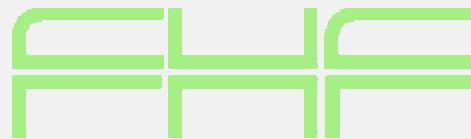


Unified Modeling Language 2.0

Einblicke in die neue Version der Standardmodellierungssprache

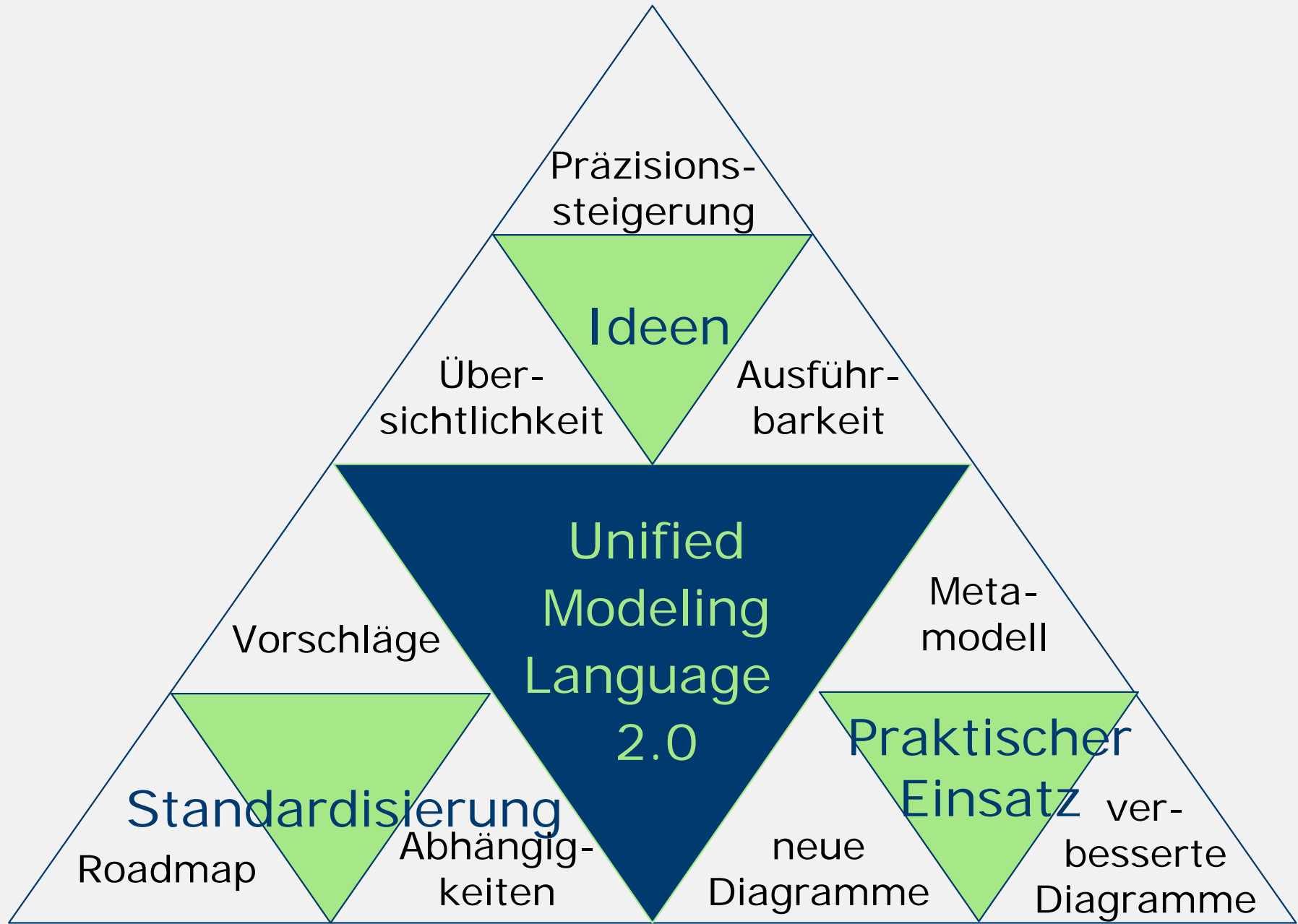


Prof. Mario Jeckle

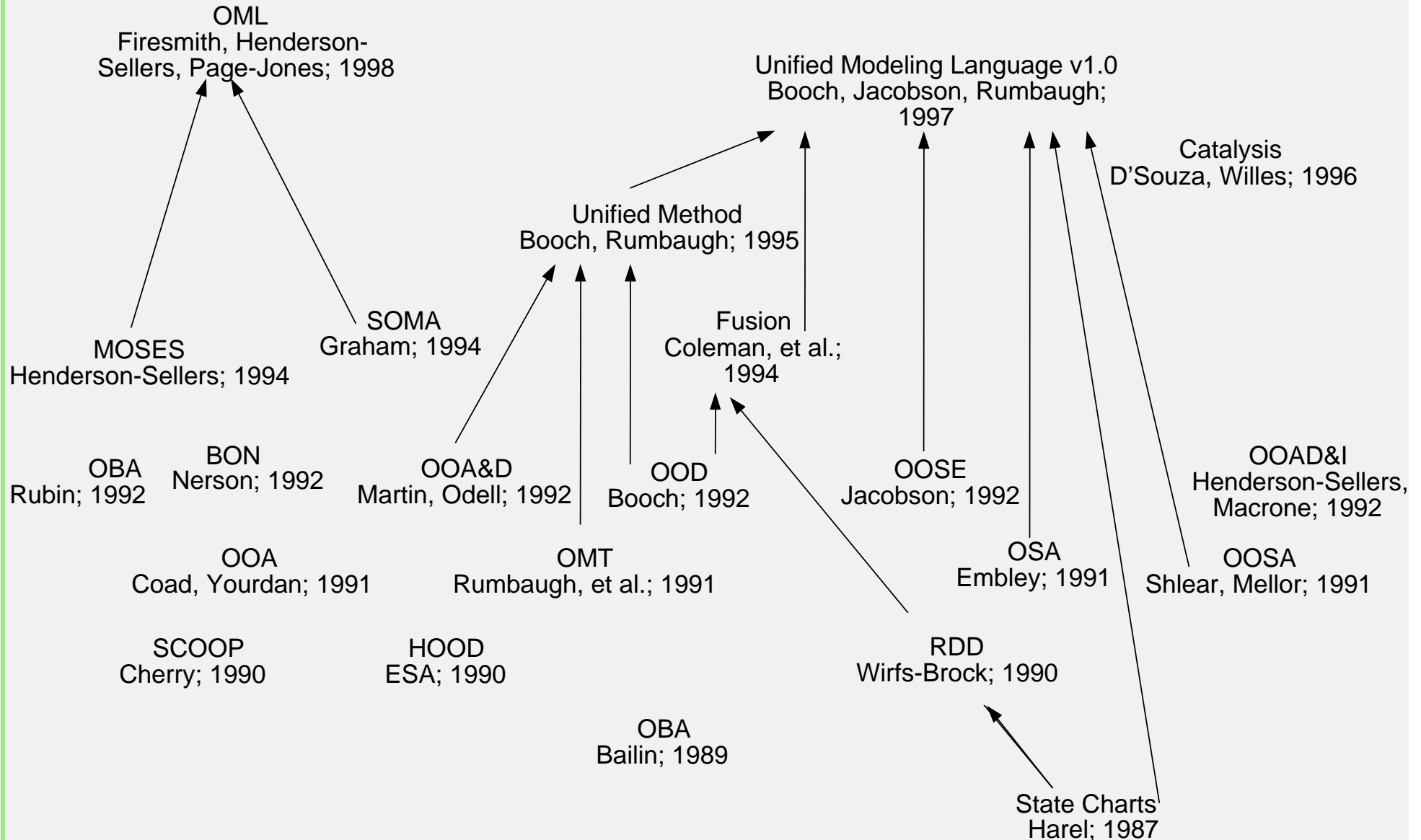
Fachhochschule Furtwangen

mario@jeckle.de

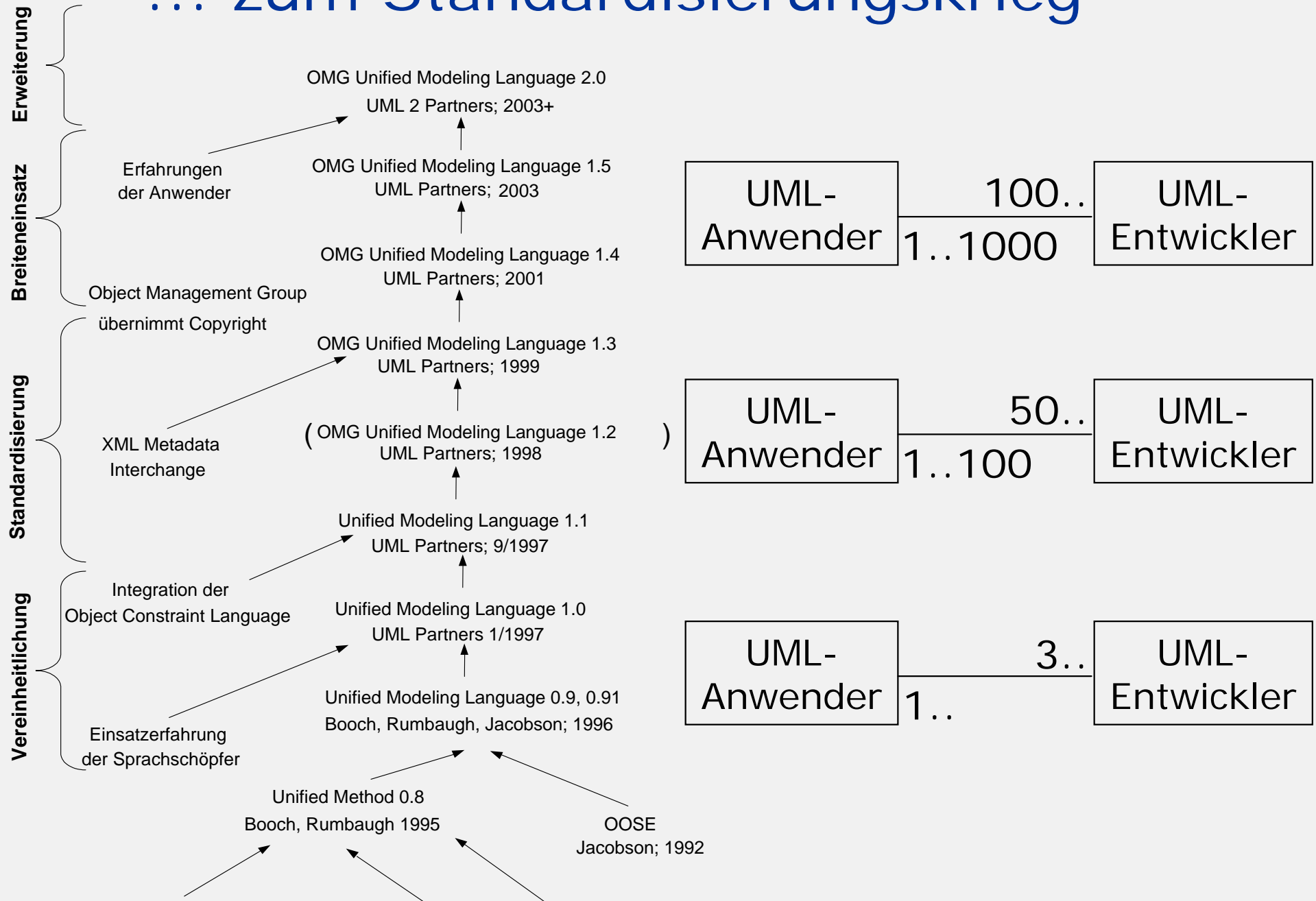
<http://www.jeckle.de>



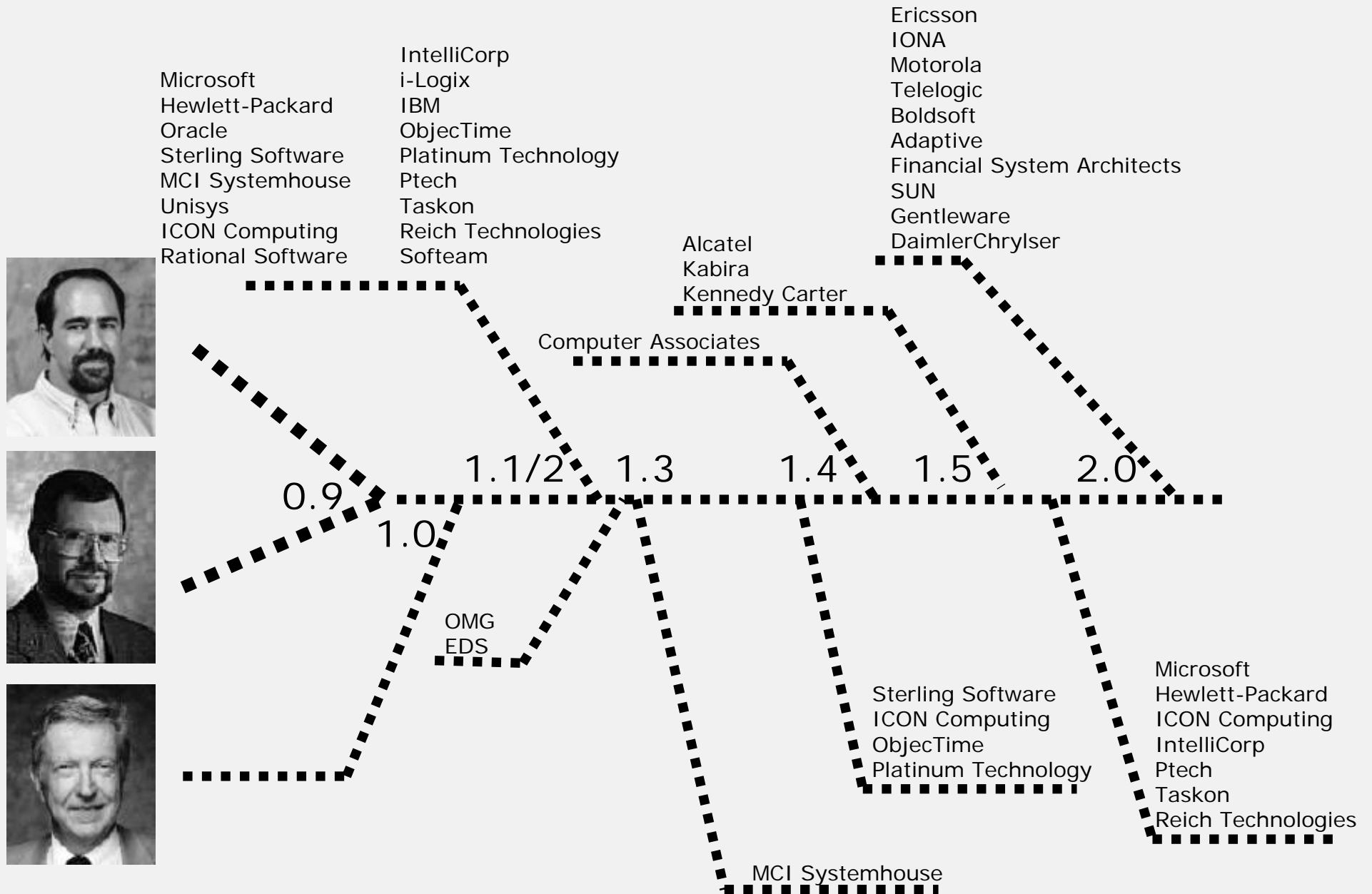
Vom Methodenkrieg ...



... zum Standardisierungskrieg



Der Weg zur UML 2



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)

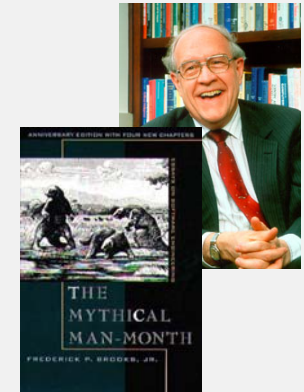
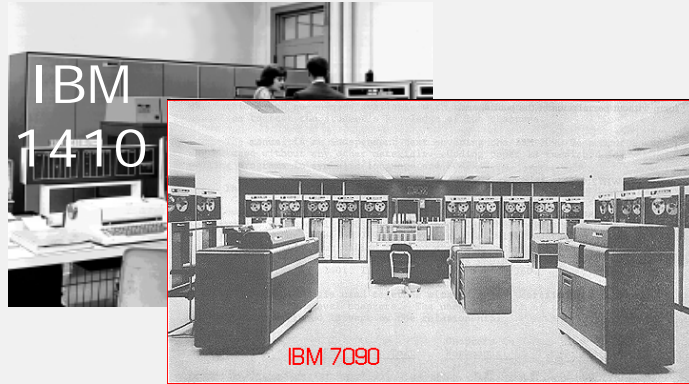
- **Erfahrung**

- 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)

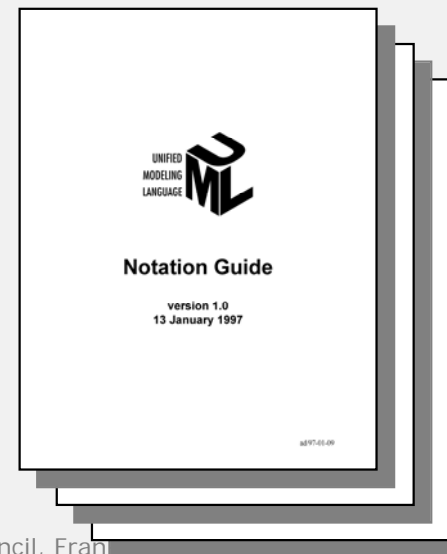
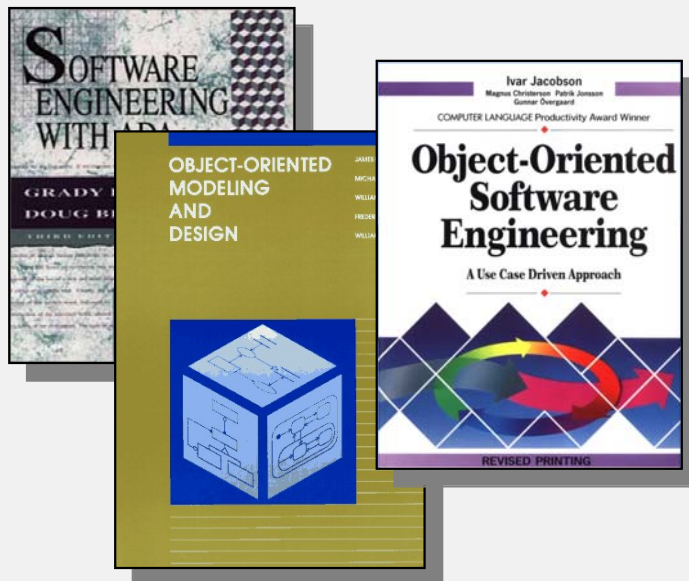
Der Second-System-Effekt



1. System

Adde parvum paravo magnus acervus erit

2. System

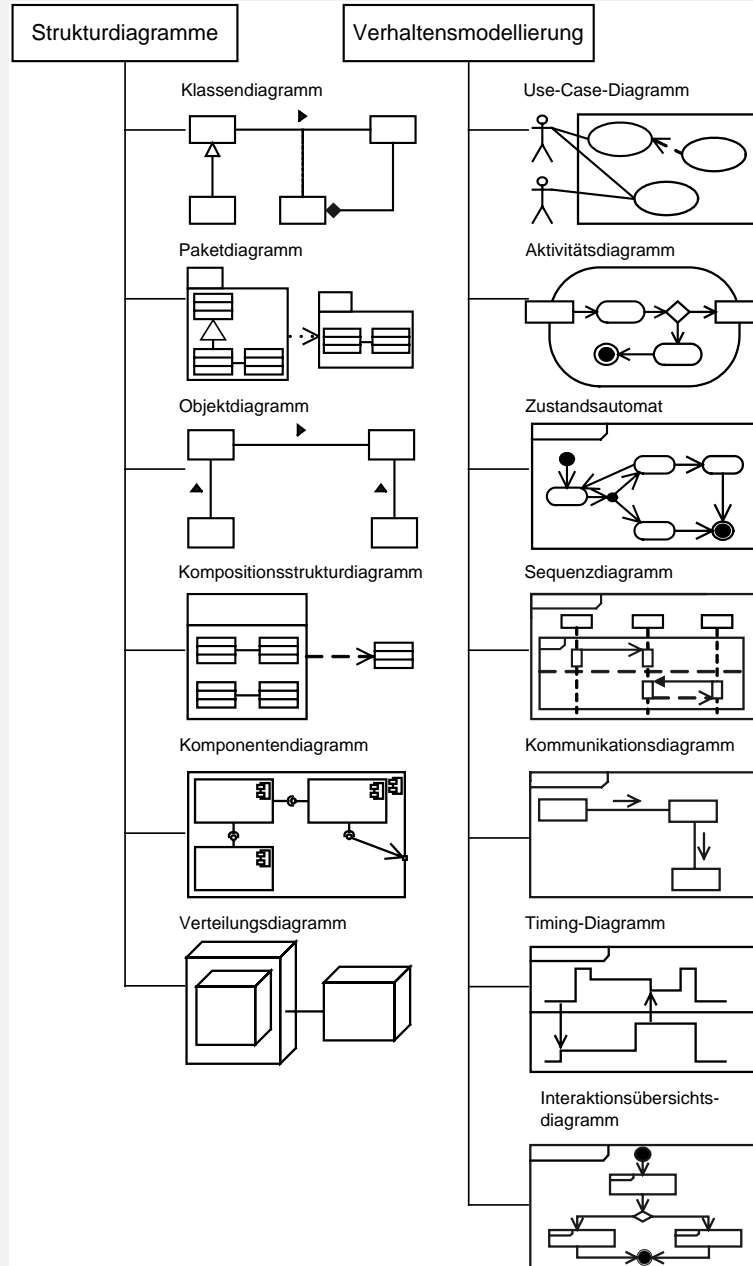


Ziele der UML 2

- **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 im Überblick

Die Diagrammsprachen



Nur marginale Änderungen

- Klassendiagramm
- Use-Case-Diagramm
- Objektdiagramm

Klein(-e Änderungen) aber oho:

- Paketdiagramm
- Verteilungsdiagramm

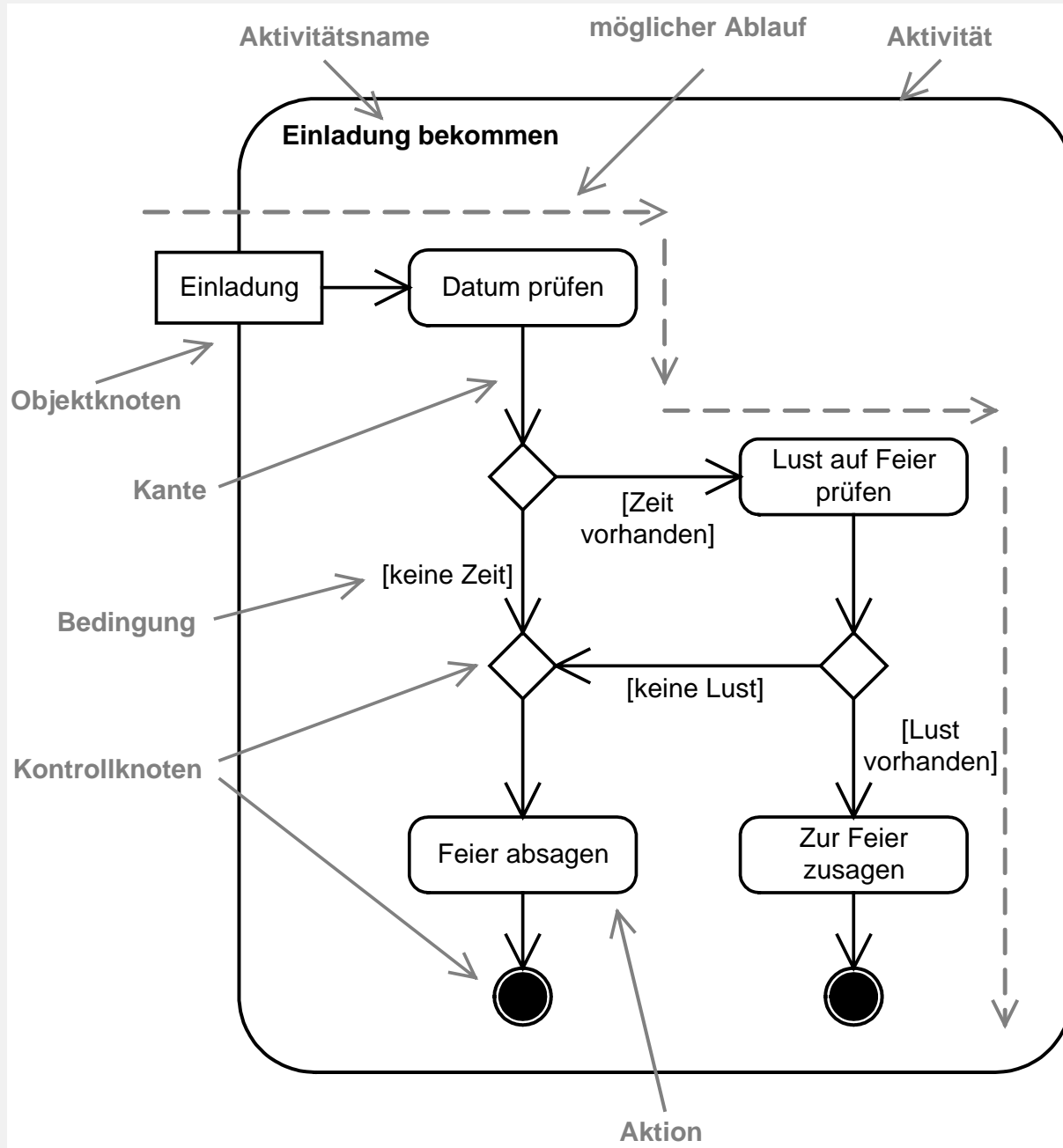
Massiv und tiefgreifend verändert:

- Aktivitätsdiagramm
- Sequenzdiagramm
- Zustandsautomat

Vollständig neu:

- Kompositionsstrukturdiagramm
- Interaktionsübersichtsdiagramm
- Timing-Diagramm
- Kommunikationsdiagramm

UML 2 im Überblick



Nur marginale Änderungen

- Klassendiagramm
- Use-Case-Diagramm
- Objektdiagramm

Klein(-e Änderungen) aber oho:

- Paketdiagramm
- Verteilungsdiagramm

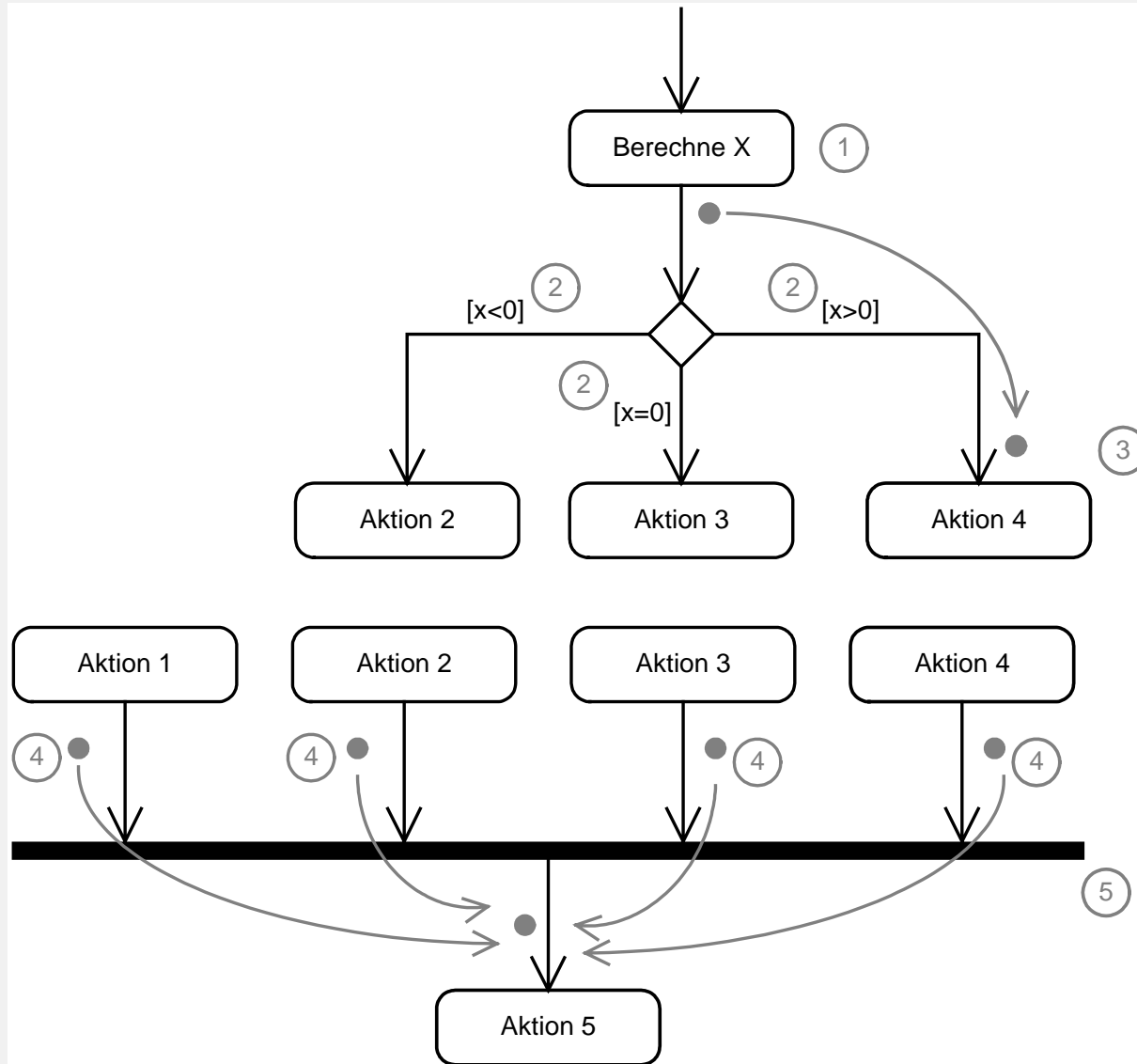
Massiv und tiefgreifend verändert:

- Aktivitätsdiagramm
- Sequenzdiagramm
- Zustandsautomat

Vollständig neu:

- Kompositionsstrukturdiagramm
- Interaktionsübersichtsdiagramm
- Timing-Diagramm
- Kommunikationsdiagramm

UML 2 im Überblick



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 im Überblick

Nur marginale Änderungen

- Klassendiagramm
- Use-Case-Diagramm
- Objektdiagramm

Klein(-e Änderungen) aber oho:

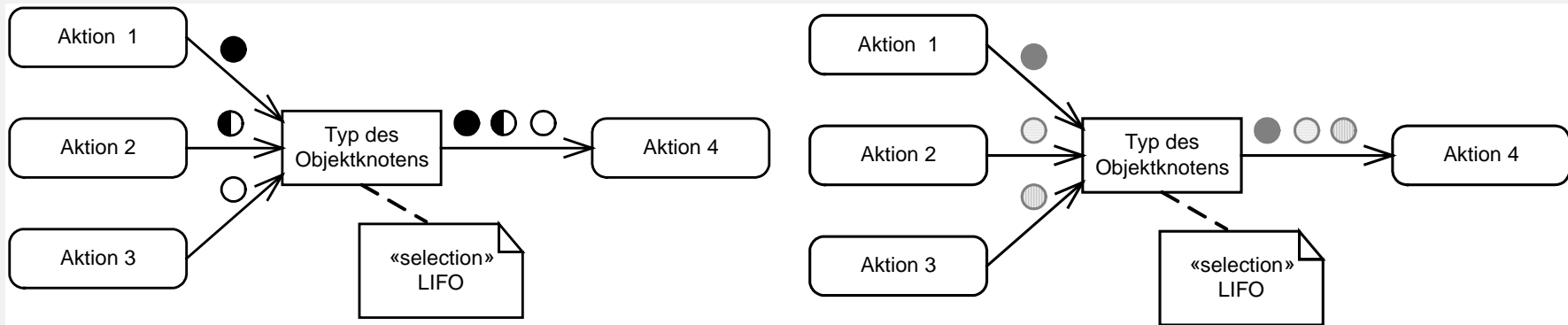
- Paketdiagramm
- Verteilungsdiagramm

Massiv und tiefgreifend verändert:

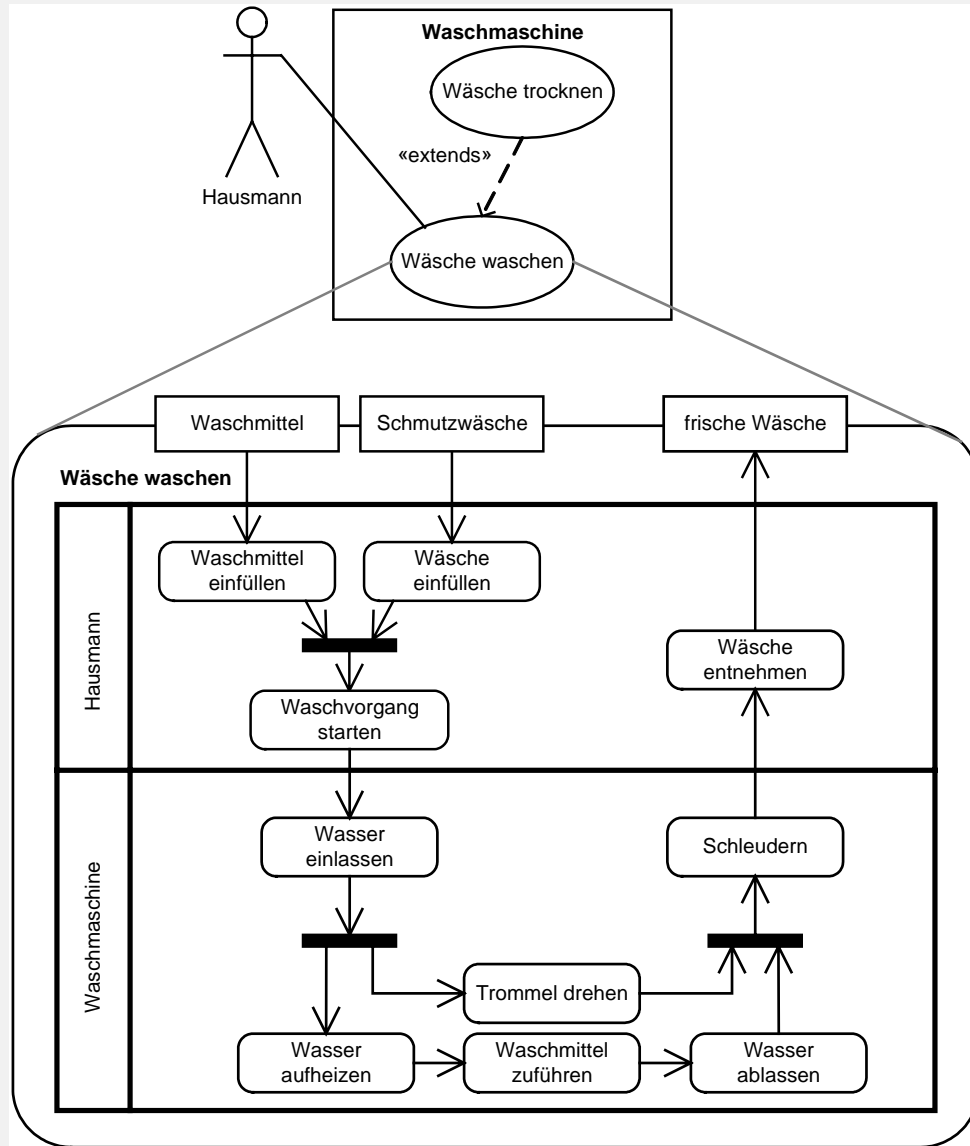
- Aktivitätsdiagramm
- Sequenzdiagramm
- Zustandsautomat

Vollständig neu:

- Kompositionsstrukturdiagramm
- Interaktionsübersichtsdiagramm
- Timing-Diagramm
- Kommunikationsdiagramm



UML 2 im Überblick



Nur marginale Änderungen

- Klassendiagramm
- Use-Case-Diagramm
- Objektdiagramm

Klein(-e Änderungen) aber oho:

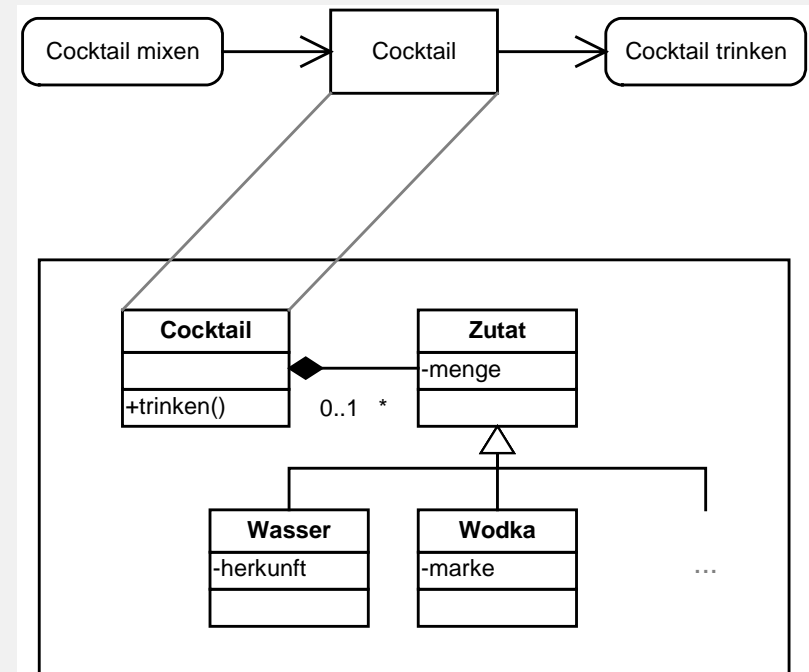
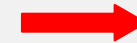
- Paketdiagramm
- Verteilungsdiagramm

Massiv und tiefgreifend verändert:

- Aktivitätsdiagramm
- Sequenzdiagramm
- Zustandsautomat

Vollständig neu:

- Kompositionsstrukturdiagramm
- Interaktionsübersichtsdigramm
- Timing-Diagramm
- Kommunikationsdiagramm



UML 2 im Überblick

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 1.x	UML 2
Aktivitätsdiagramm als „Sonderform“ des Zustandsdiagramms.	Aktivitäten sind unabhängig von Zustandsautomaten.
Aktivitätsdiagramme entsprechen in der Struktur den Zustandsdiagrammen.	Aktivitäten verwenden eine den Petrinetzen ähnliche Semantik (Token-Konzept).
Das komplette Diagramm heißt Aktivitätsdiagramm.	Das komplette Diagramm wird als Aktivität bezeichnet.
Einzelne Schritte im Aktivitätsdiagramm werden Aktivitäten genannt.	Die Schritte im Ablauf werden Aktionen genannt und beschreiben einen Verhaltensaufwurf und nicht das Verhalten selber. Daher können aus einer Aktion eine neue Aktivität, eine Interaktion, ein Use Case, ... aufgerufen werden.
Aktivitäten haben keine Vor- oder Nachbedingungen.	Aktionen können mit Vor- und Nachbedingungen verknüpft werden.
Notationssymbole für Zustand und Aktivität unterschiedlich: 	Die Notation von Aktionen entspricht der Notation von Zuständen der UML 1.x 
In einem Aktivitätsdiagramm ist nur ein Anfangszustand erlaubt.	Es sind mehrere Startknoten erlaubt. Das bedeutet, dass parallele Abläufe gestartet werden können.

UML 2 im Überblick

UML 1.x	UML 2
Ein Endzustand beendet den gesamten Ablauf im Aktivitätsdiagramm.	Es wird unterschieden zwischen dem Endknoten für Aktivitäten, der die gesamte Aktivität beendet, und dem Endknoten für Abläufe, der nur einen Ablaufstrang beendet.
Aktivitätsdiagramme haben keine Ein- und Ausgabeparameter.	Aktivitäten können Ein- und Ausgabeparameter enthalten; diese können auch den Start- und den Endknoten ersetzen.
Es gibt Objektzustände, die semantisch äußerst vage definiert sind.	Die UML 2 führt Objektknoten zur besseren Modellierung der Objektflüsse ein.
Objektflüsse und Kontrollflüsse werden durch Transitionen modelliert.	Objektflüsse und Kontrollflüsse sind explizit, die Übergänge zwischen den Aktionen werden als Kanten bezeichnet.
Parallele Stränge müssen zusammengeführt werden.	Parallele Abläufe müssen <i>nicht</i> wieder zusammengeführt werden.
Swimlanes sind immer eindimensional.	Aktivitätsbereiche können hierarchisch oder multidimensional sein.
	Neue Notationselemente: Strukturierte Knoten Mengenverarbeitungsknoten Entscheidungsknoten Schleifenknoten Datenspeicher und Bufferknoten Unterbrechungsbereich Parametersatz
JumpHandler	ExceptionHandler

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 im Überblick

Nur marginale Änderungen

- Klassendiagramm
- Use-Case-Diagramm
- Objektdiagramm

Klein(-e Änderungen) aber oho:

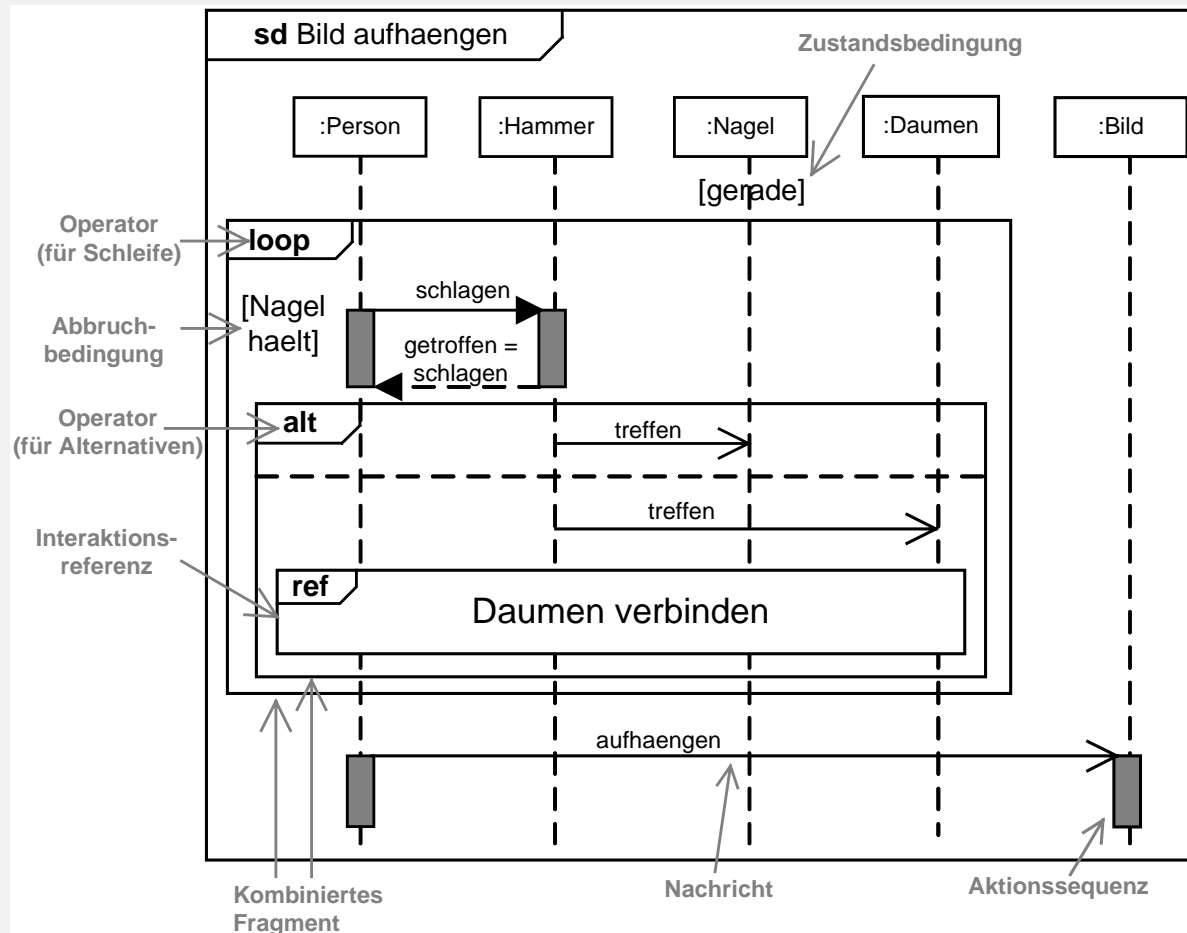
- Paketdiagramm
- Verteilungsdiagramm

Massiv und tiefgreifend verändert:

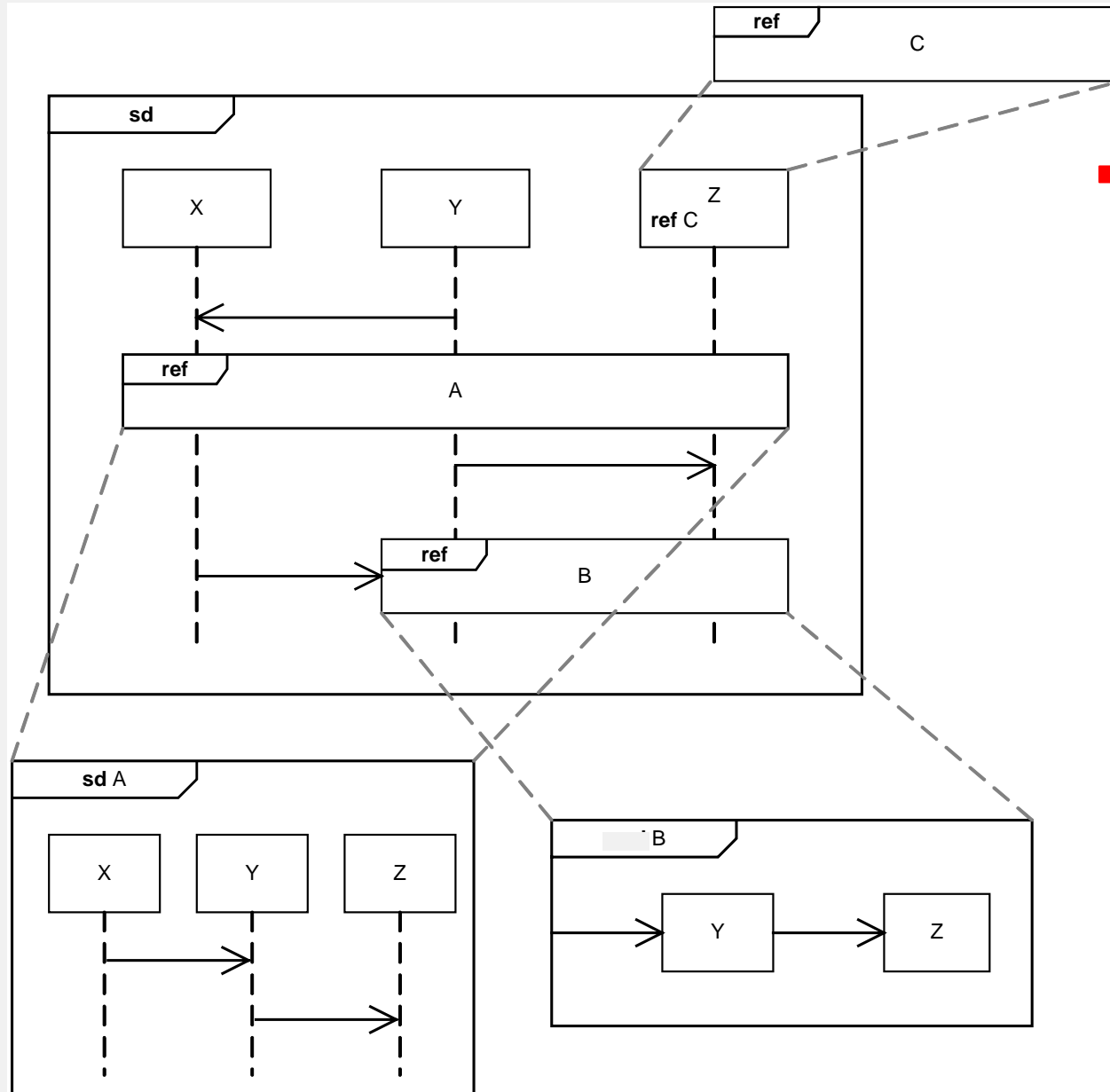
- Aktivitätsdiagramm
- Sequenzdiagramm
- Zustandsautomat

Vollständig neu:

- Kompositionsstrukturdiagramm
- Interaktionsübersichtsdiagramm
- Timing-Diagramm
- Kommunikationsdiagramm



UML 2 im Überblick



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 im Überblick

Nur marginale Änderungen

- Klassendiagramm
- Use-Case-Diagramm
- Objektdiagramm

Klein(-e Änderungen) aber oho:

- Paketdiagramm
- Verteilungsdiagramm

Massiv und tiefgreifend verändert:

- Aktivitätsdiagramm
- Sequenzdiagramm
- Zustandsautomat

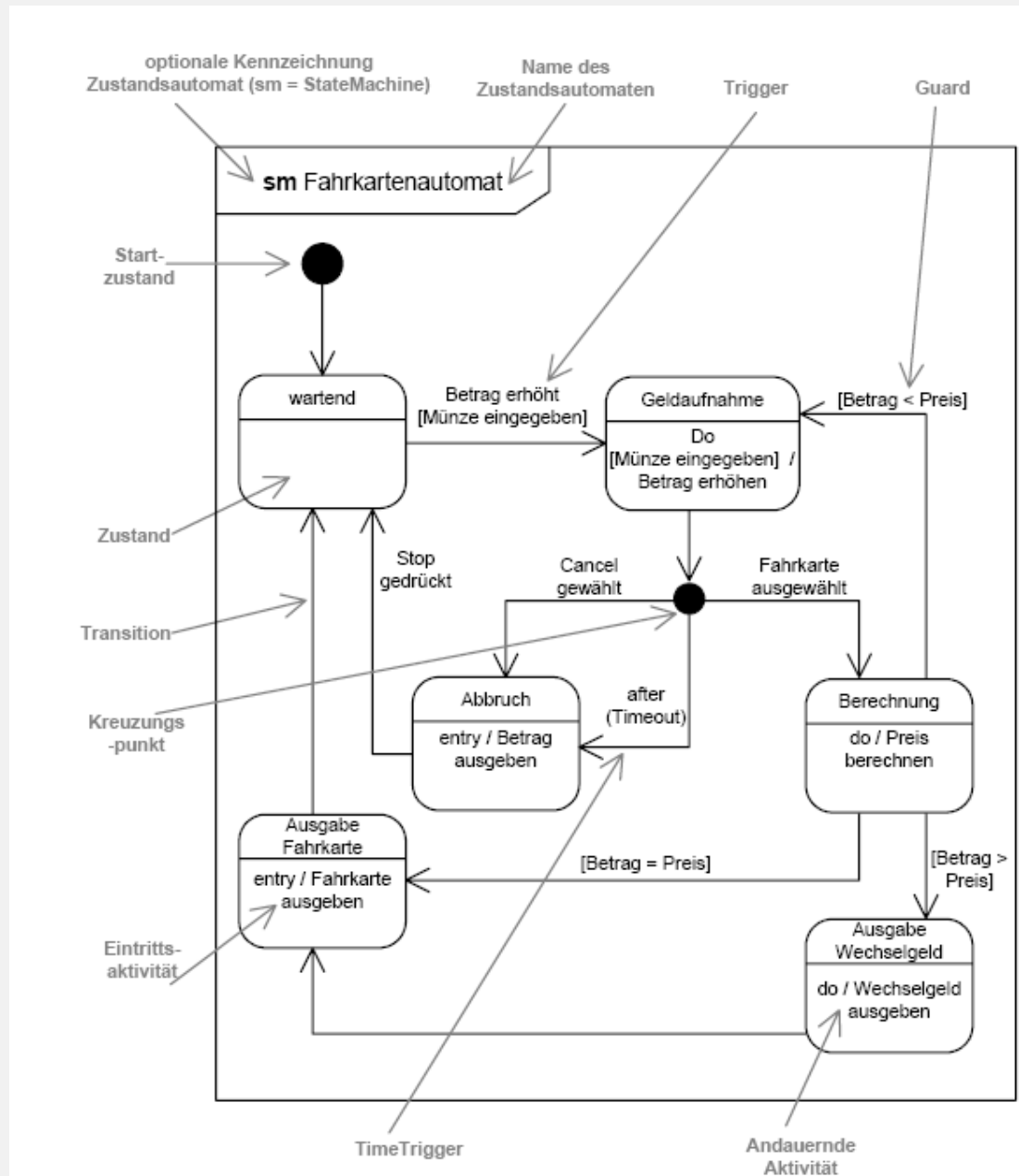
Vollständig neu:

- Kompositionsstrukturdiagramm
- Interaktionsübersichtsdiagramm
- Timing-Diagramm
- Kommunikationsdiagramm



UML 1.x	UML 2
Im Sequenzdiagramm interagieren Objekte, die durch Objektlinien repräsentiert werden.	Im Sequenzdiagramm interagieren jegliche „Partner“, die zur Kommunikation fähig sind. Sie sind durch Lebenslinien repräsentiert.
Referenzen auf andere Interaktionen sind nicht möglich.	Innerhalb von Sequenzdiagrammen kann auf andere Interaktionen verwiesen werden. Die Zerlegung von Lebenslinien ist erlaubt. Damit sind hierarchische Diagramme auf unterschiedlichem Abstraktionsgrad möglich.
Es gibt nur wenige Kontrollflussmöglichkeiten in Sequenzdiagrammen (Schleifen und Nachrichtenbedingungen sind notierbar).	Nahezu alle Kontrollflussmöglichkeiten (if, switch, exceptions, ...) höherer Programmiersprachen sind durch kombinierte Fragmente notierbar. Die Unterstützung nebenläufiger Modellierung wurde verbessert.
Es gibt 3 Nachrichtenarten: asynchron, synchron, unspezifiziert	Die unspezifizierte Nachricht entfällt, lost- und found-Nachrichten werden eingeführt.
	Neu eingeführte Elemente: <ul style="list-style-type: none"> ▪ Interaktionsrahmen ▪ Kombinierte Fragmente ▪ Sprungmarken und Coregionen ▪ Interaktionsreferenzen ▪ Gates für Nachrichten

UML 2 im Überblick



Nur marginale Änderungen

- Klassendiagramm
- Use-Case-Diagramm
- Objektdiagramm

Klein(-e Änderungen) aber oho:

- Paketdiagramm
- Verteilungsdiagramm

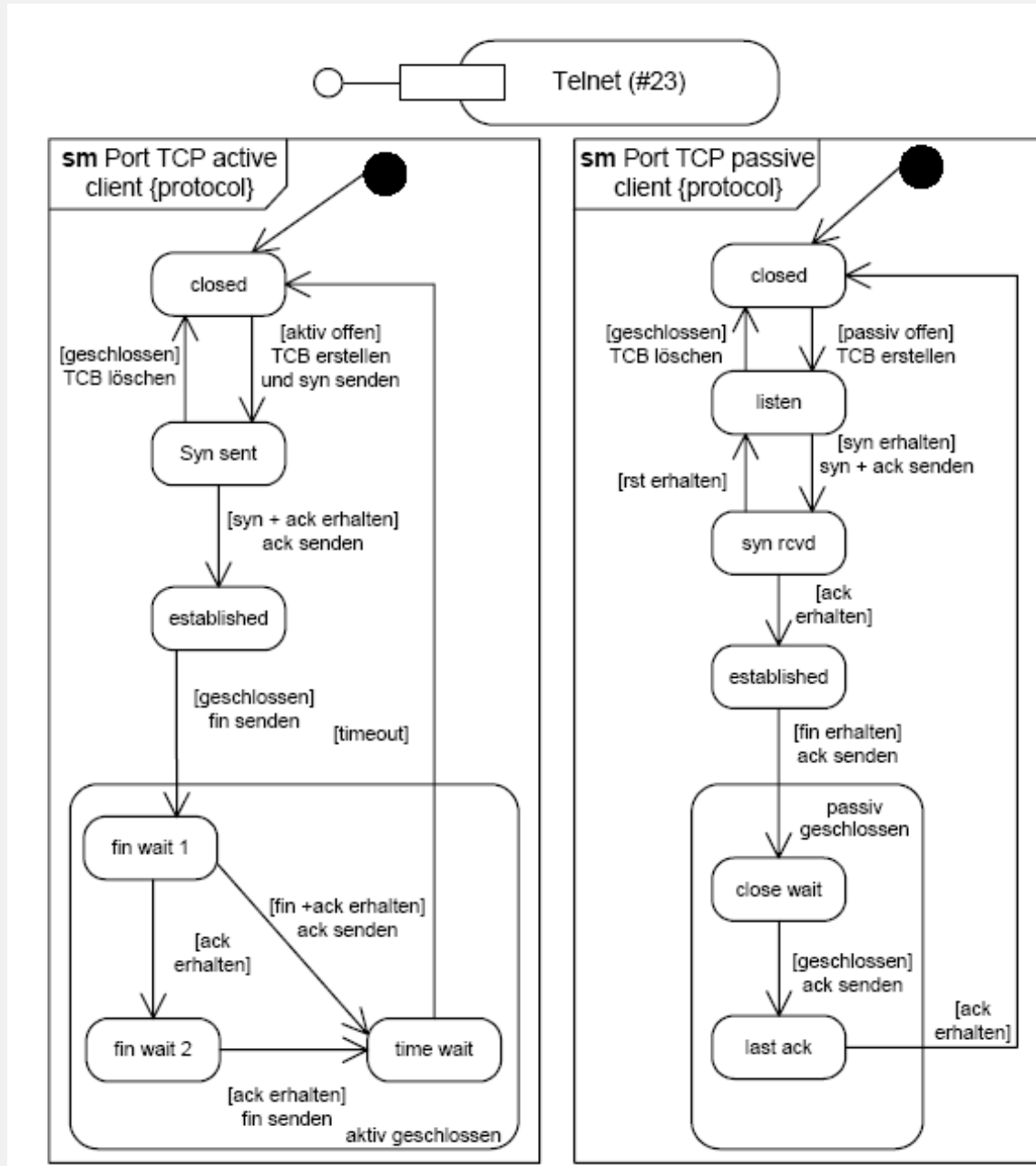
Massiv und tiefgreifend verändert:

- Aktivitätsdiagramm
- Sequenzdiagramm
- Zustandsautomat

Vollständig neu:

- Kompositionsstrukturdiagramm
- Interaktionsübersichtsdiagramm
- Timing-Diagramm
- Kommunikationsdiagramm

UML 2 im Überblick



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 im Überblick

Nur marginale Änderungen

- Klassendiagramm
- Use-Case-Diagramm
- Objektdiagramm

Klein(-e Änderungen) aber oho:

- Paketdiagramm
- Verteilungsdiagramm

Massiv und tiefgreifend verändert:

- Aktivitätsdiagramm
- Sequenzdiagramm
- Zustandsautomat

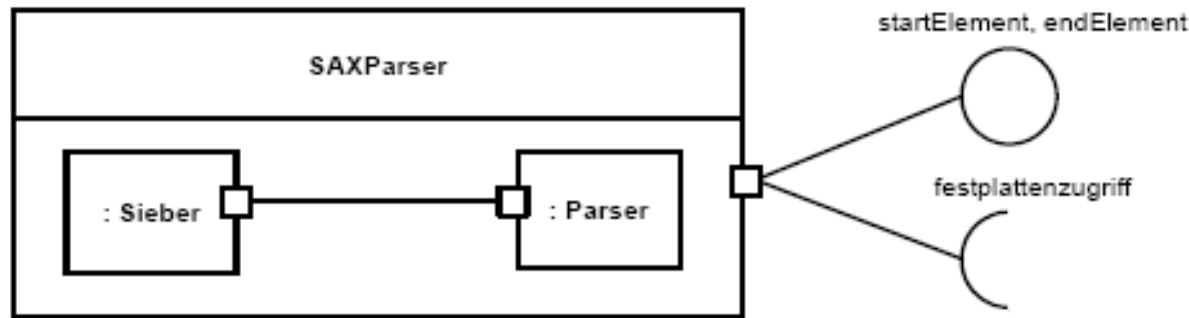
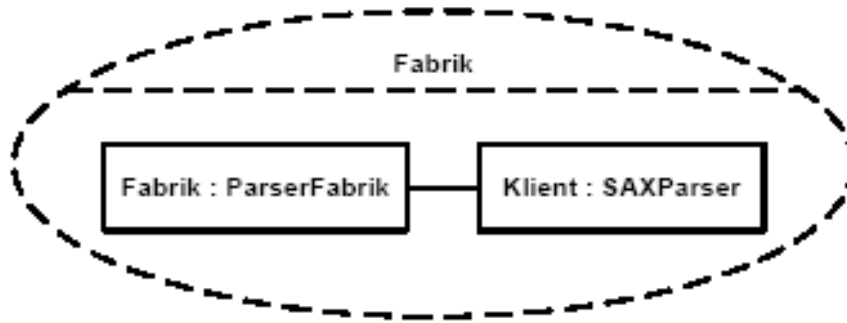
Vollständig neu:

- Kompositionsstrukturdiagramm
- Interaktionsübersichtsdiagramm
- Timing-Diagramm
- Kommunikationsdiagramm



UML 1.x	UML 2
	Interfaces können nun Protokollzustandsautomaten besitzen.
	Ein- und Austrittspunkte und Terminatoren wurden eingeführt.
	Regeln zur Ergänzung und Ersetzung von Transitionen bei vererbten Zustandsautomaten wurden hinzugefügt.
Tiefe Historien können nur eingehende Transitionen von außerhalb des enthaltenden Zustands haben.	Tiefe Historien können auch Ziel einer Transition innerhalb des enthaltenen Zustands sein (also nicht nur von außen).
Generalisierung von Zustandsautomaten durch Notizzettel	Das vererbte Verhalten wird mit einem Zustandsautomaten dargestellt.

UML 2 im Überblick



Nur marginale Änderungen

- Klassendiagramm
- Use-Case-Diagramm
- Objektdiagramm

Klein(-e Änderungen) aber oho:

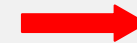
- Paketdiagramm
- Verteilungsdiagramm

Massiv und tiefgreifend verändert:

- Aktivitätsdiagramm
- Sequenzdiagramm
- Zustandsautomat

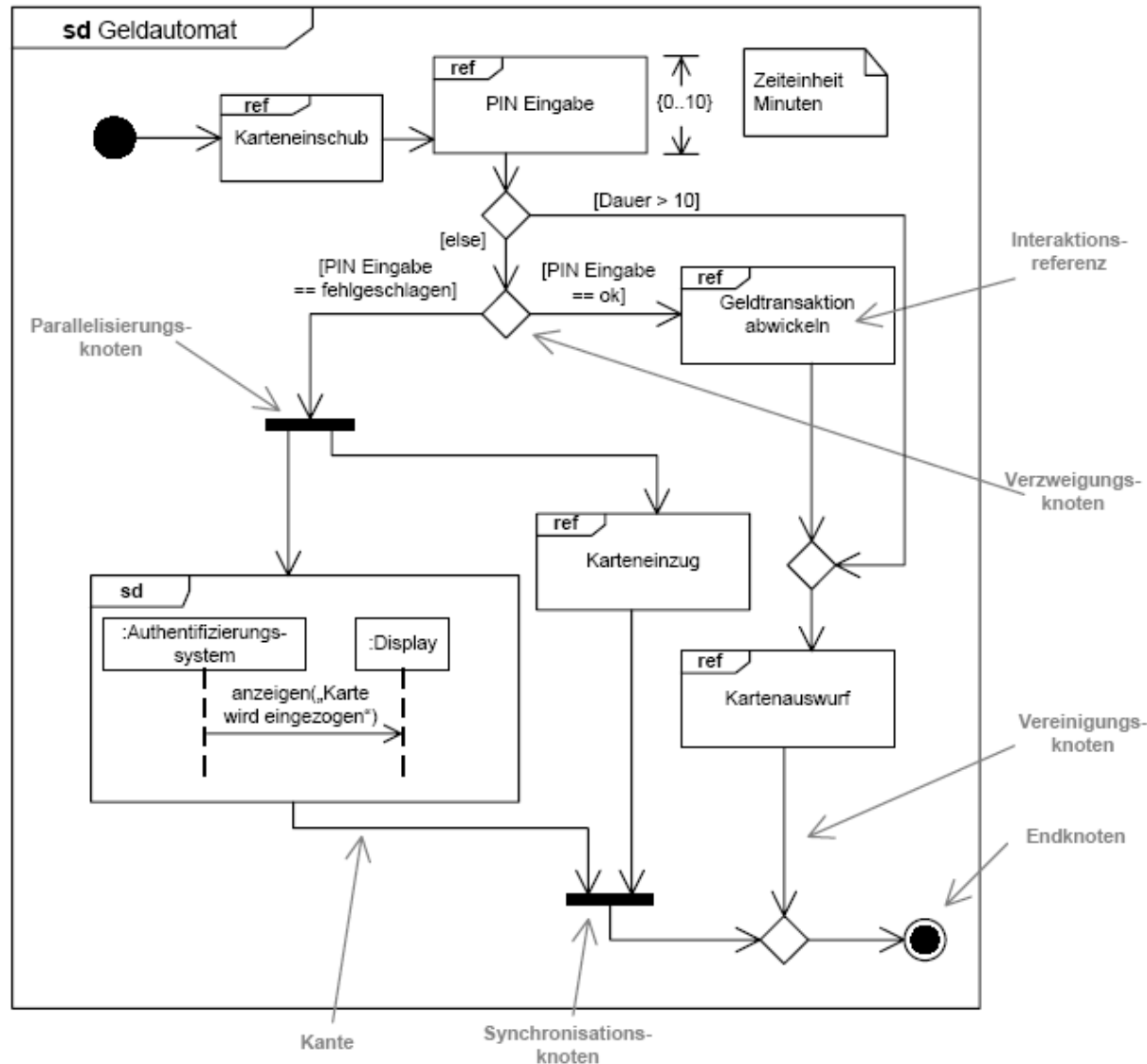
Vollständig neu:

- Kompositionsstrukturdiagramm
- Interaktionsübersichtsdiagramm
- Timing-Diagramm
- Kommunikationsdiagramm



UML 1.x	UML 2
	Kompositionsstrukturdiagramm neu eingeführt
	Konnektor neu eingeführt
	Kollaborationstyp und Kollaboration sind eigenständige Modellelemente

UML 2 im Überblick



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 im Überblick

Nur marginale Änderungen

- Klassendiagramm
- Use-Case-Diagramm
- Objektdiagramm

Klein(-e Änderungen) aber oho:

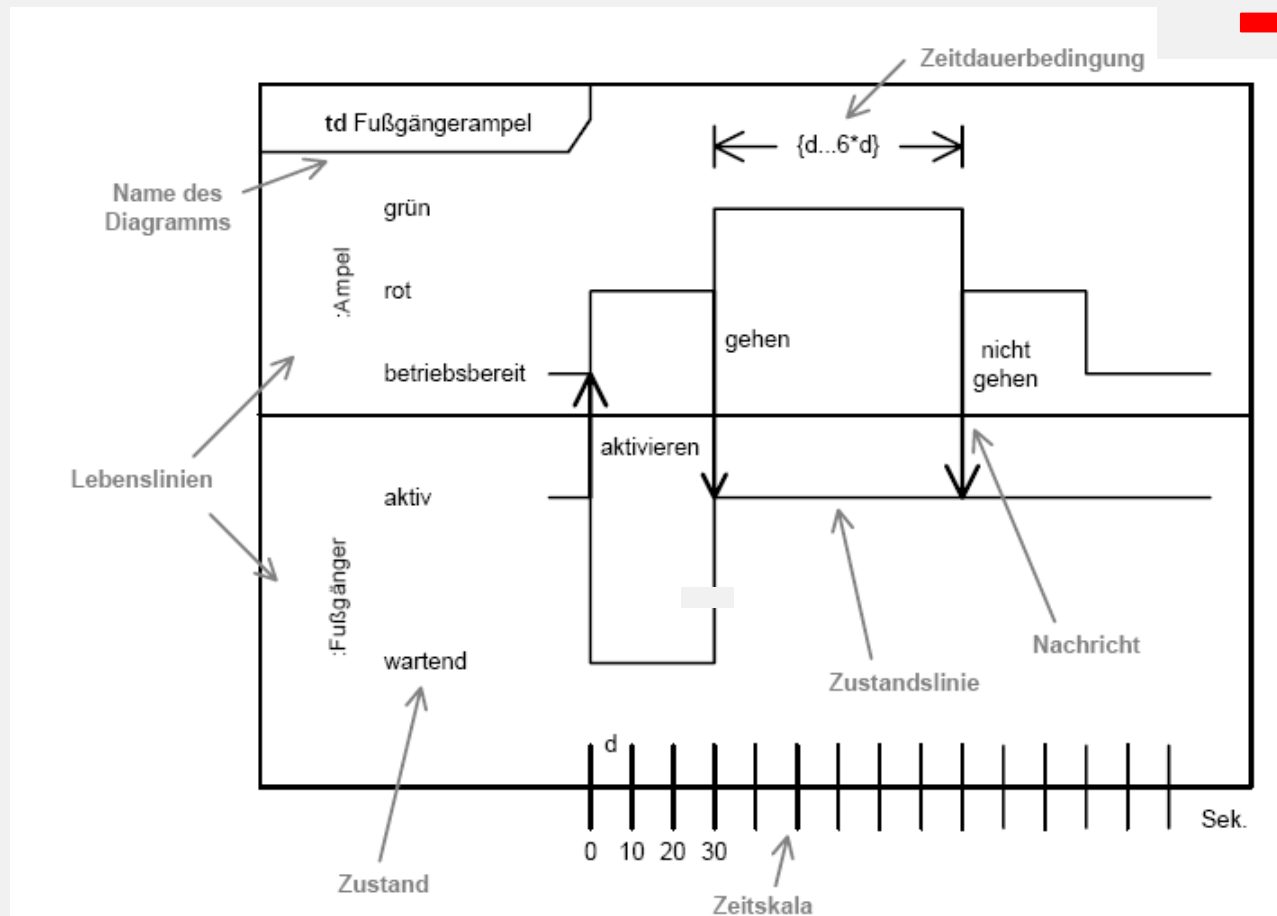
- Paketdiagramm
- Verteilungsdiagramm

Massiv und tiefgreifend verändert:

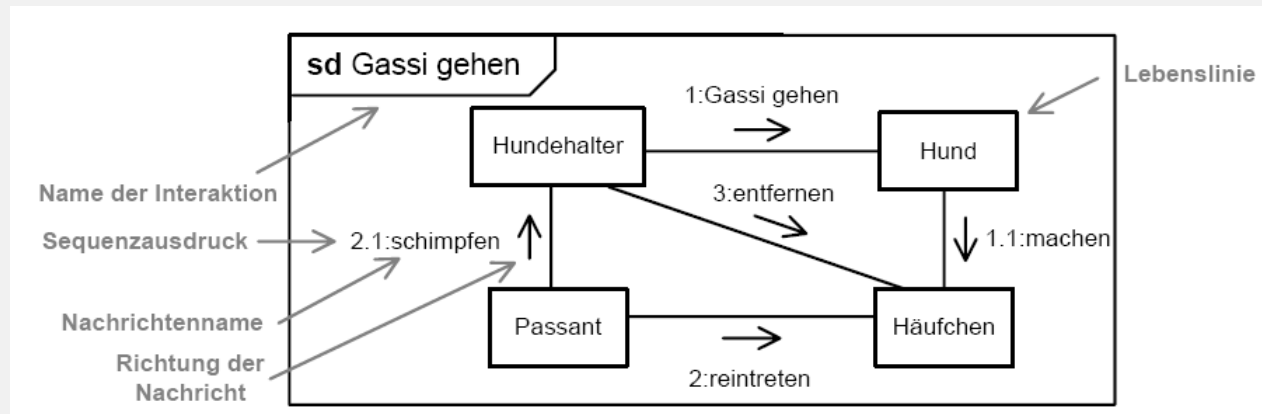
- Aktivitätsdiagramm
- Sequenzdiagramm
- Zustandsautomat

Vollständig neu:

- Kompositionsstrukturdiagramm
- Interaktionsübersichtsdiagramm
- Timing-Diagramm
- Kommunikationsdiagramm



UML 2 im Überblick



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

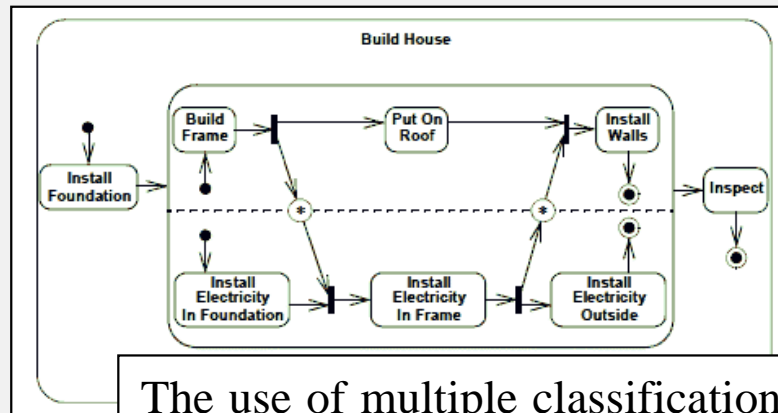


UML 1.x	UML 2
Kollaborationsdiagramm	Kommunikationsdiagramm
Vollständig äquivalent zum Sequenzdiagramm	Subset des Sequenzdiagramms, z.B. <ul style="list-style-type: none"> - keine Interaktionsverweise („ref“) - keine kombinierten Fragmente - Ereignisreihenfolge wird ignoriert

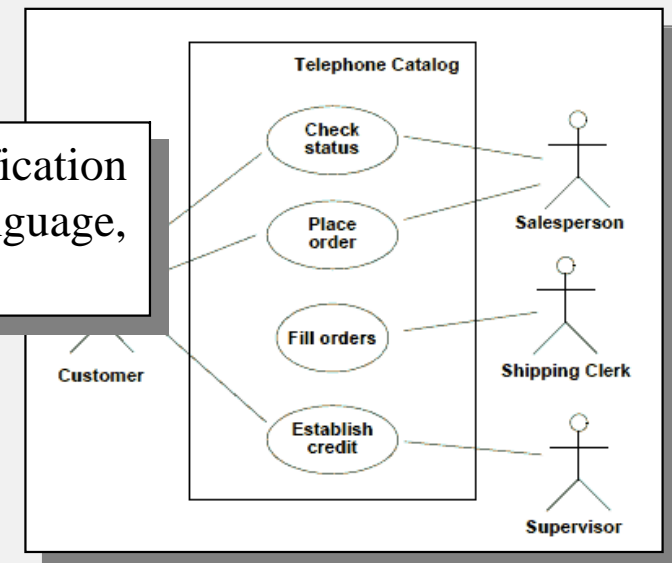
UML 2 im Überblick

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.



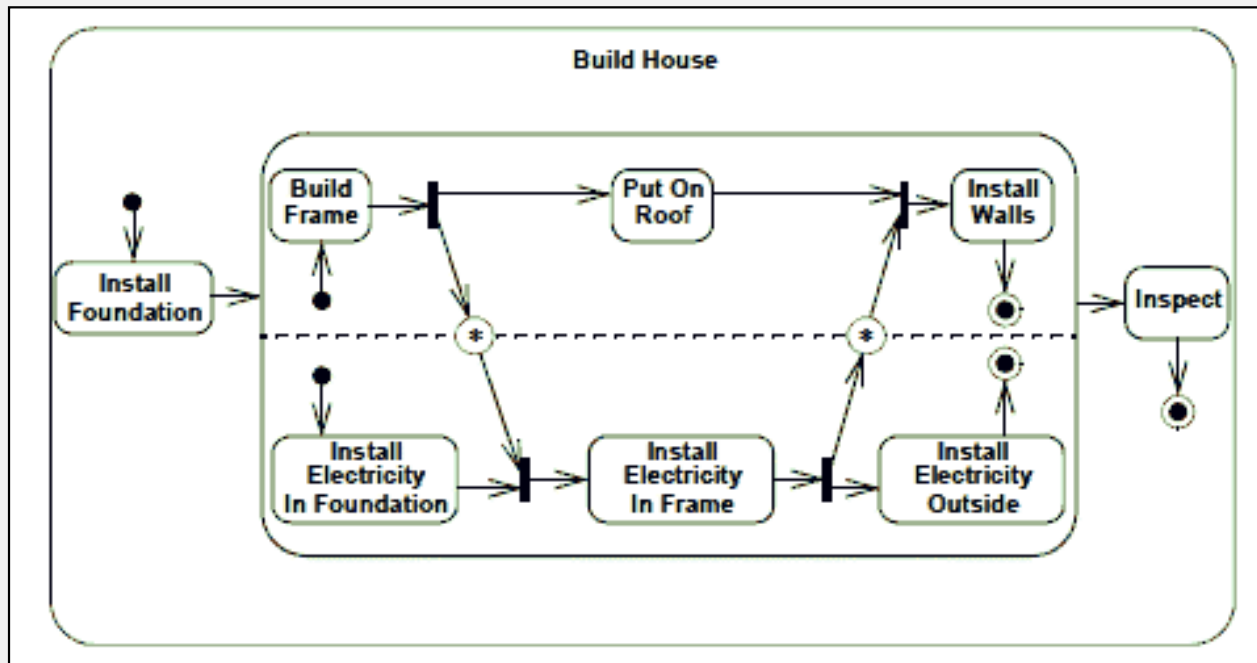
UML v2.0

- Eigenes Metamodell für OCL
- Durchgängige Verwendung der OCL im UML-Metamodell
- Alle Diagrammsprachen im Metamodell berücksichtigt

UML 2 im Überblick

UML v1.x

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



UML v2.0

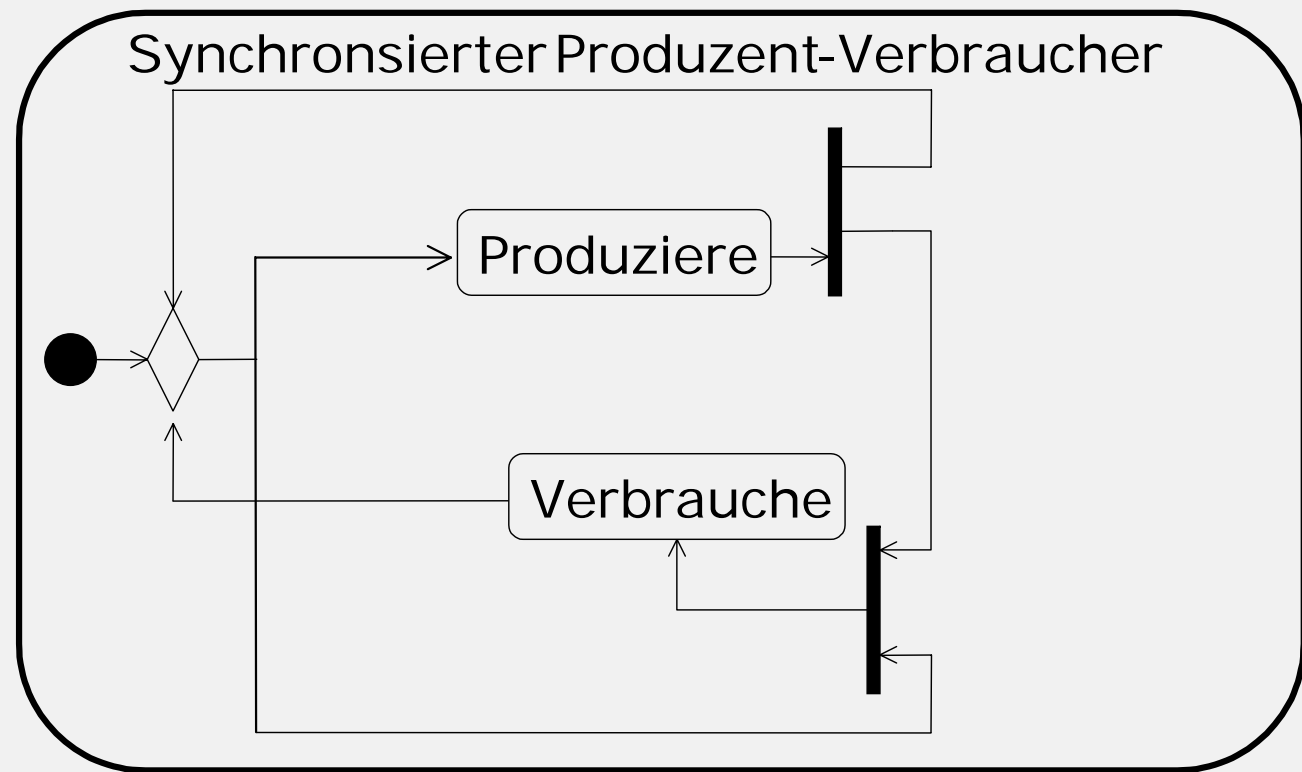
- Aktivitätsdiagramm eigenständiger Diagrammtyp
- Basiert auf Petri-Netzen (Tokenkonzept)
- Explizierung des Objektflusses zwischen Aktivitäten

UML 2 im Überblick

UML v2.0

- Aktivitätsdiagramm eigenständiger Diagrammtyp
- Basiert auf Petri-Netzen (Tokenkonzept)
- Explizierung des Objektflusses zwischen Aktivitäten

UML v1.x

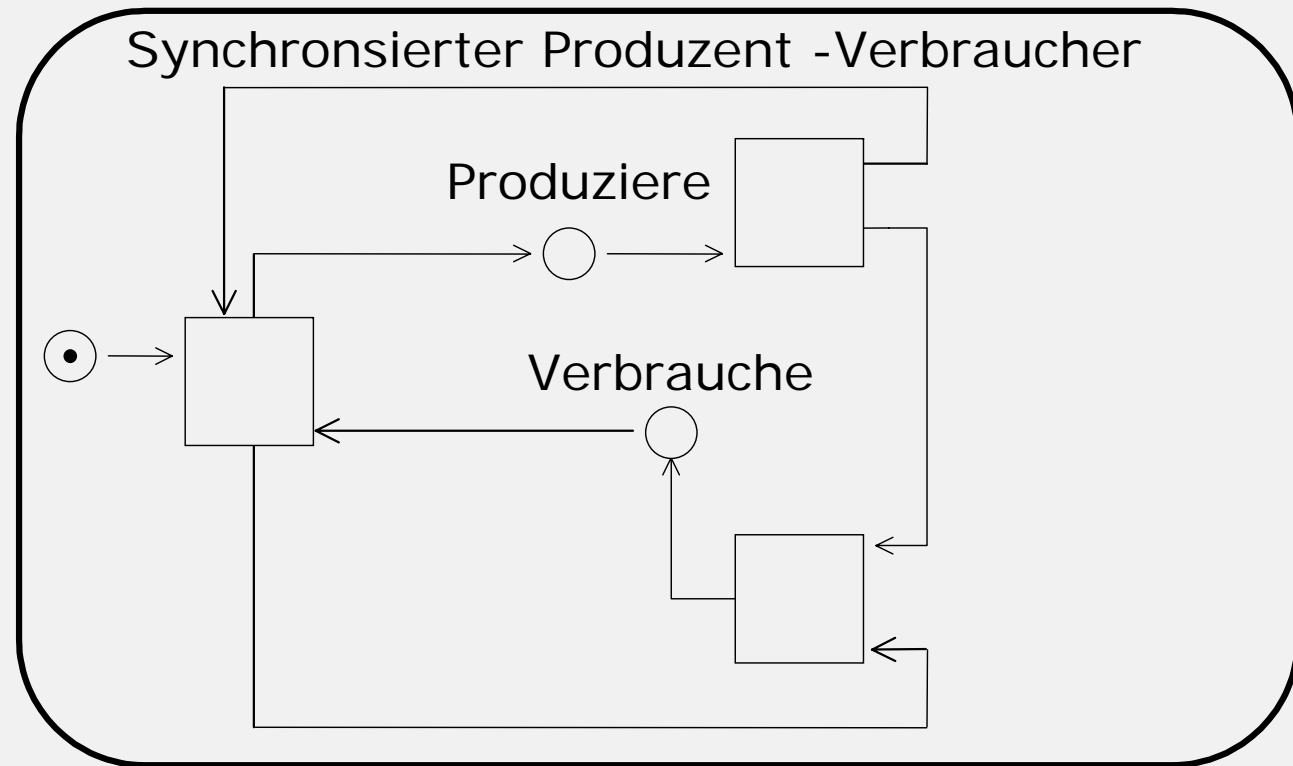


UML 2 im Überblick

UML v2.0

- Aktivitätsdiagramm eigenständiger Diagrammtyp
- Basiert auf Petri-Netzen (Tokenkonzept)
- Explizierung des Objektflusses zwischen Aktivitäten

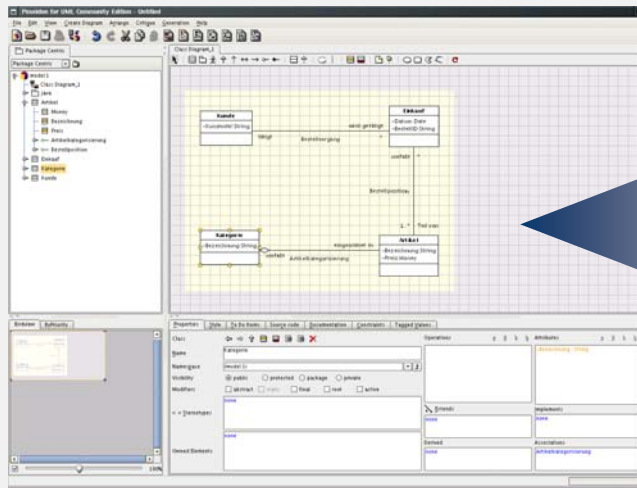
UML v2.0



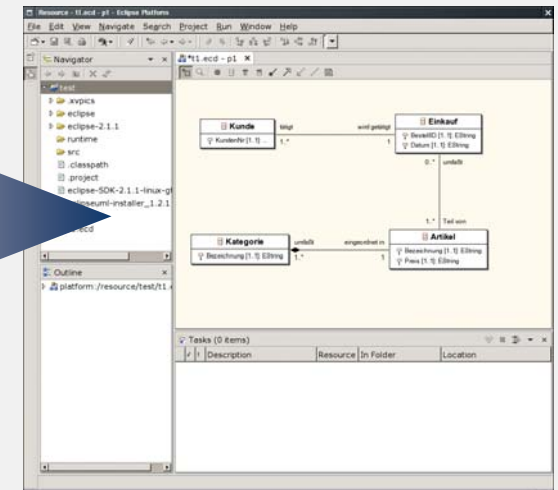
UML 2 im Überblick

UML v1.x

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



XMI
+
Diagram
Interchange



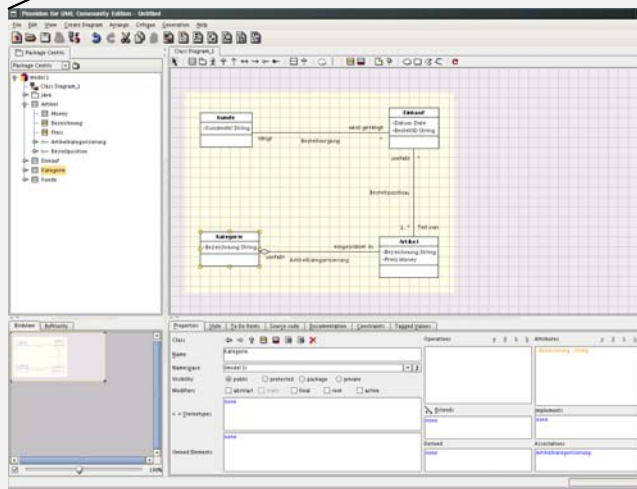
UML v2.0

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

UML 2 im Überblick

UML v2.0

- Erweitert das bestehende Metamodell
- Enthält die Graphik*daten* (nicht die Pixeldaten)
- Kann automatisiert in andere Formate (GIF, PNG, SVG) übersetzt werden



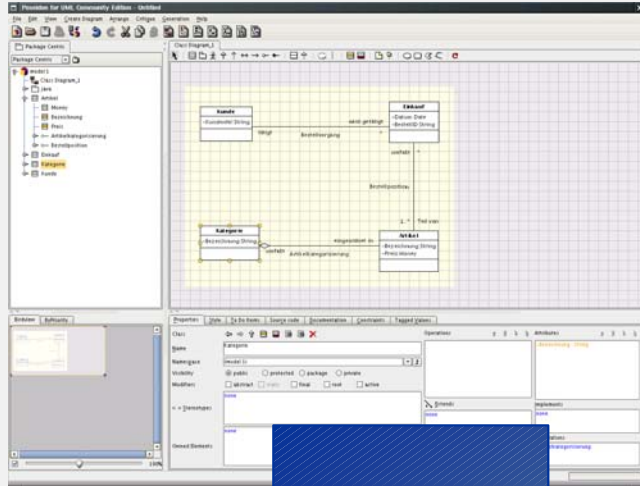
```
<XML xmi.version = '1.2' xmlns:UML = 'org.omg.xml.namespace.UML' timestamp = 'Wed Dec 31 12:49:37 CET 2003'>
<XML.header>
<XML.documentation>
<XML.exporter>Netbeans XMI Writer</XML.exporter>
<XML.exporterVersion>1.0</XML.exporterVersion>
</XML.documentation>
</XML.header>
<XML.content>
<UML:Model xmi.id = 'Ism:aa10fc:f9cbbf26cb:-7ffa' name = 'model 1' isSpecification = 'false'
isRoot = 'false' isLeaf = 'false' isAbstract = 'false'>
```

„klassische“
Modelldaten

```
...
<UML:Diagram xmi.id = 'Idi:aa10fc:f9cbbf26cb:-7fd2' isVisible = 'true' name = 'Class Diagram_1'
zoom = '1.0'>
<UML:GraphElement.position>
<XML.field>0.0</XML.field>
<XML.field>0.0</XML.field>
</UML:GraphElement.position>
<UML:GraphNode.size>
<XML.field>540.0</XML.field>
<XML.field>403.0</XML.field>
</UML:GraphNode.size>
<UML:Diagram.viewport>
<XML.field>0.0</XML.field>
<XML.field>0.0</XML.field>
</UML:Diagram.viewport>
<UML:GraphElement.semanticModel>
<UML:SimpleSemanticModelElement xmi.id = 'Ism:aa10fc:f9cbbf26cb:-7fd1' presentation = ''
typeInfo = 'ClassDiagram'/>
</UML:GraphElement.semanticModel>
```

Neu mit UML 2.0:
graphische Modelldaten

UML 2 im Überblick



XMI
+
DI



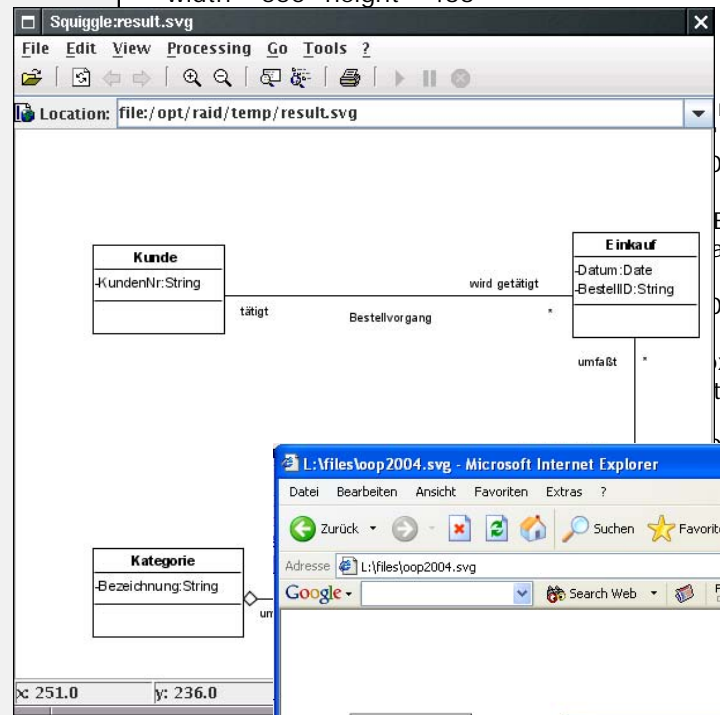
```
<xsl:stylesheet version="1.0"
xmlns="http://www.w3.org/2000/svg"
xmlns:UML="org.omg.xml.namespace.UML"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:output media-type="image/svg+xml" method="xml" indent="yes"/>
<xsl:include href="globalfunctions.xsl"/>
<xsl:include href="Templates_for_cd_dd.xsl"/>
<xsl:param name="diagramID" />
<xsl:param name="fillWith"/>
<xsl:if test="SRIIWith"><xsl:value-of select="SRIIWith"/></xsl:if>
<xsl:if test="not(SRIIWith)">white</xsl:if>
</xsl:if>
<xsl:variable name="DiagramID">
<xsl:if test="SdiagramID"><xsl:value-of select="SdiagramID"/></xsl:if>
<xsl:if test="not(SdiagramID)">
<xsl:value-of
select="/UML:Diagram[position()=1][child::UML:GraphElement/semanticModel/UML:SimpleSemanticModelElement[attribute::typeInfo='ClassDiagram']/@attribute::xml:Id"/>
</xsl:if>
...

```

XSLT

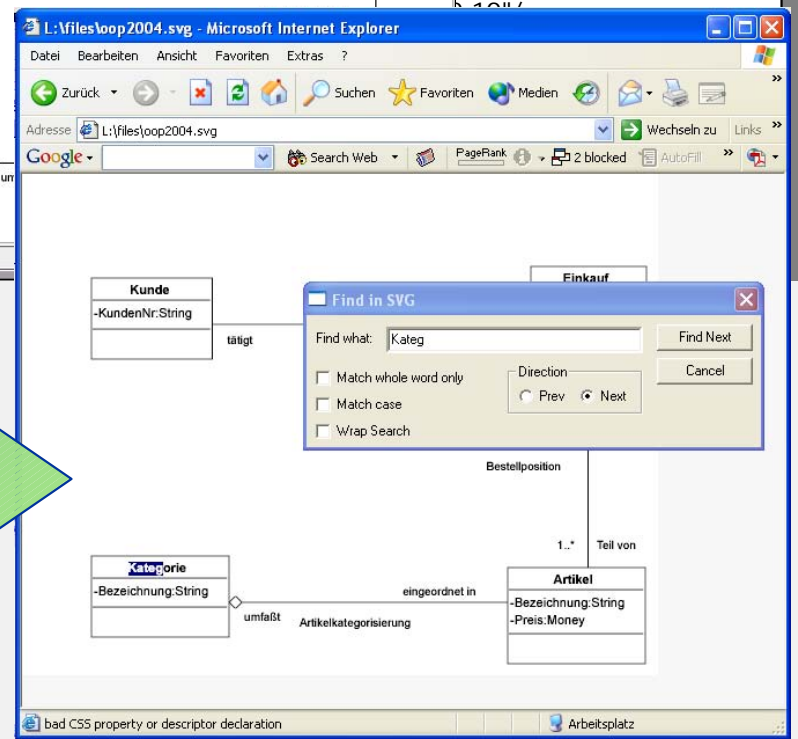
```
<?xml version="1.0" encoding="UTF-8"?>
<svg xmlns="http://www.w3.org/2000/svg"
width="550" height="433"
markerWidth="11"
>
  <!-- ... -->
  <rect x="0" y="0" width="10" height="10" />
  <!-- ... -->
  <rect x="0" y="0" width="10" height="10" refX="10"
refY="10" orient="auto" />
  <!-- ... -->
</svg>

```



```
markerWidth="11"
>
  <rect x="0" y="0" width="10" height="10" />
  <!-- ... -->
  <rect x="0" y="0" width="10" height="10" refX="10"
refY="10" orient="auto" />
  <!-- ... -->
</svg>

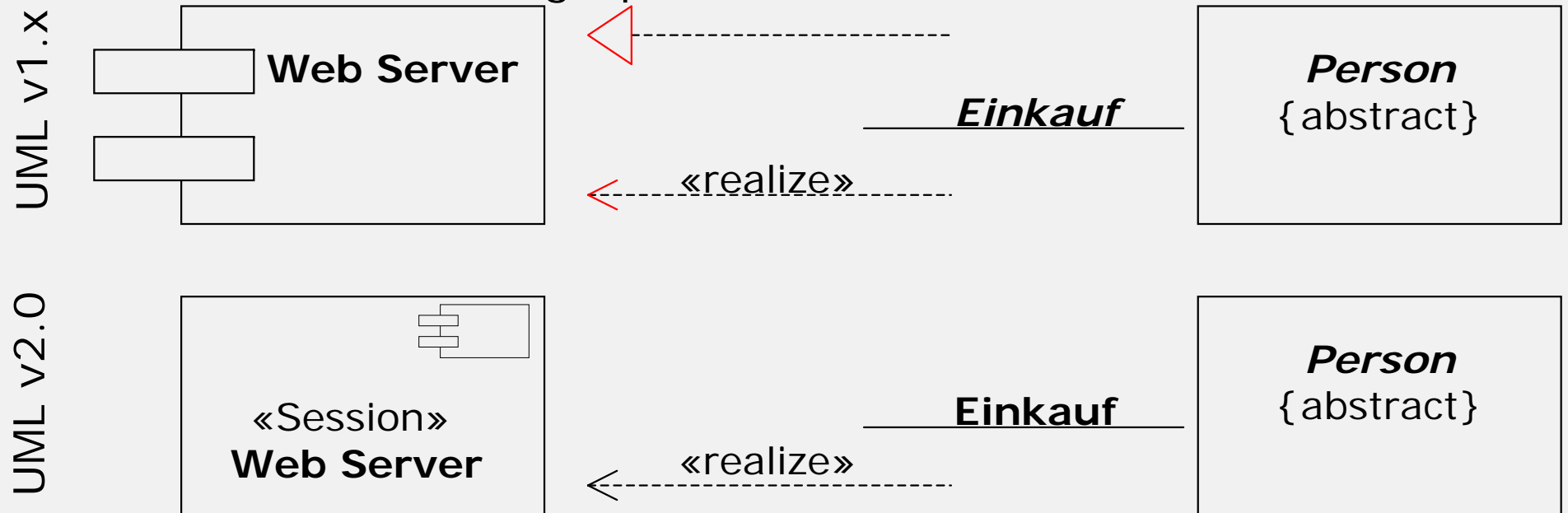
```



UML 2 im Überblick

UML v1.x

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



UML v2.0

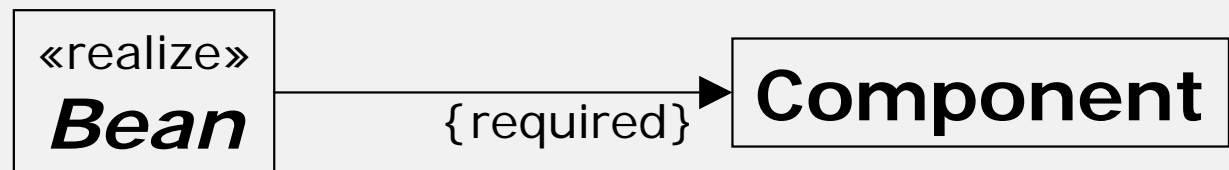
- Weniger symbolhafte Darstellungen, mehr textuelle Explizierung
- Wiederverwendung bestehender Basiskonzepte
- Stärkere Orthogonalität

UML 2 im Überblick

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 **ausschließlich textuell** (im Metamodell) oder **gar nicht** (für den Anwender) **möglich**.

UML v2.0



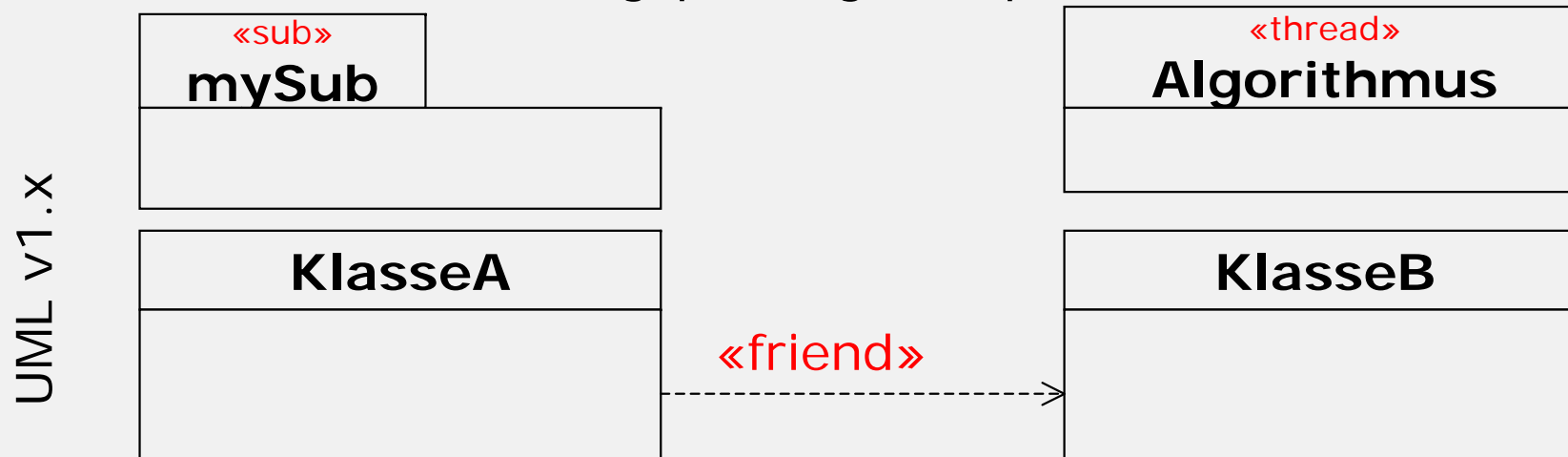
UML v2.0

- Graphische Definition von Stereotypen
- Semantische Beziehung zu stereotypisiertem (d.h. spezialisiertem) Modellelement

UML 2 im Überblick

UML v1.x

- Kein echtes logisches Metamodell
- Vorhandene Sprach- und Spracherweiterungsmechanismen sind oft programmiersprachen- oder umsetzungsparadigmenspezifisch



UML v2.0

- Sprachspezifische oder implementierungnahe Konstrukte in Profile verschoben oder vollständig entfernt

UML 2 im Überblick

UML v1.x

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

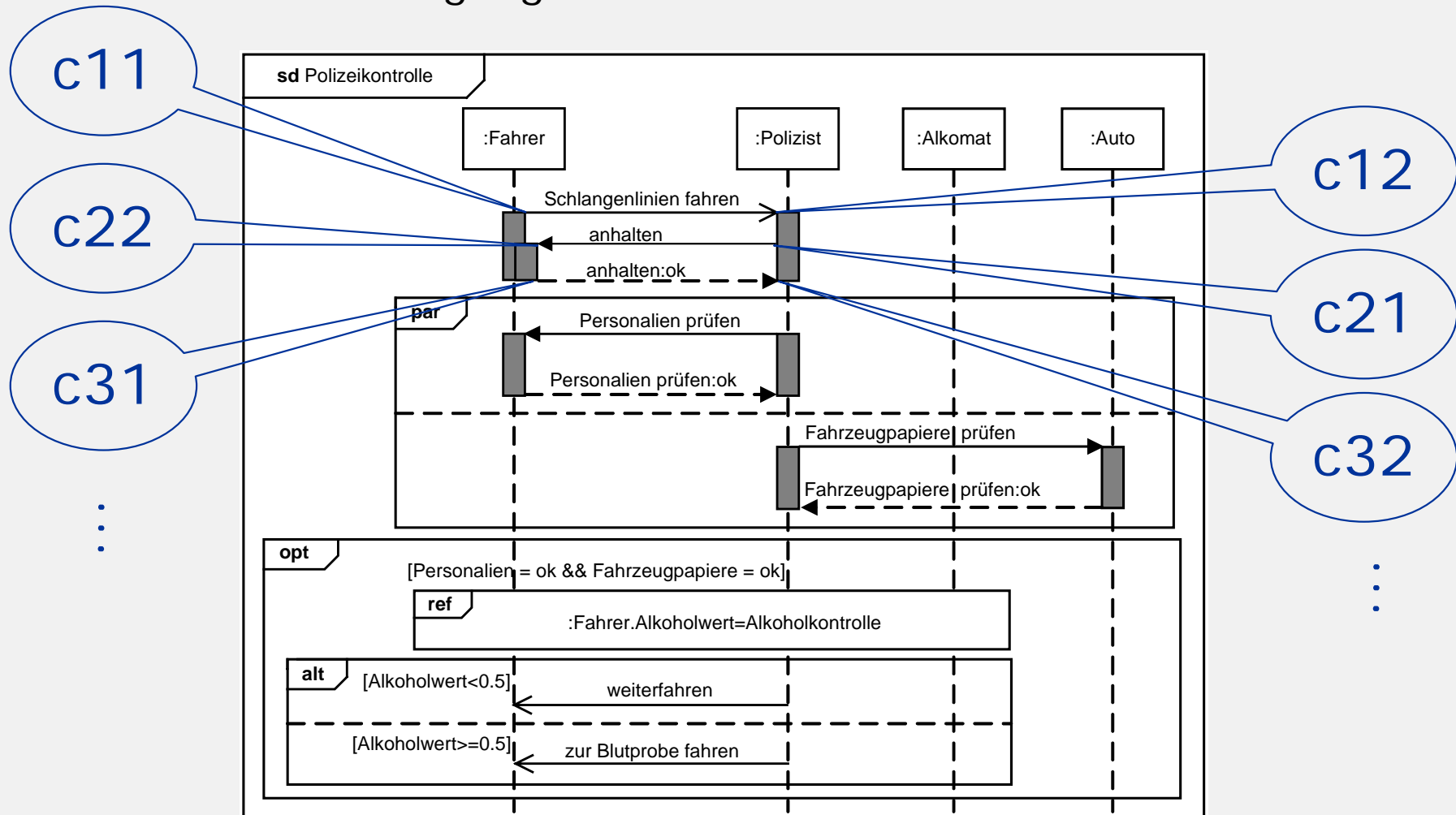
UML v2.0

- 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

UML 2 im Überblick

UML v2.0

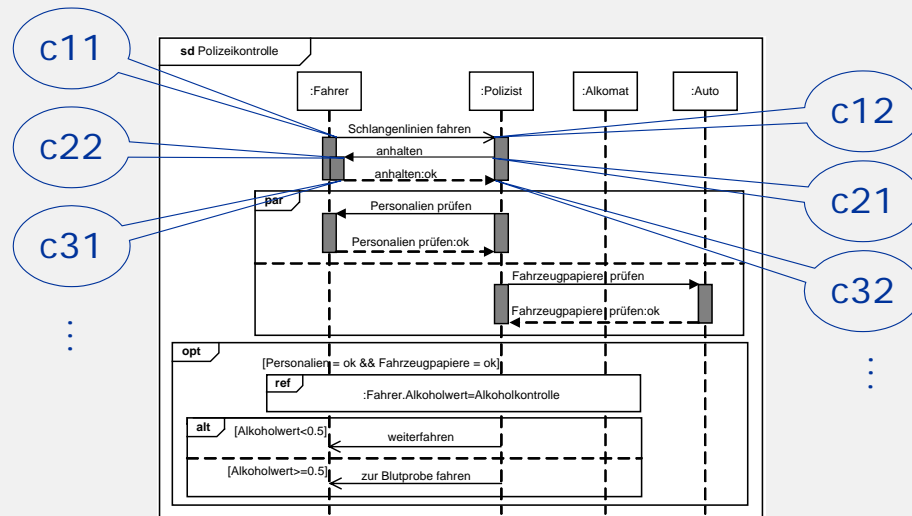
- Als Alternative zu der graphischen Darstellung von Sequenzdiagrammen stehen tabellenartige Texte zur Verfügung



UML 2 im Überblick

UML v2.0

- Als Alternative zu der graphischen Darstellung von Sequenzdiagrammen stehen tabellenartige Texte zur Verfügung



Optionale tabellarische Darstellung

- Maschinell verarbeitbar
- Möglichkeit der Konsistenzprüfung
- Übersichtlichkeit bei großen Diagrammmengen

Lifeline Instance	Sending msg. Instance	Order	Msg. Name	Msg. Receive Instance
Fahrer		c11	Schlangenlinien fahren	Polizist
Polizist	Fahrer	c12	Schlangenlinien fahren	Polizist
Polizist		c21	anhalten	Fahrer
Fahrer	Polizist	c22	anhalten	Fahrer
Fahrer		c31	anhalten:ok	Polizist
	Fahrer	c32	anhalten:ok	Polizist

UML 2 im Überblick

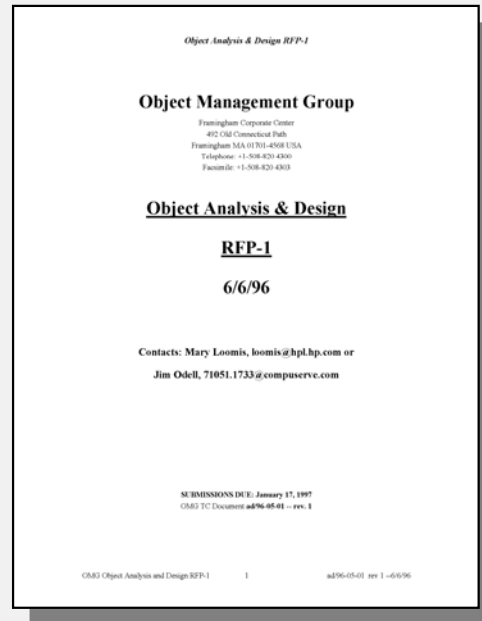
Metamodell

UML v2.0

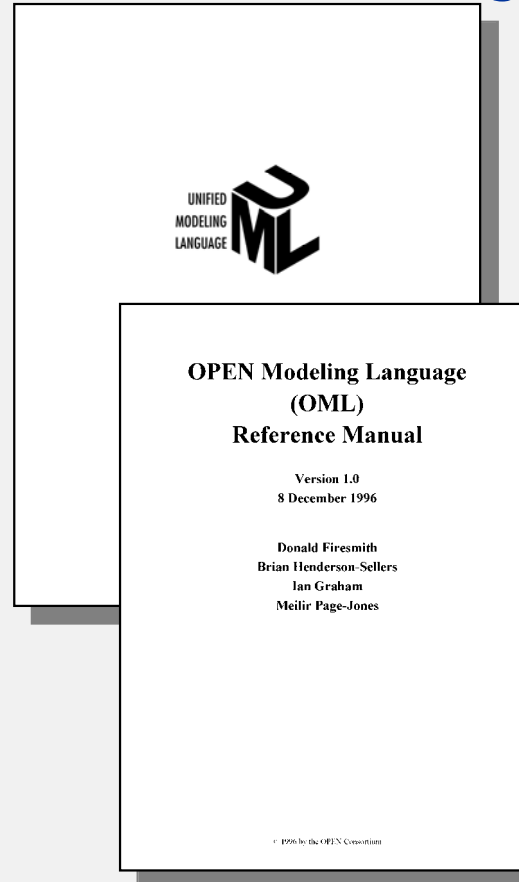
- Bereits erwähnt:
 - Diagrammaustausch
 - Profile
 - Object Constraint Language... im Metamodell berücksichtigt.
- Zusätzlich:
 - Basiskonzepte grundlegend überarbeitet
 - Verallgemeinert
 - Orthogonalität
 - Wiederverwendet (im Metamodell)⇒ fast vollständig neues, kompaktes Metamodell
 - Jedoch: Einzelne Primitive (Note) bleiben immernoch völlig unberücksichtigt

UML 2 im Überblick

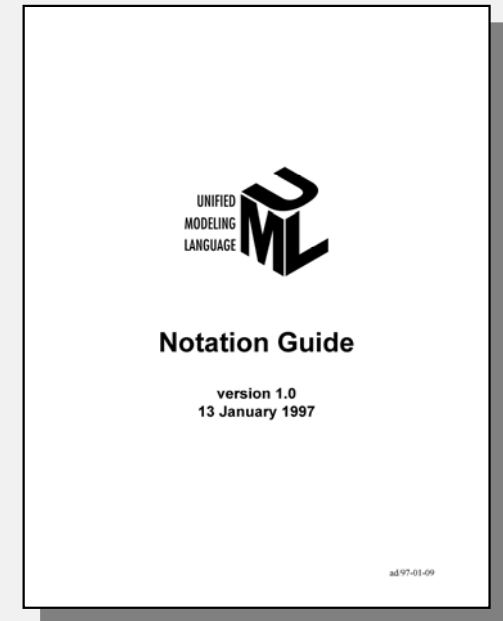
Hintergründe der Entstehung – UML 1



Request for Proposals
1996



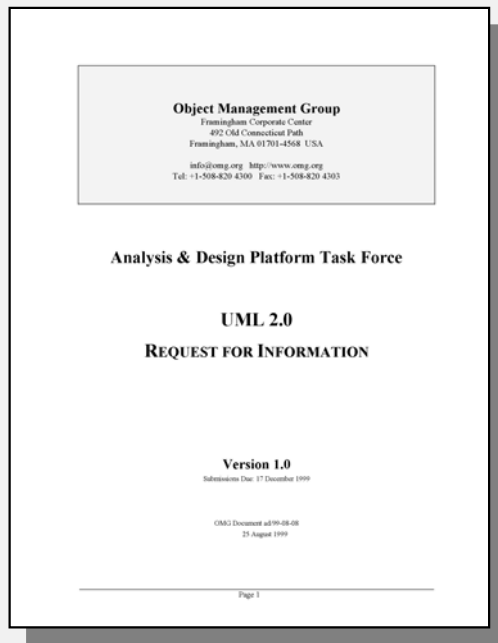
Zwei konkurrierende
Vorschläge



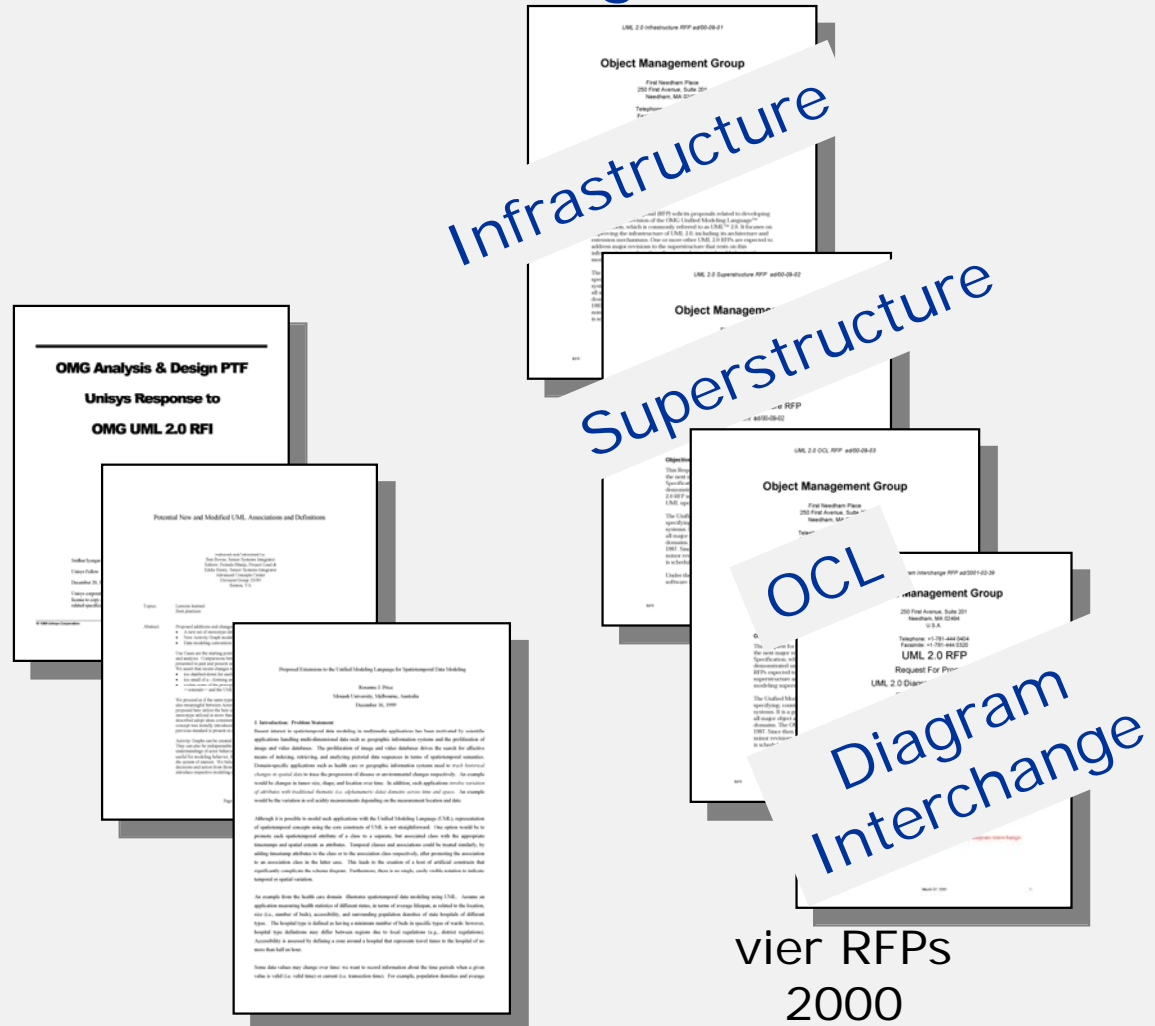
Finale Spezifikation
1997

UML 2 im Überblick

Hintergründe der Entstehung



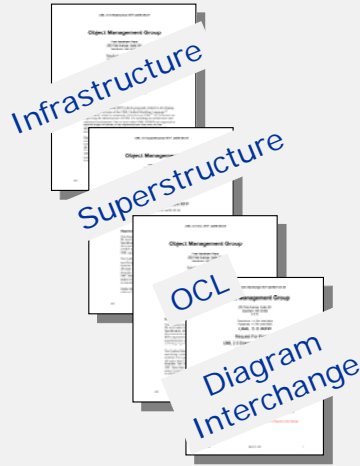
Request for Information
1999



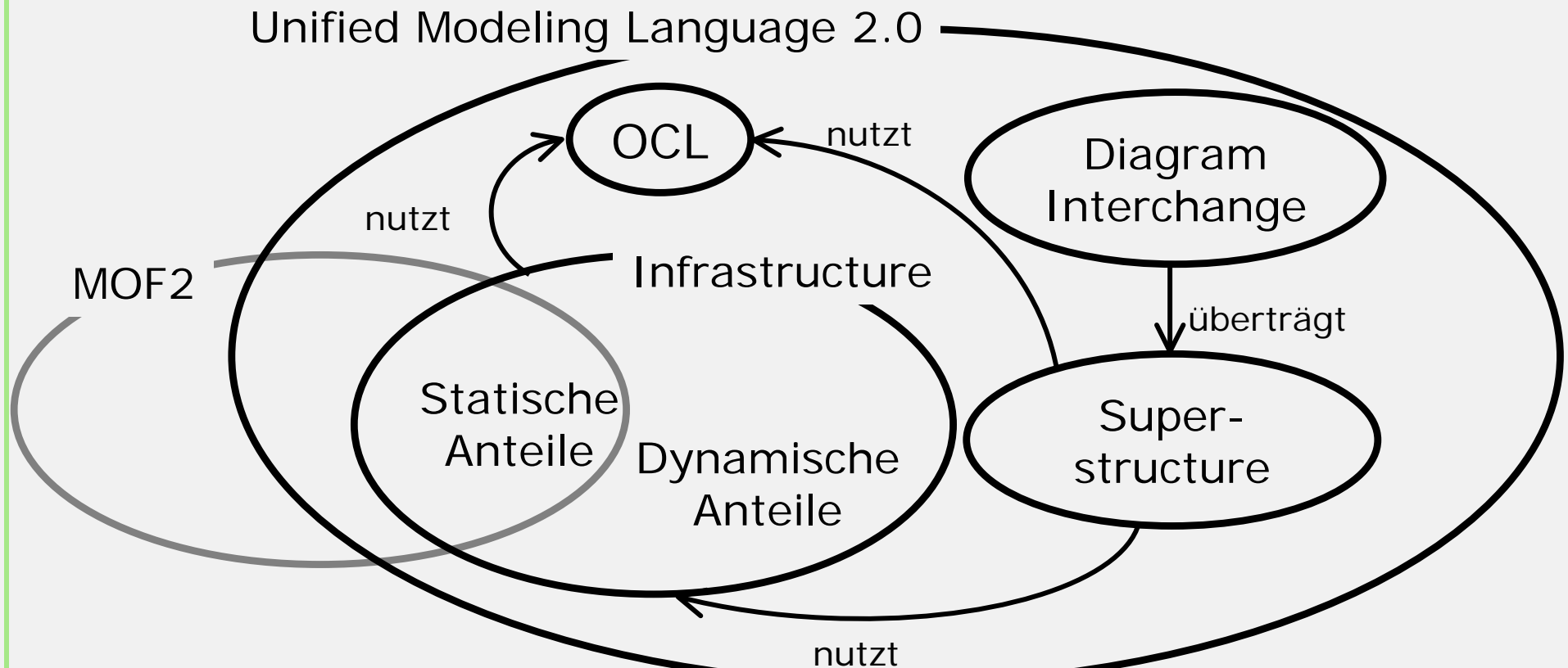
20 Einreichungen

UML 2 im Überblick

Hintergründe der Entstehung

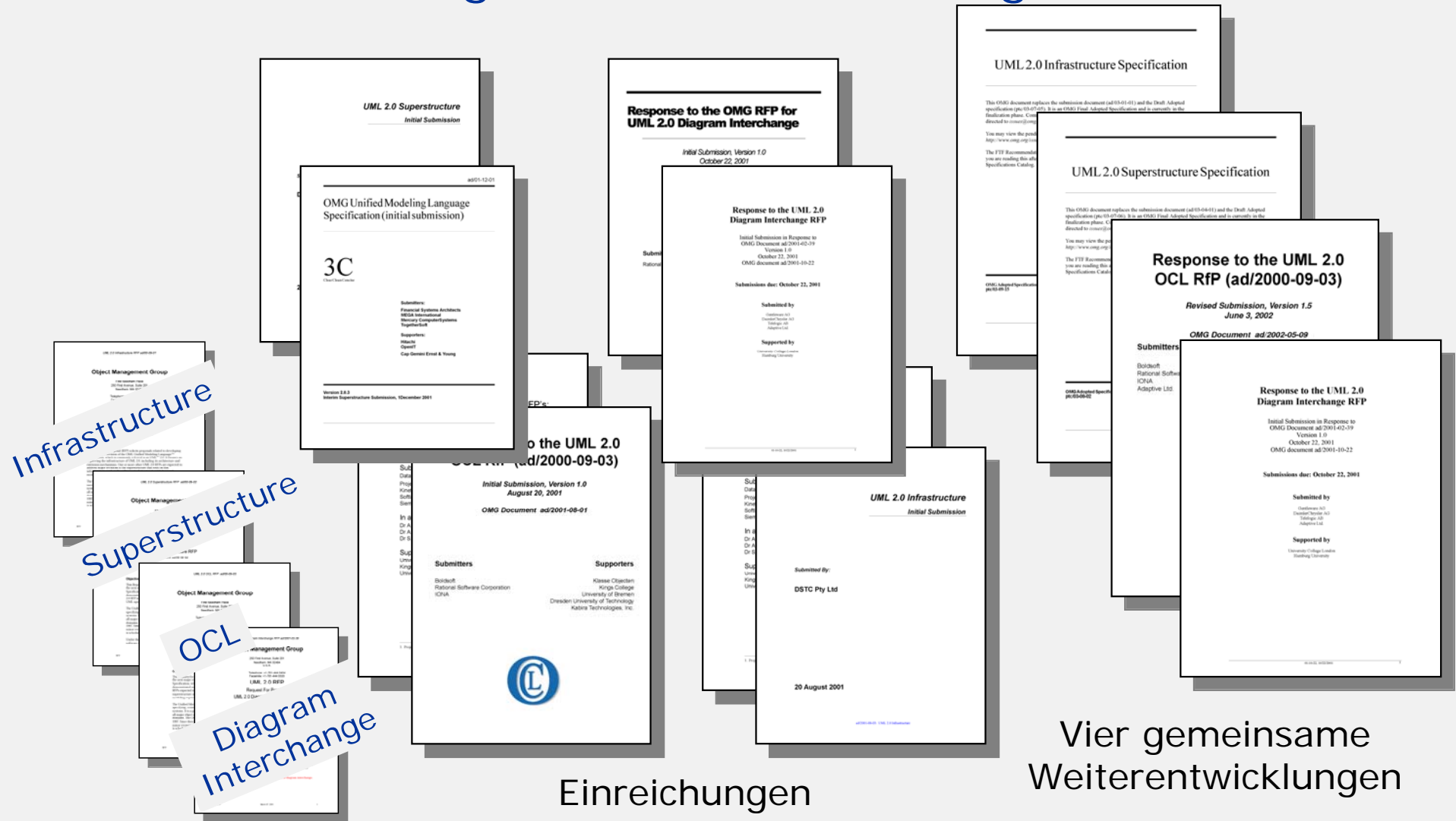


- Kein monolithischer Sprachentwurf (mehr)
- Vier separate Entwicklungsgruppen erzeugen vier separate Weiterentwicklungen mit starkem inneren Zusammenhang



UML 2 im Überblick

Hintergründe der Entstehung



Infrastructure

Superstructure

OCL

Diagram Interchange

Vier gemeinsame Weiterentwicklungen

Einreichungen durch 12 Gruppen

vier RFPs
2000

UML 2 im Überblick

Ein Ausblick auf UML 3.0

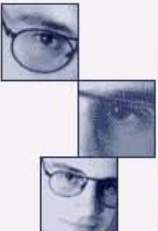
- Mit UML 2.0 ist die Reise noch nicht zu Ende
- Mögliche Weiterentwicklungen
 - Vollständig ausführbare Modelle?
 - Visuelle Modelle -> Visuelle Programme?
 - Neue Paradigmen?
 - 3-D Modelle?
 - ...
- Zeitabschätzung
 - $F_{UML3} = F_{UML2} + 3 * T_{RTF} + T_{RFP}$
 - $F_{UML3} = Q2\ 2001 + (3*1) + 2$
 ≈ 2006

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


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

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

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

jeckle.de

Diese Folien und weitere Informationen
rund um Standardisierung und Einsatz der UML

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)
- Mehr Informationen:
www.uml-glasklar.de

