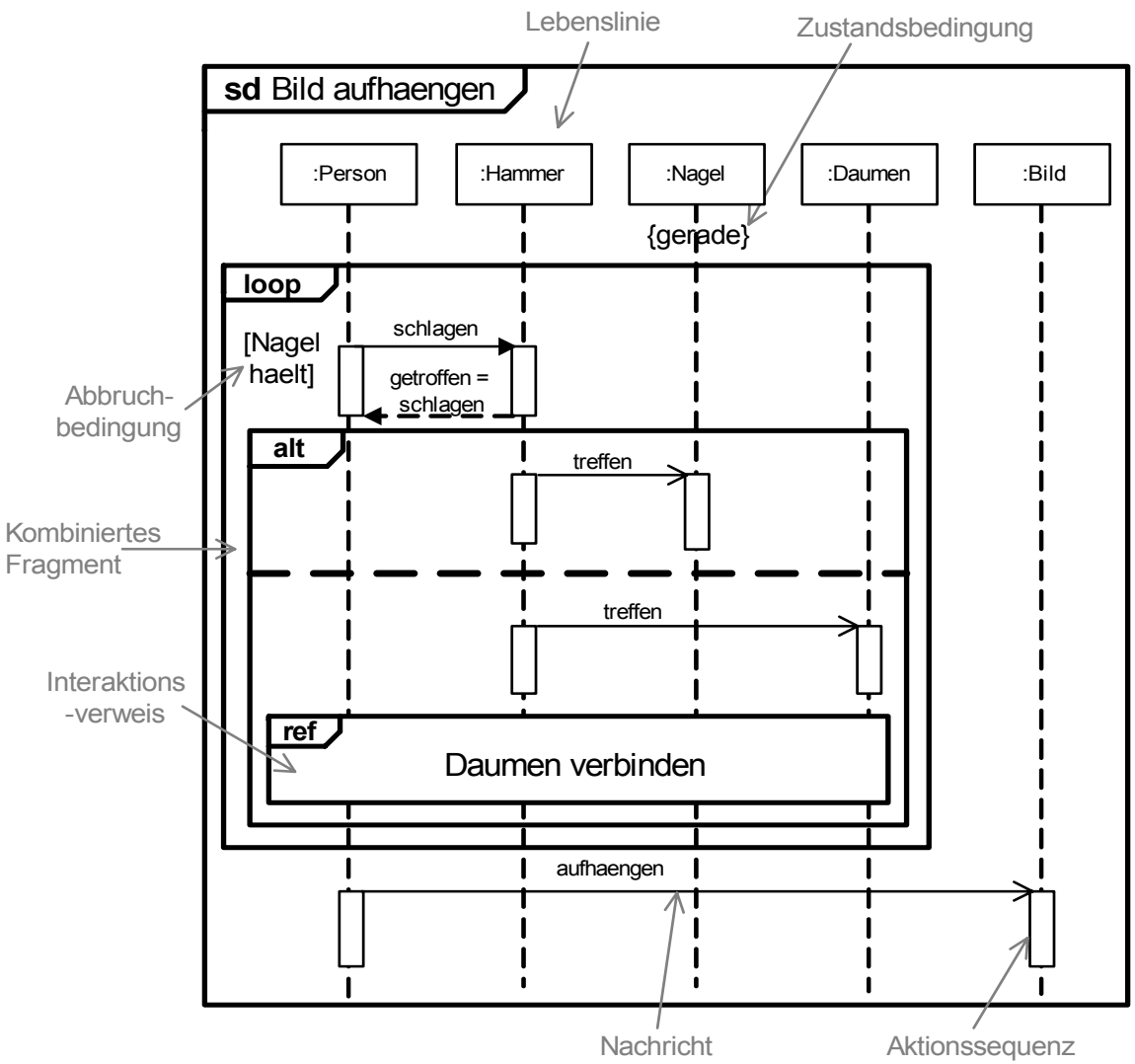
- **Funktion**:
  - Interaktion ist der Nachrichten- und Datenaustausch zwischen zwei Kommunikationspartnern
  - Nachrichtenaustausch verbindet Sende- und Empfangsereignisse

- **Aufgabe im Projekt**:
  - Interaktionsdiagramme zeigen nur Sichten auf Interaktionen
  - Diagrammwahl nach hervorzuhebenden Modellierungsaspekt

Interaktionsmodellierung

- Diagrammwahl / Modellierungsaspekt :
  - Sequenzdiagramm:
    - + Die Reihenfolge, in der der Nachrichtenaustausch stattfindet
  - Kommunikationsdiagramm:
    - + Das Zusammenspiel von **strukturierten** Kommunikationspartnern
    - + Modellierung von Prinzipien und Konzepten
  - Timing-Diagramm:
    - + Darstellung der zeitlichen Veränderung eines Classifiers
    - + Modellierung zeitkritischer Zustands- und Wertänderungen sinnvoll
  - Interaktionsübersichtsdiagramm:
    - + Darstellung der Ablaufreihenfolge mehrerer Interaktionen
    - + Logischer Zusammenhang zwischen Interaktionsdiagrammen
    - + Brückenschlag zwischen Aktivitäts- und Interaktionsdiagrammen

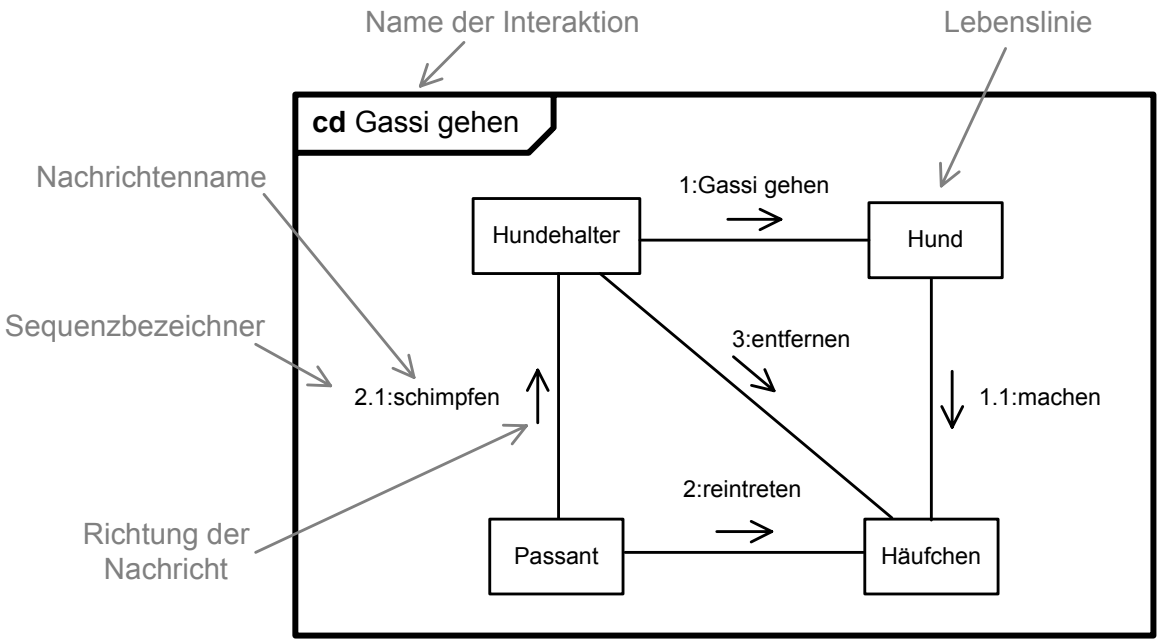
Sequenzdiagramme



Sequenzdiagramme

- Neu in UML 2:
  - Zustandsinvarianten können an Lebenslinien angetragen werden
  - „lost“ und „found“ Nachrichten eingeführt
- Geändert in UML 2:
  - Innerhalb von Sequenzdiagrammen kann auf andere Interaktionen verwiesen werden (Zerlegung von Abläufen oder Lebenslinien)
  - Alle Kontrollflussmöglichkeiten (if, switch, ...) höherer Programmiersprachen durch kombinierte Fragmente, sehr gute Unterstützung von nebenläufiger Modellierung
- Entfällt in UML 2:
  - Nachrichtenart „unspezifiziert“ entfällt

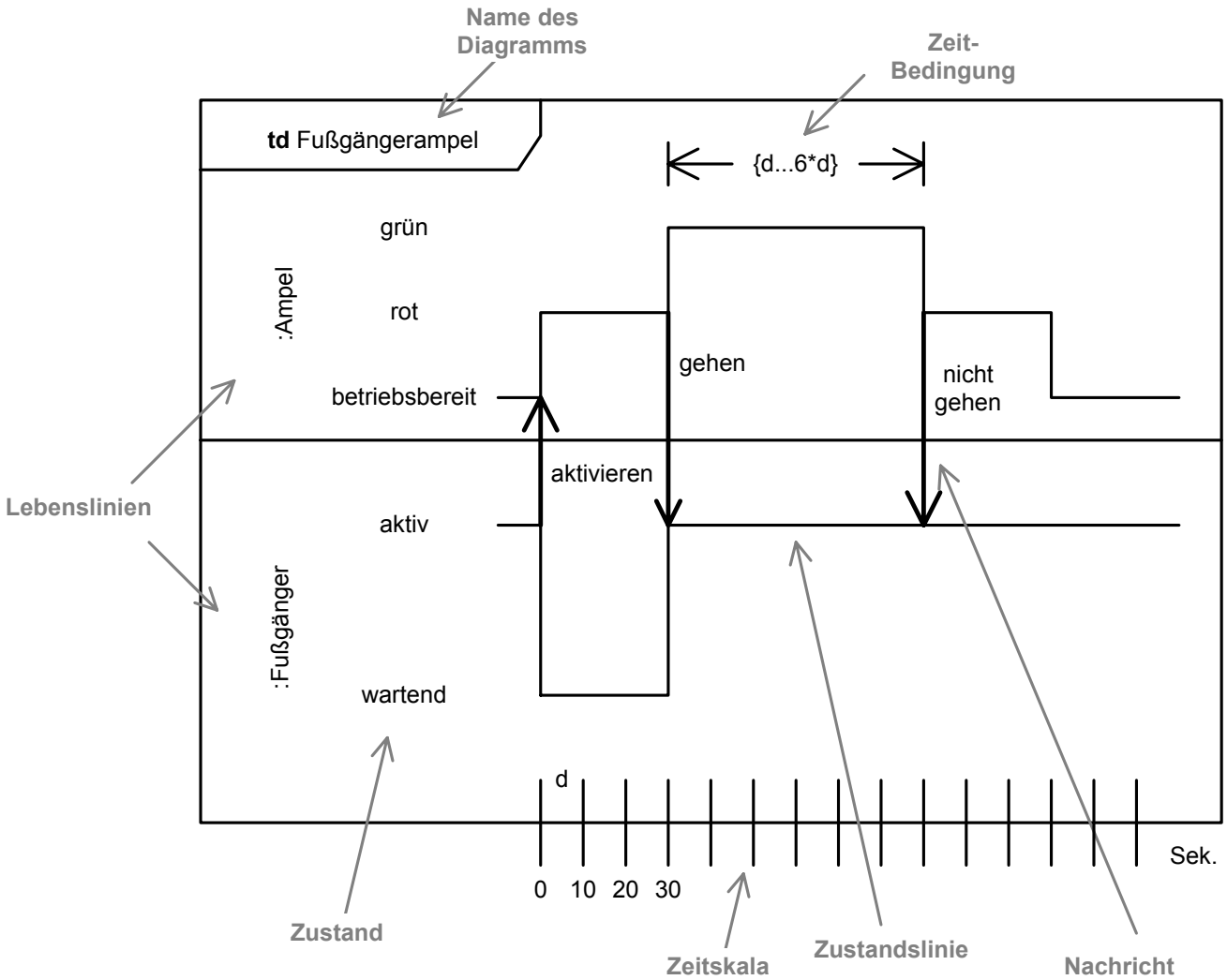
Kommunikationsdiagramme



Kommunikationsdiagramme

- Geändert in UML 2:
  - Kommunikationsdiagramm statt Kollaborationsdiagramm
  - Subset des Sequenzdiagramms, z. B.
    - keine Interaktionsverweise („ref“)
    - keine kombinierten Fragmente
    - Ereignisreihenfolge wird ignoriert

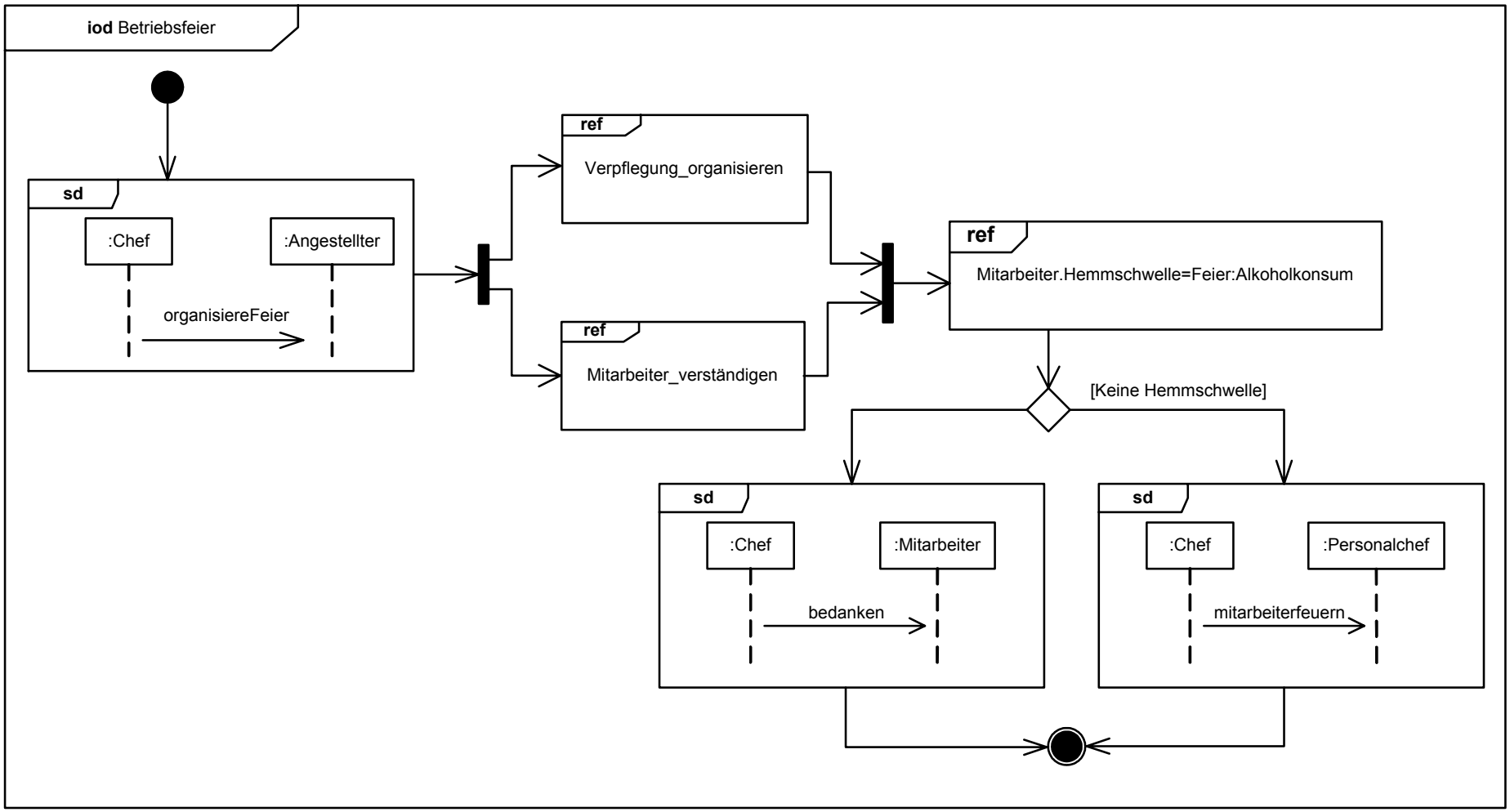
Timing-Diagramm



Timing-Diagramm

- Neu in UML 2:
  - Komplett neu in die UML 2 eingeführt
  - Bereits seit Jahren in Elektrotechnik genutzt (z. B. für elektronische Schaltvorgänge)

Interaktionsübersichtsdiagramme



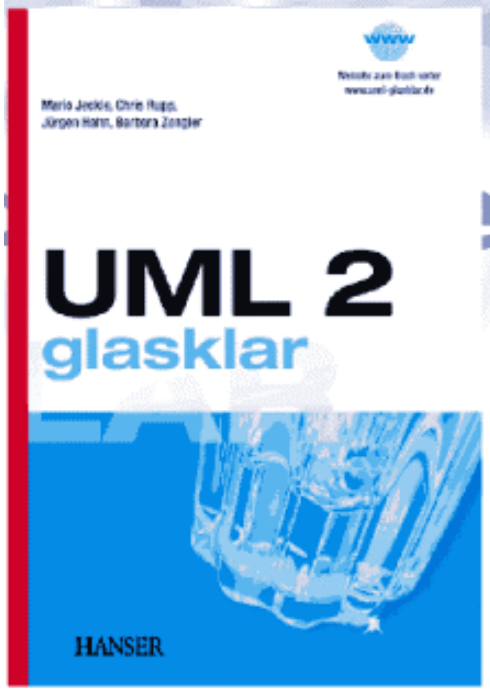
Interaktionsübersichtsdiagramme

- Neu in UML 2:
  - Komplett neu in die UML 2 eingeführt

Fazit ---
dynamische Diagramme in UML 2

- Viel neu - viel geklaut
- Aktivitätsdiagramme und Sequenzdiagramme faktisch neu
- Verbesserte Unterstützung der Codeabbildung
- Bessere Modellierung von Nebenläufigkeiten und Zeitverhalten möglich ➔ dennoch keine „Echtzeitsprache“
- Deutlich bessere Schnittstelle zu Strukturdiagrammen
- Notationsvielfalt erfordert Tailoring
- Interaktionsübersichtsdiagramm, Timingdiagramm erfordern Nachbesserungen

Damit Sie klar sehen!



Besuchen Sie uns bei unserem Buchvorstellungsevent am 3. Dezember!

Ort: SOPHIST GROUP Nürnberg ab 17 Uhr

damit Sie klar sehen!

Trainings:

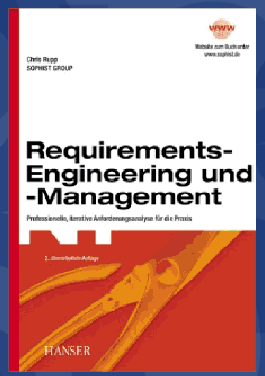
	UML 2 Update:	27.11.2003
	OOA:	09.12.-10.12.2003
	OOD:	11.12.-12.12.2003

Beratung: wann immer Sie wollen ;-))

Infos: www.uml-glasklar.de www.sophist.de

Unsere Bücher

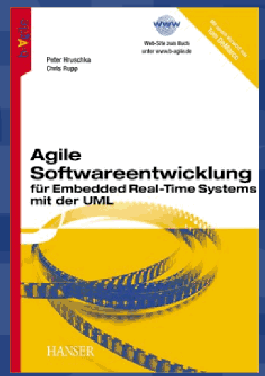
Requirements Engineering und -Management



Chris Rupp und die SOPHISTen

Aktualisierte, zweite Auflage seit 23. Mai 2002 im Handel
ISBN:3-446-21960-9

Agile Software-entwicklung

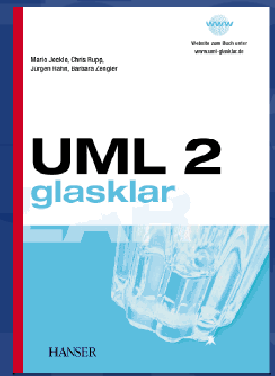


Chris Rupp und Peter Hruschka

Im Handel seit 21. Mai 2002

ISBN:3-446-21997-8

UML 2 glasklar

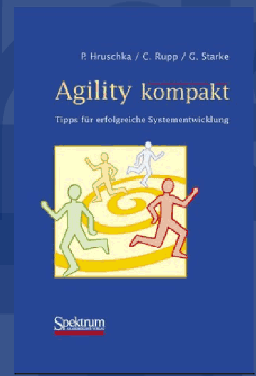


Chris Rupp, Mario Jeckle, Jürgen Hahn, Barbara Zengler und Stefan Queins

Erstauflage ab November 2003

ISBN: 3-446-22575-7

Agilität kompakt



Chris Rupp, Peter Hruschka und Gernot Starke

Erstauflage ab Oktober 2003

ISBN 3-827-41483-0