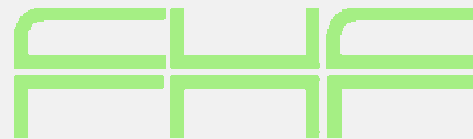


Die neun Leiden des (jungen) UML-Anwenders

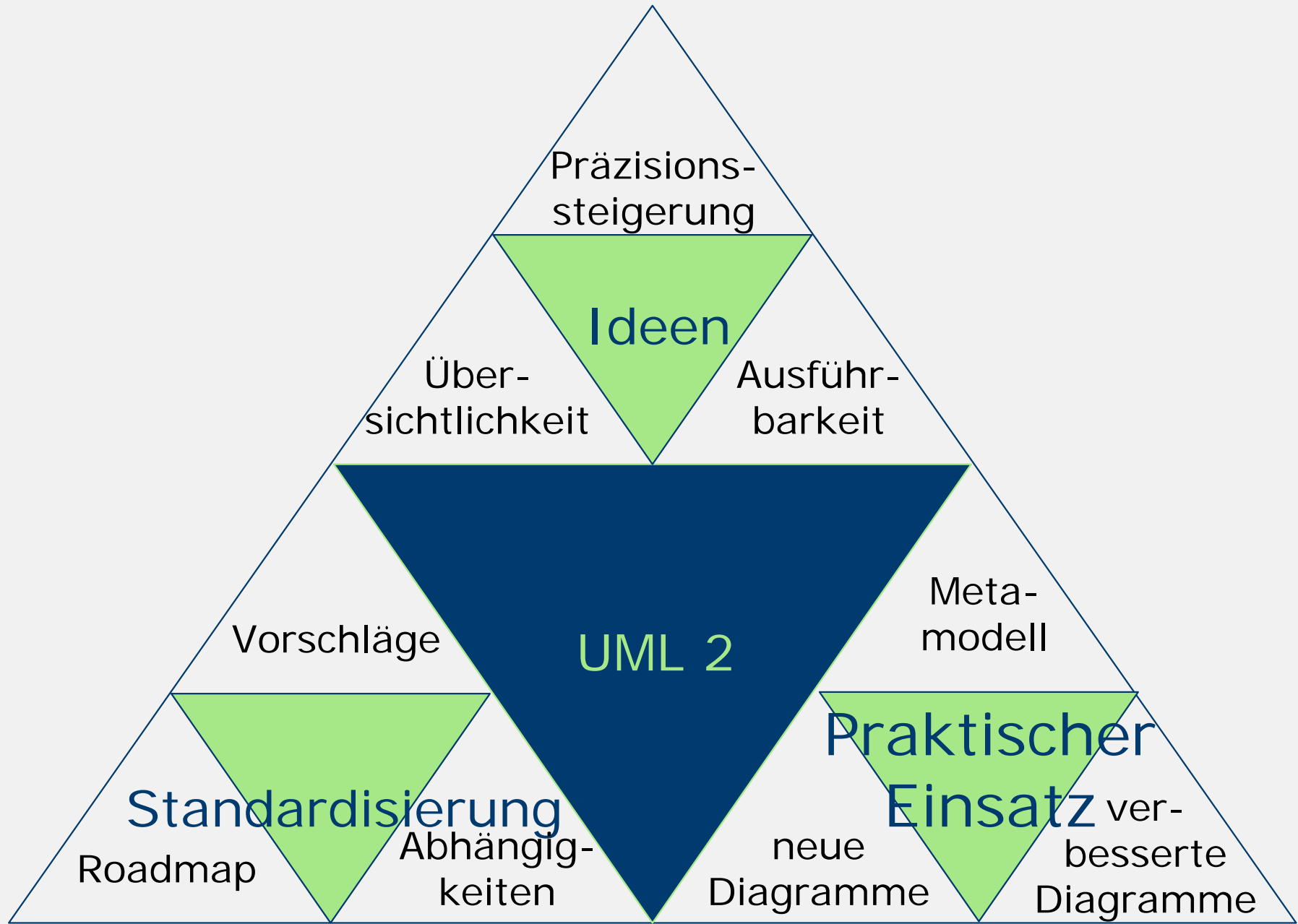


Prof. Mario Jeckle

Fachhochschule Furtwangen

mario@jeckle.de

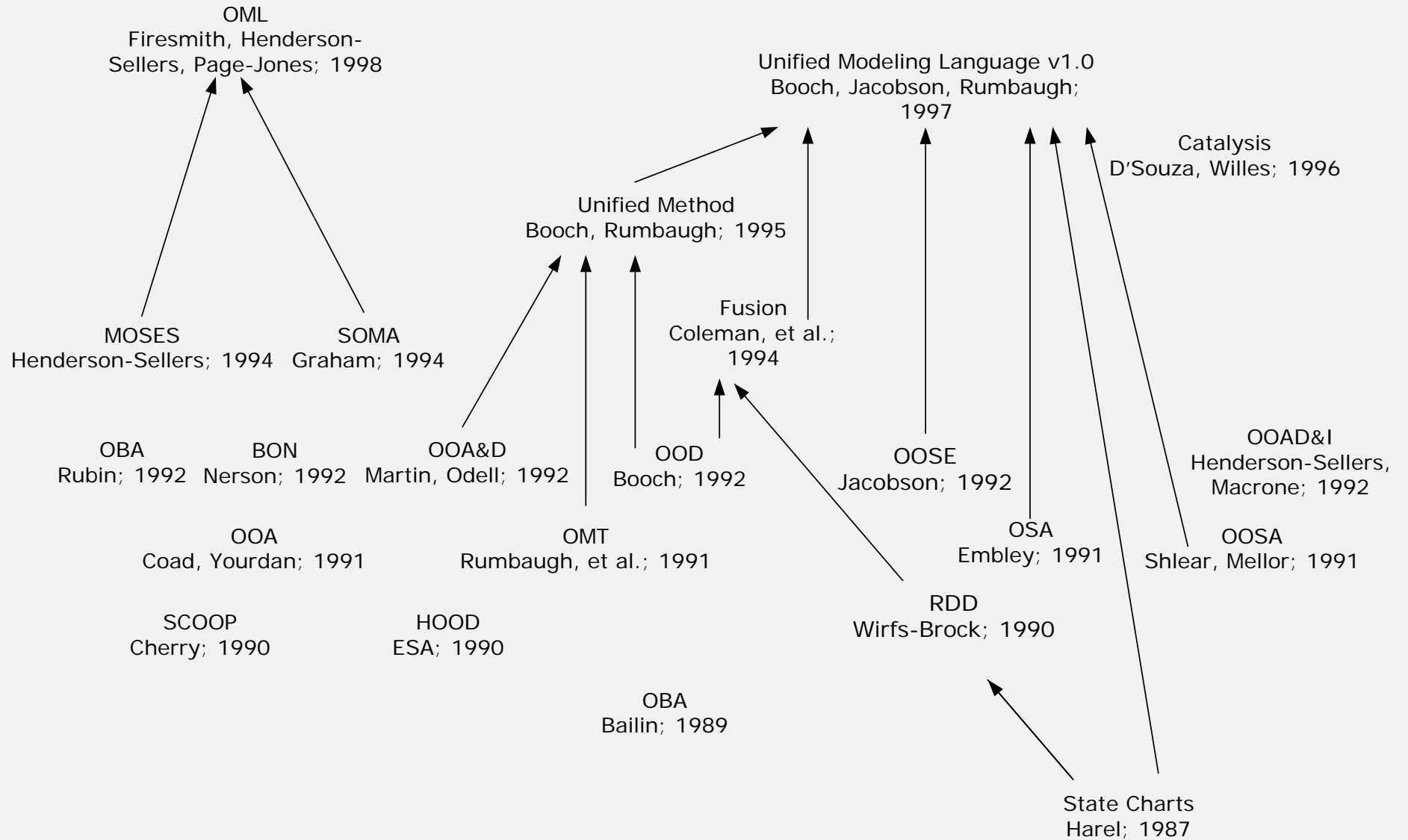
<http://www.jeckle.de>



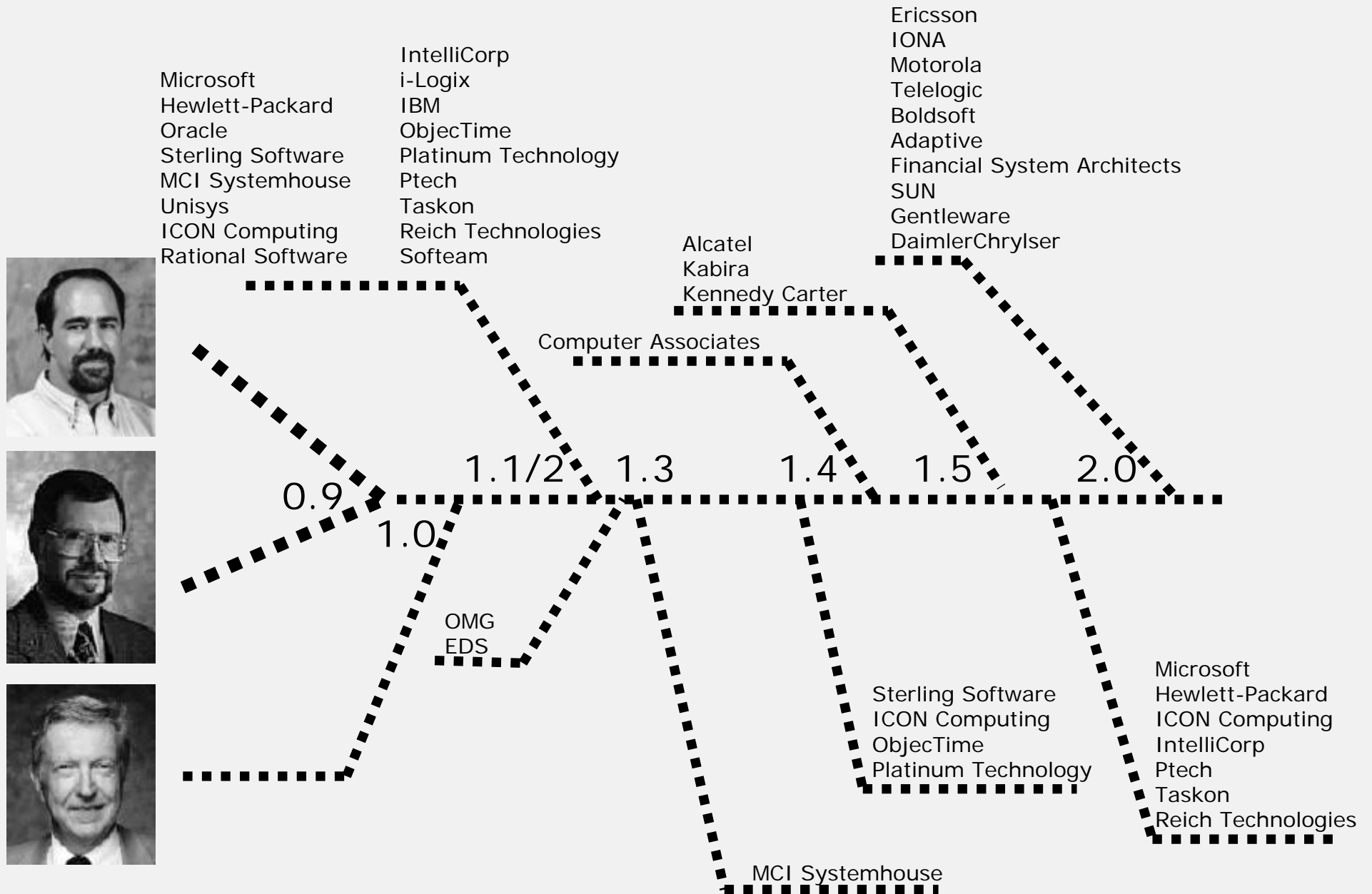
Die neun Leiden ...

- I. Mangelnde semantische Präzision
- II. Mangelnde Ausführbarkeit
- III. Fehlender Diagrammaustausch
- IV. Unnötige Sprachkomplexität
- V. Keine „eingebaute“ Anpaßbarkeit vorgesehen
- VI. Kein „echtes“ logisches Modell
- VII. *Not invented here*-Syndrom
- VIII. Keine Unterstützung aktueller Anwendungskontexte
- IX. Inkonsistentes oder gar in Teilen fehlendes Metamodell

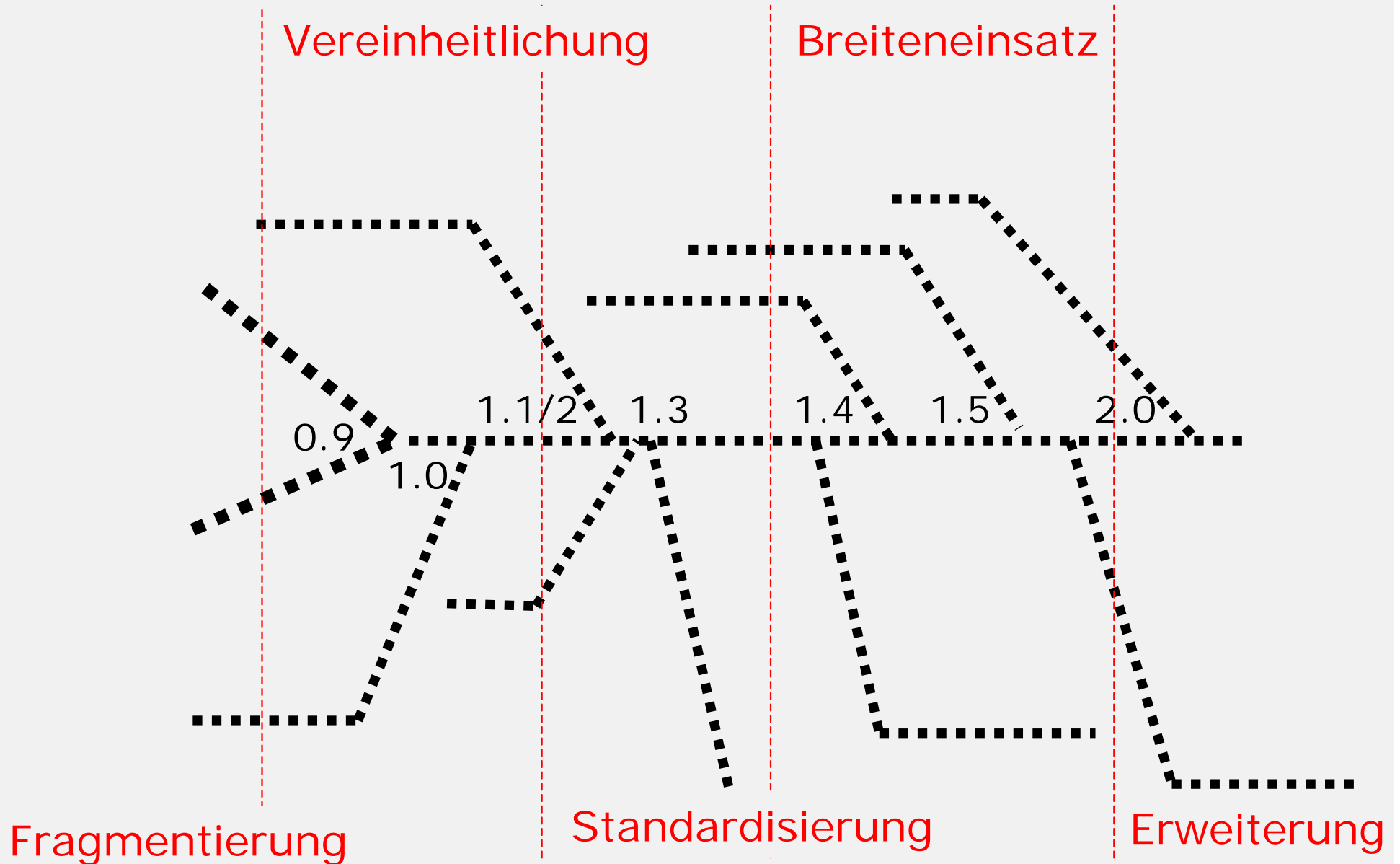
Vergangenes ...



... und Zukünftiges



Vergangenes und Zukünftiges

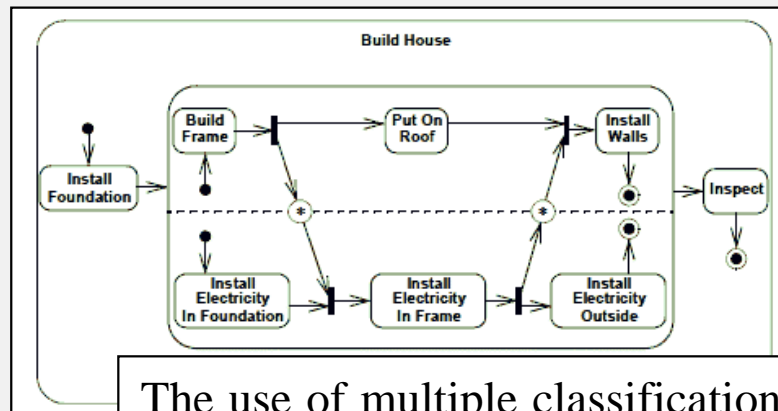


Wangelnde semantische Präzision

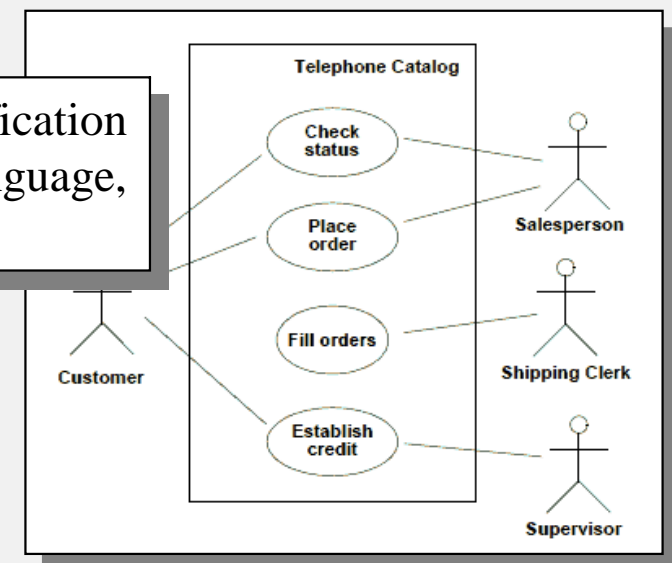


UML v1.x

- Einige Definitionen sind vage und inpräzise
- Einige Diagrammsprachen sind lediglich „Bildchen“



The use of multiple classification or dynamic classification affects the dynamic execution semantics of the language, but is not usually apparent from a static model.

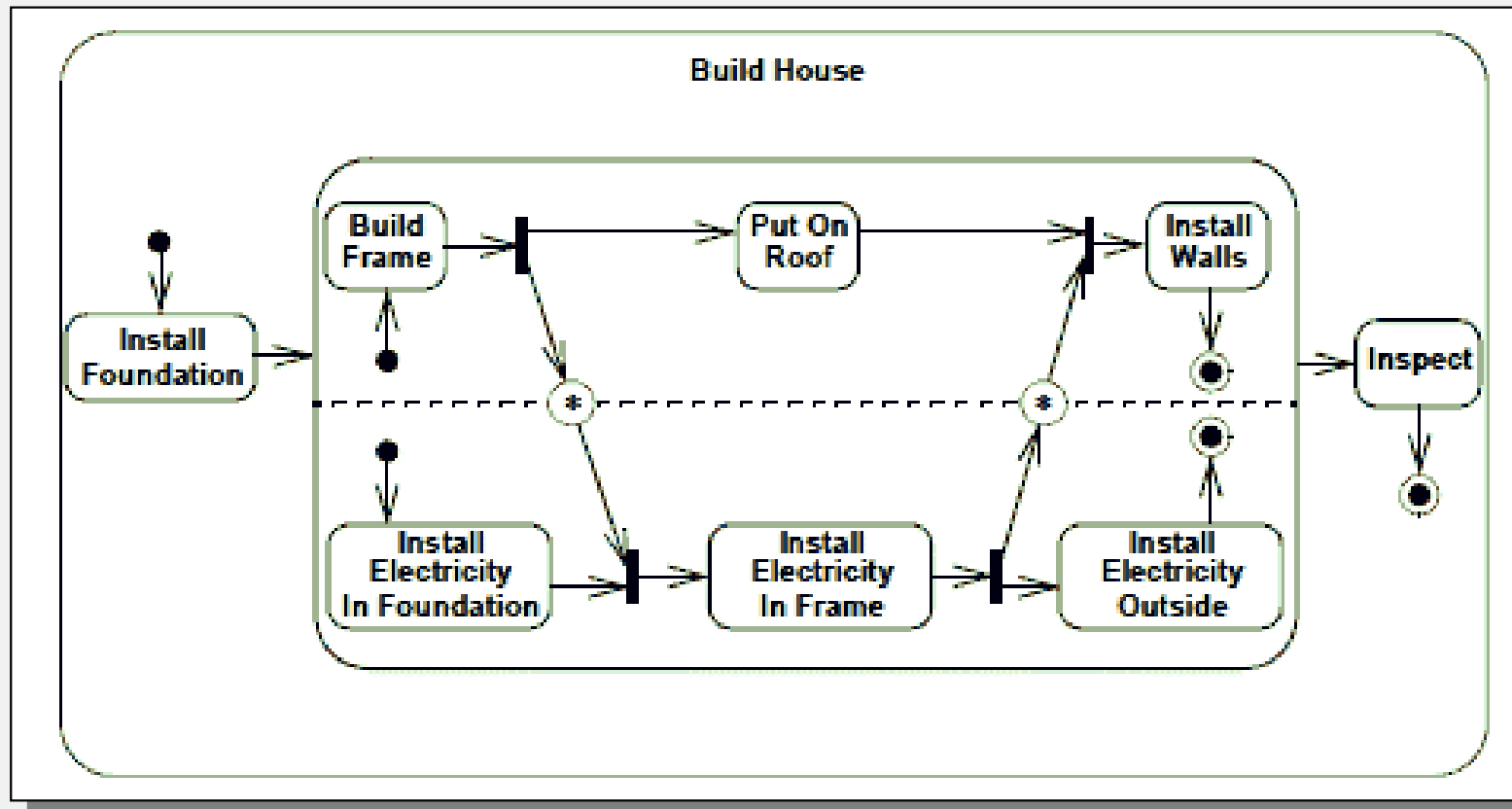


Mangelnde Ausführbarkeit



UML v1.x

- Fundierung der Aktivitätsdiagramme fehlt völlig; Eigentlich sind sie lediglich eine andere Sicht auf Zustandsmaschinen.

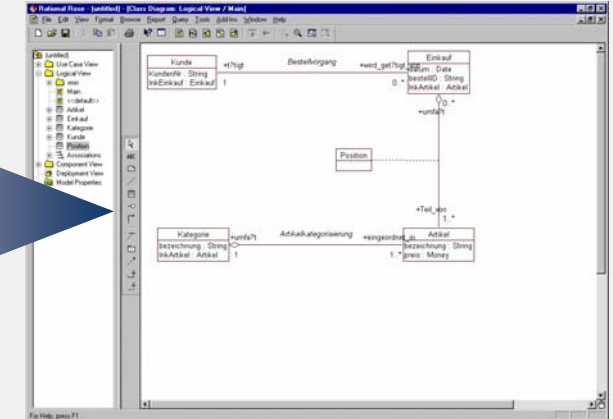
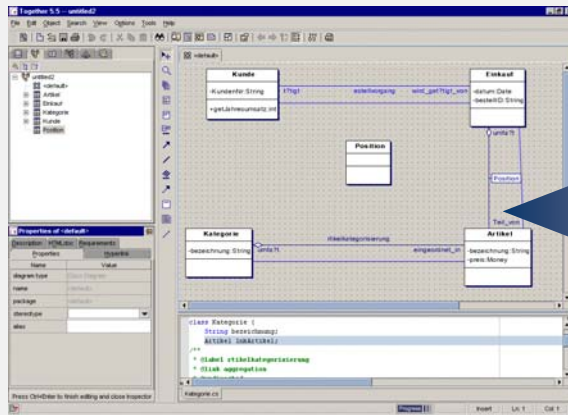


Fehlender Diagrammaustausch



UML v1.x

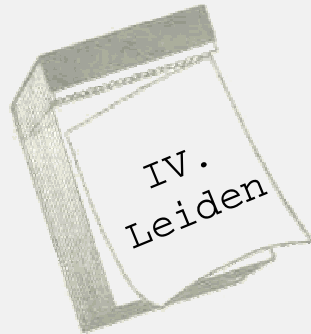
- UML-Metamodell kennt den Begriff *Diagramms* nicht
- Modellaustausch mit dem XML Metadata Interchange (XMI)-Format als Resultat davon auch nicht.



UML v2.0

- Erweitert das bestehende Metamodell
- Enthält die *Graphikinformation* (nicht die Pixeldaten)
- Kann automatisiert in Graphikdaten (GIF, PNG, SVG) übersetzt werden

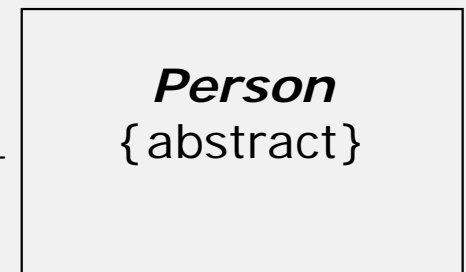
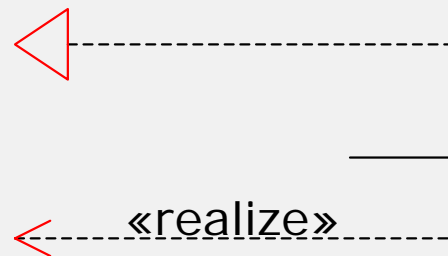
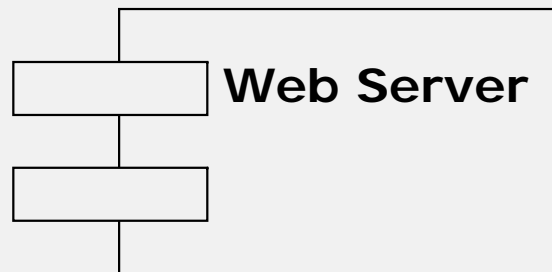
Unnötige Sprachkomplexität



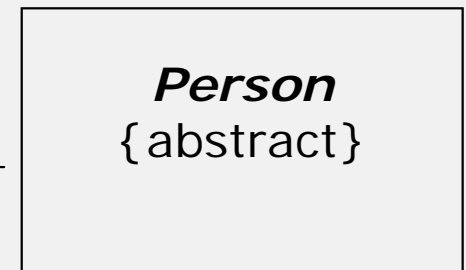
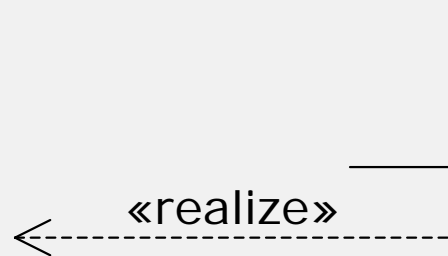
UML v1.x

- Zeichnerisch komplexe Symbolik
- Redundante nicht-orthogonale Konstrukte oder Darstellungsoptionen

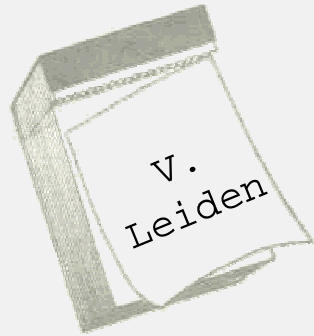
UML v1.x



UML v2



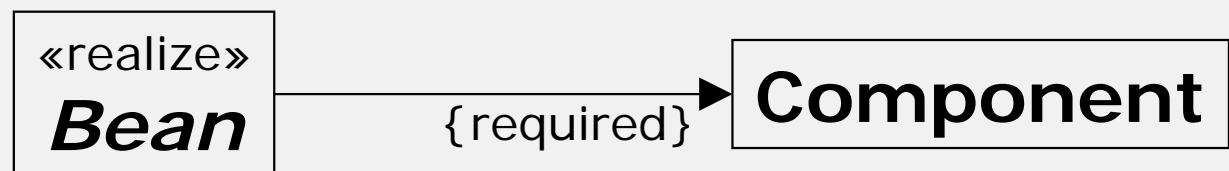
Keine „eingebaute“ Anpassbarkeit vorgesehen



UML v1.x

- Stereotypen und Eigenschaftswerte (*tagged values*) verwendet und für Verwendung durch Anwender offenstehend.
- In der Spezifikation erfolgte Verwendung liefert eine Reihe von gleichzeitigen Anpassungen bereits mit.
- Logische Gruppierung und gezielte (De-)Aktivierung dieser Erweiterungen **weder möglich noch vorgesehen**. Begriff des *Profiles* existiert zwar, findet sich jedoch nicht im Metamodell.
- Zuordnung von Stereotypen zu Modellelementen **nur textuell** (im Metamodell) oder **gar nicht** (für den Anwender) **möglich**.

UML v2

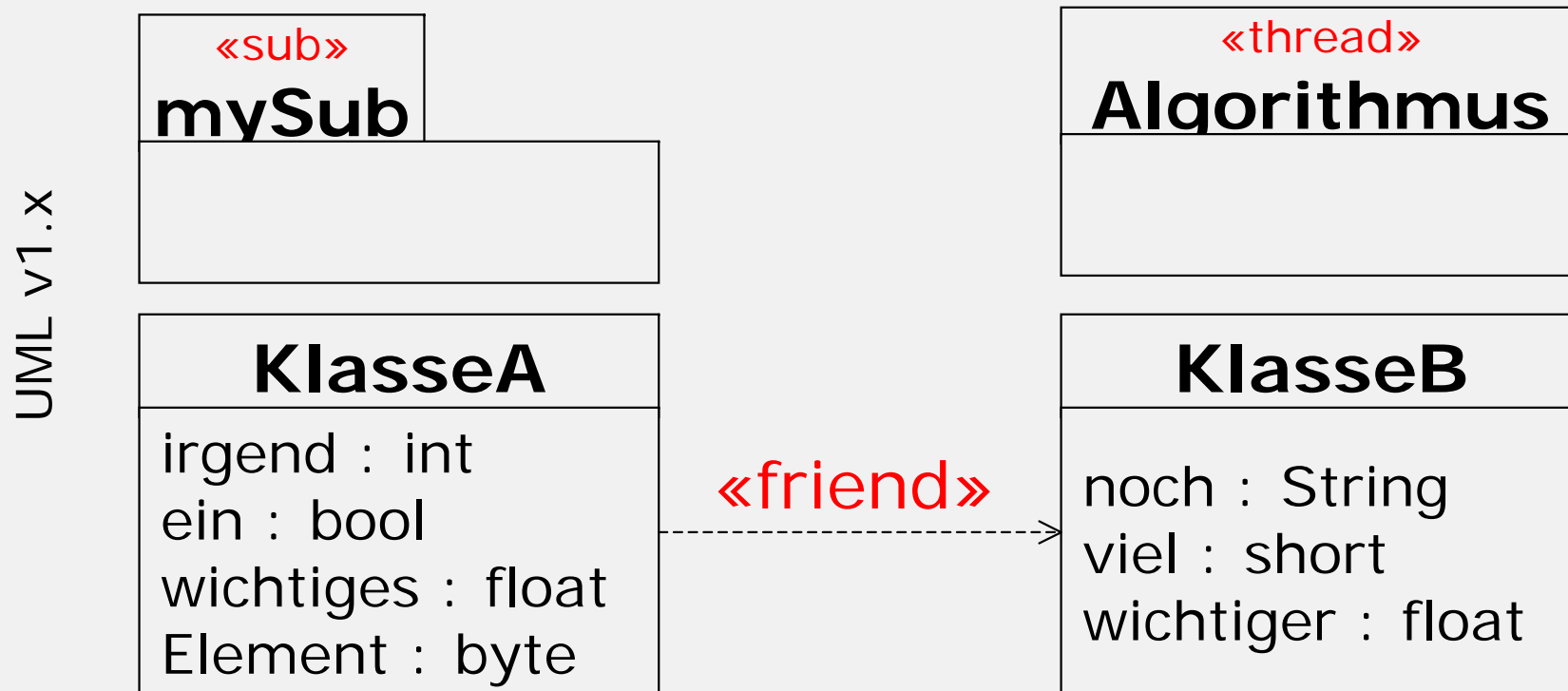


Kein „echtes“ logisches Modell



UML v1.x

- Vorhandene Sprach- und Spracherweiterungsmechanismen sind oft programmiersprachen- oder umsetzungsparadigmenspezifisch



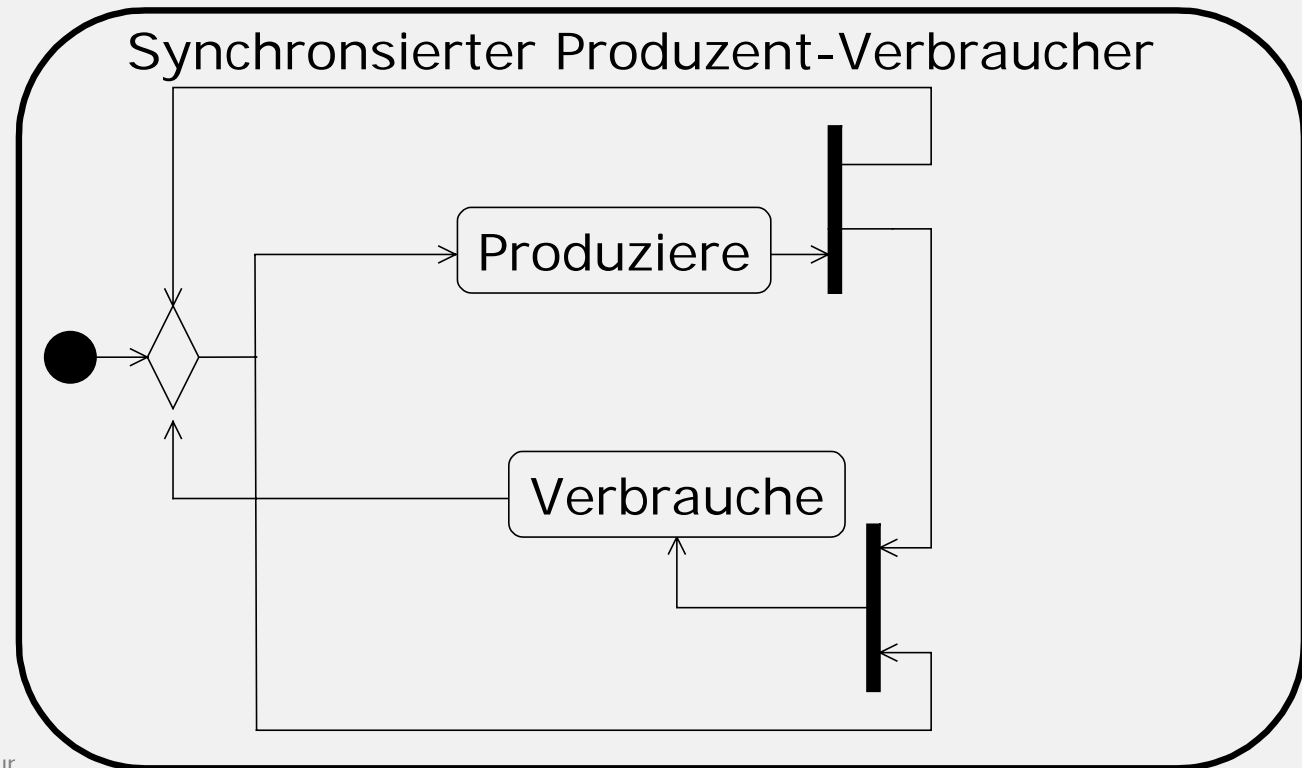
UML v1.x



Not invented here-Syndrom

UML v1.x

- UML v1.x etwas widerstrebend in der Aufnahme von Ideen, die außerhalb der OO-Gemeinde entwickelt wurden.



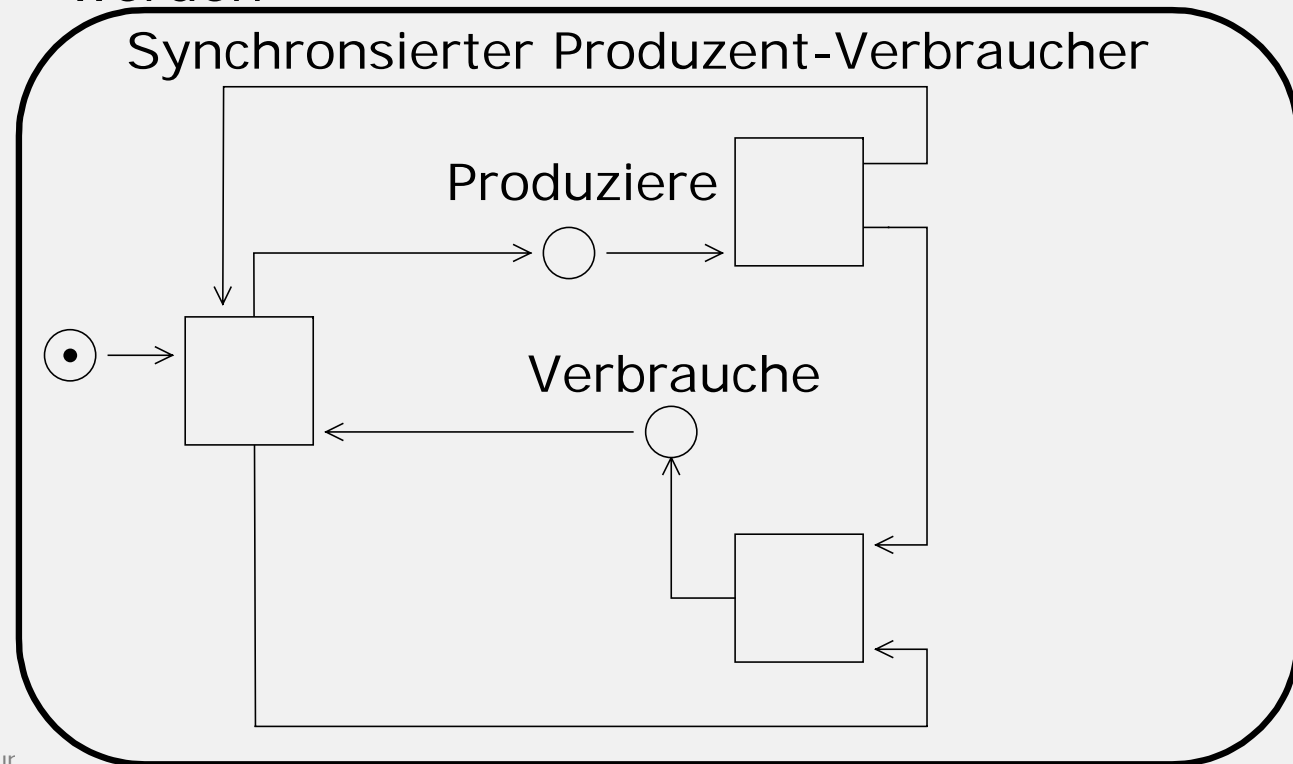
UML v2



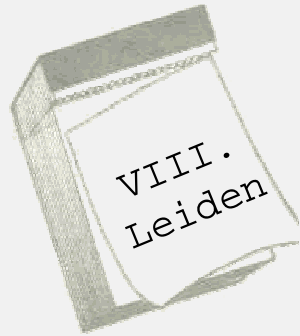
Not invented here-Syndrom

UML v2

- Aktivitäten
 - sind eigener Diagrammtyp mit eigener Metamodelrepräsentation
 - setzen auf Formalismen der PETRI-Netze auf
 - unterstützen parallele Flüsse besser
 - sind (fast) direkt ausführbar
 - können formal auf Verklemmungsfreiheit untersucht werden



Keine Unterstützung aktueller Anwendungskontexte



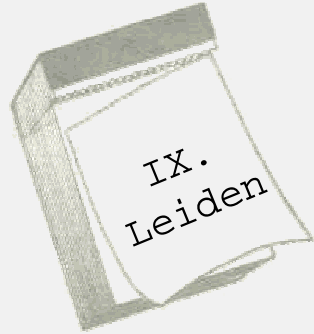
UML v1.x

- Teilweise noch sehr programmiersprachenspezifisch
 - Paketkonzept (Java)
 - Laufzeitsemantik (C++)
- Aktuelle Anwendungsgebiete nicht berücksichtigt
 - Web-Anwendungen (Deployment- und Container-Semantik)
 - Echtzeitanwendungen
 - Komponenten

UML v2

- Von den konkreten Programmiersprachen weiter abstrahiert (z.B. *Manifestation* statt *Implementierung*) (Ziel: logisches Modell)
- Berücksichtigung von Echtzeitanforderungen
- Neu: *Kommunikationsdiagramm* stellt Interaktionen im Lebenszyklus einzelner Komponenten in den Vordergrund. (Im Prinzip: Eine Variante des Sequenzdiagramms.)
- Als Alternative zu der graphischen Darstellung von Sequenzdiagrammen stehen tabellenartige Texte zur Verfügung

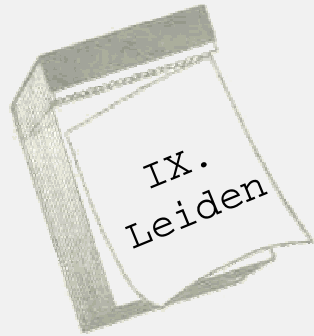
Inkonsistentes oder gar in Zeilen fehlendes Metamodell



UML v1.x

- Bereits erwähnt:
 - Diagrammaustausch
 - Profile
- Zusätzlich:
 - Viele Konzepte (Generalisierung, Abhängigkeit, Eigenschaften, Abstraktion) nicht verallgemeinert
 - Kein Metamodell für die *Object Constraint Language*
 - Einzelne Primitive (Note) bleiben völlig unberücksichtigt

Inkonsistentes oder gar in Zeilen fehlendes Metamodell



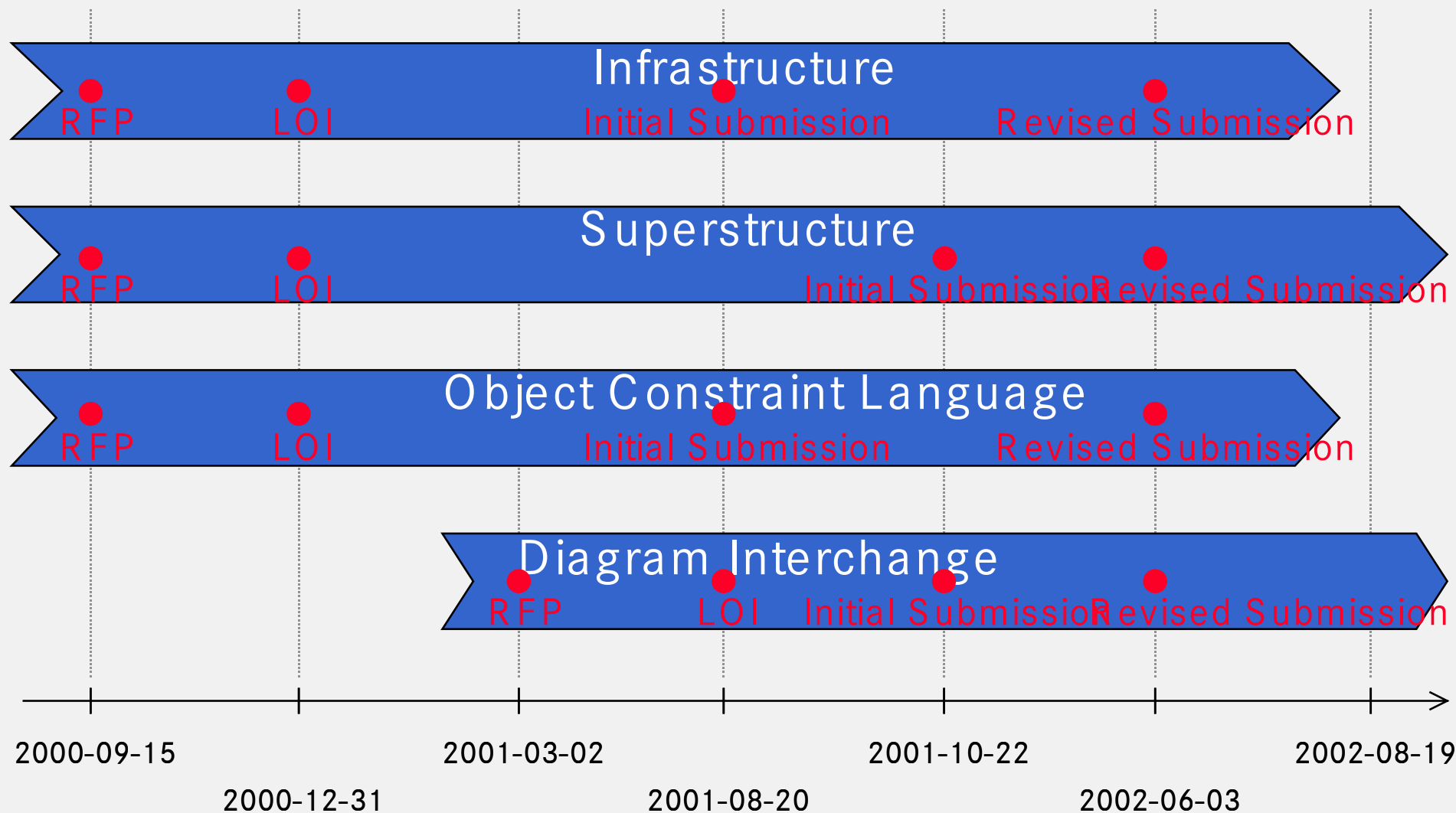
UML v2

- Bereits erwähnt:
 - Diagrammaustausch
 - Profile... im Metamodell berücksichtigt.
- Zusätzlich:
 - Basiskonzepte grundlegend Überarbeitet
 - Verallgemeinert
 - Wiederverwendet (im Metamodell)
 - = > fast vollständig neuem, kompakten Metamodell
 - Metamodell für die *Object Constraint Language*
 - Einzelne Primitive (Note) bleiben immernoch völlig unberücksichtigt

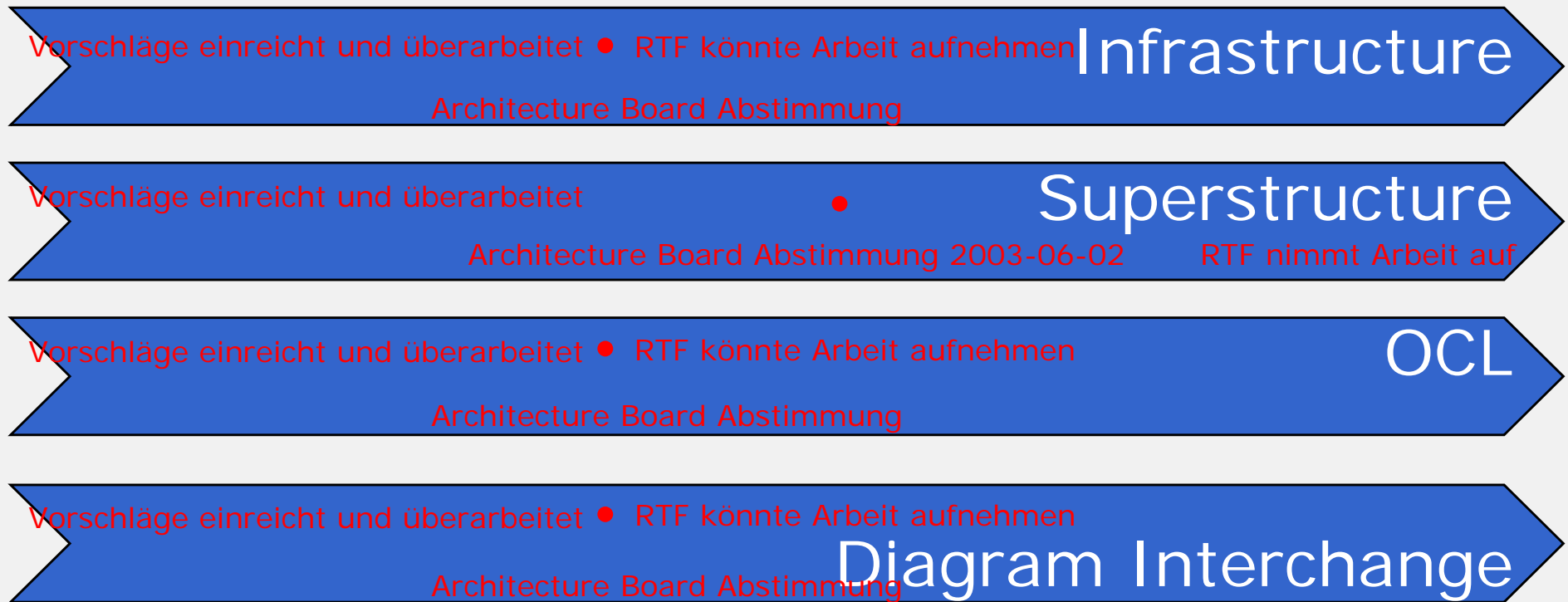
Vorschläge zur UML 2

- *Superstructure* und *Infrastructure*:
Ausgereiftester und mit breiter Unterstützung
bedachter Vorschlag durch die sog. „*UML 2
Partners*“:
 - Mitglieder:
Alcatel, Computer Associates, Ericsson,
Hewlet-Packard, IONA, Kabira
Technologies, Motorola, Oracle, Rational
Software, SOFTEAM, Telelogic, and Unisys
 - Unterstützer:
Advanced Concepts Center, Ceira
Technologies, Commissariat à L’Energie
Atomique, Compuware, DaimlerChrysler,
Embarcardero Technolgies, Enea Business
Software, France Telecom, ...

Roadmap ... die Planung

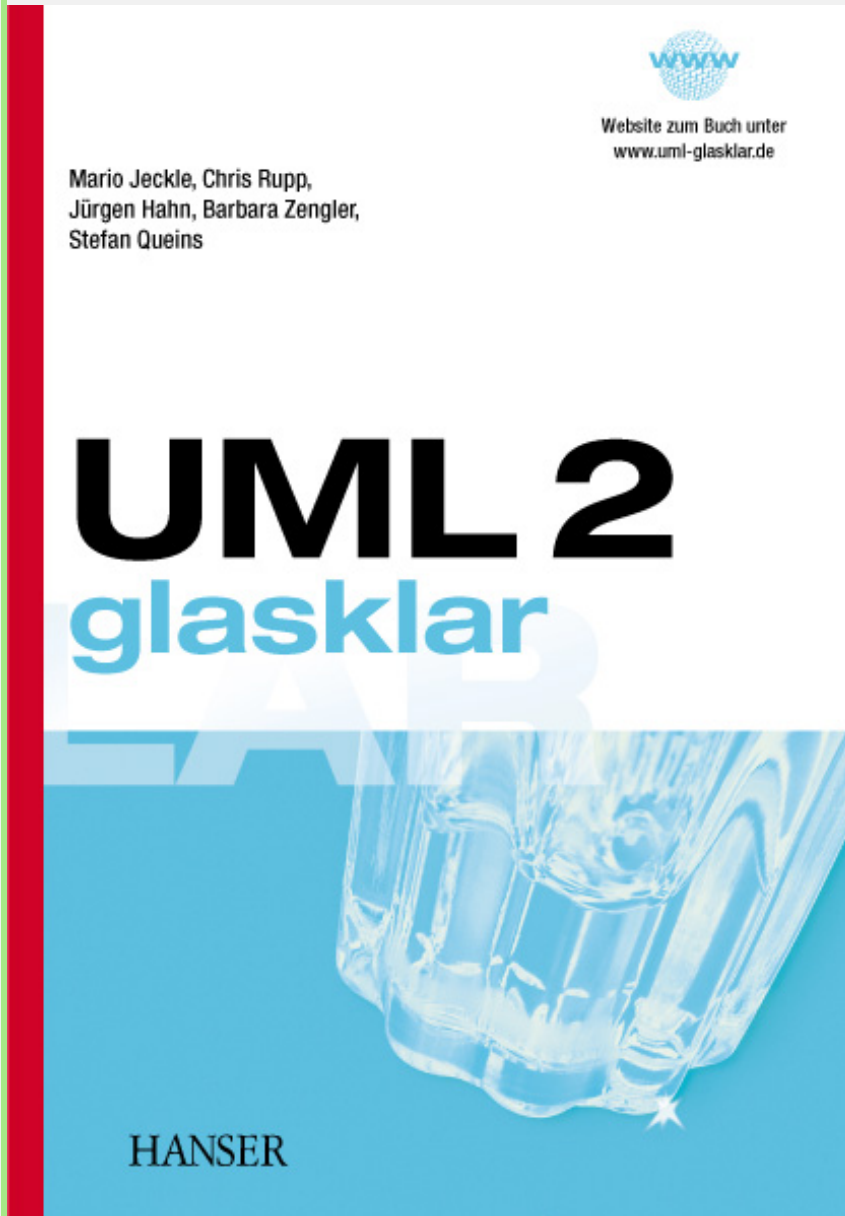


Roadmap ... die Realität



Inzwischen sind die überarbeiteten Einreichungen aller vier Teile angenommen und die Revision Task Force nimmt ihre Arbeit zur Komplettierung der UML 2 auf.

Wenn Sie immer noch nicht genug haben ...



UML 2 glasklar

Das Buch zur Sprache

- Vollständige Behandlung der UML 2
- Alle Diagrammtypen
- Übersicht der Änderungen seit UML 1
- Grundkonzepte ohne Ballast erklärt
- Viele Anwendungsbeispiele
- Language Binding
 - C++
 - Java (incl. Generics/Java 1.5)
 - C# (incl. Generics/C# 2.0)

