DaimlerChrysler

Modellaustausch zwischen heterogenen UML-Werkzeugen

Mario Jeckle
DaimlerChrysler Forschungszentrum Ulm
mario.jeckle@daimlerchrysler.com
mario@jeckle.de
www.jeckle.de

Gliederung

I Modelle:

Verwendung, Darstellung und Austausch

II XML-basierter Modellaustausch:

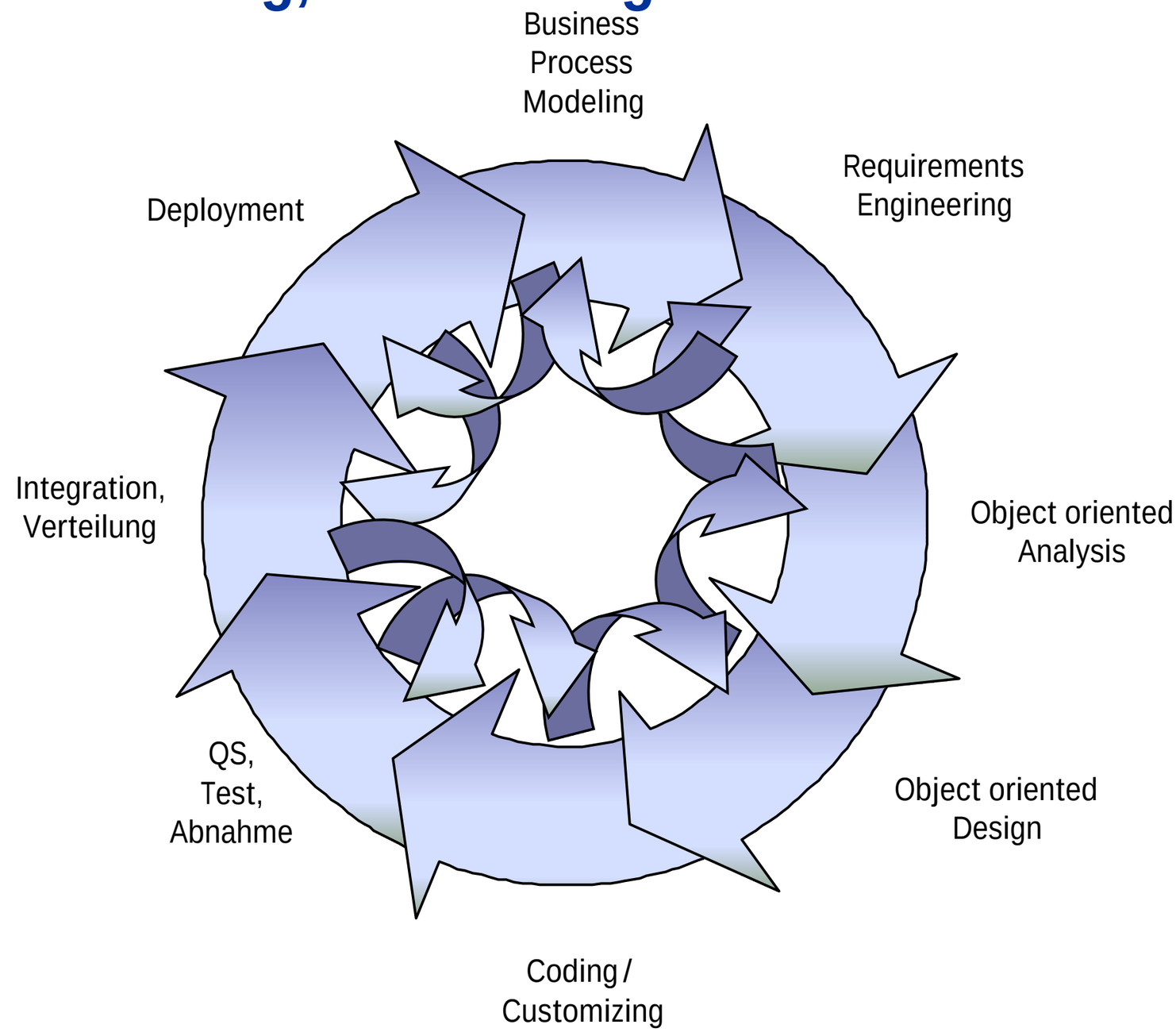
OMG's *Metadata Interchange Format*

- Ansatz
- Einbettung in die *Model Driven Architecture*

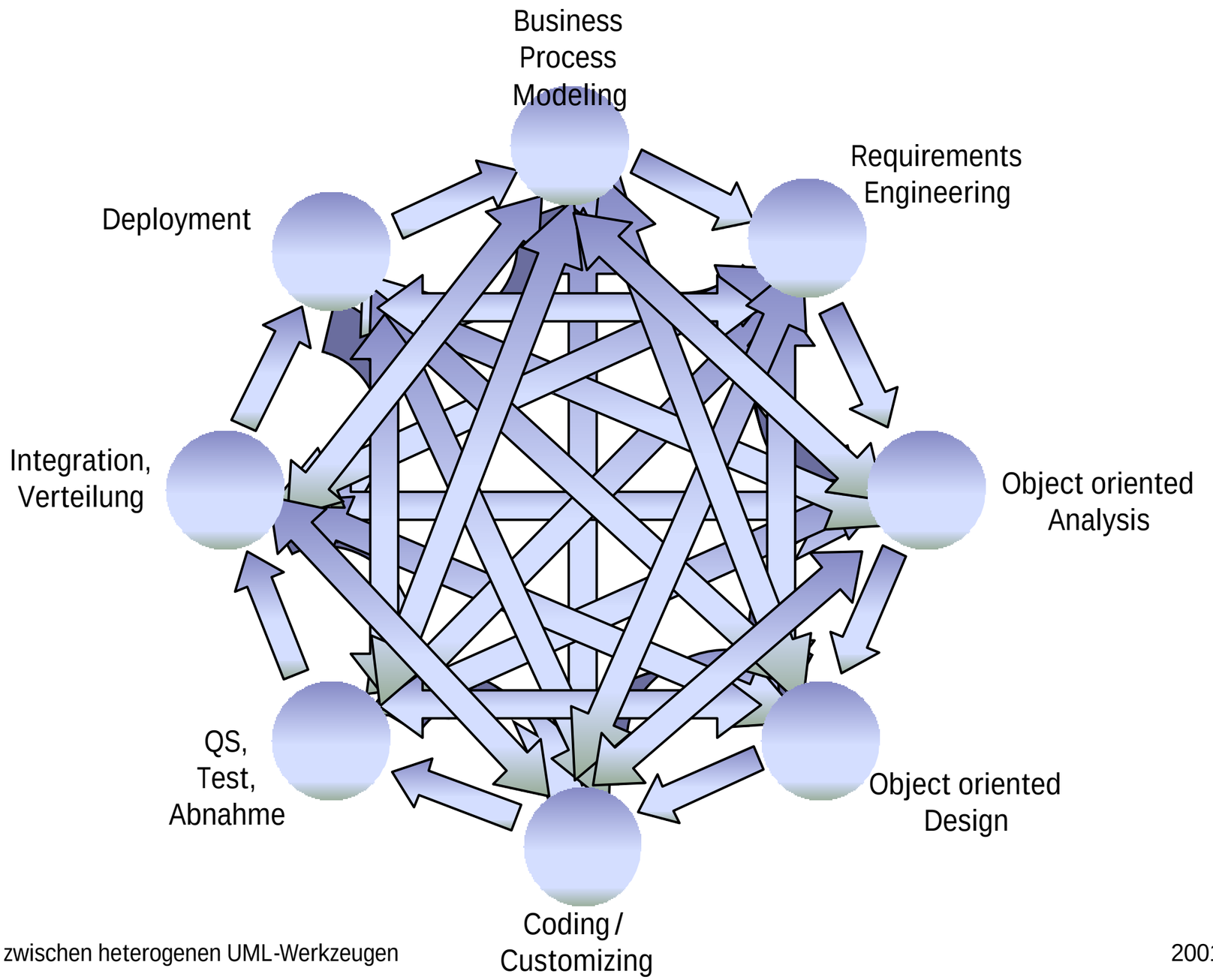
II Anwendungsfälle:

- Modellaustausch
- Designvalidierung
- Code- und Dokumentationsgenerierung

Modelle: Verwendung, Darstellung und Austausch



Modelle: Verwendung, Darstellung und Austausch



Modelle: Verwendung, Darstellung und Austausch

Deployment

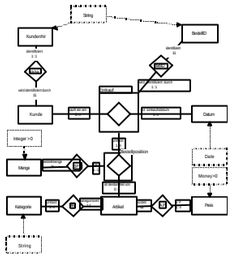
Einsatzfehlerprotokoll

Fehlerprotokoll

Datum	Beschreibung	reproduzierbar	gemeldet
2001-01-07	kein DB-Zugriff in Maske Stammdaten	Y	T. Müller
2001-01-09	Fehlerhafte Rundung bei Summenbildung	Y	R. Meier

Business Process Modeling

Informationsmodell



Requirements Engineering

informale Formulierung

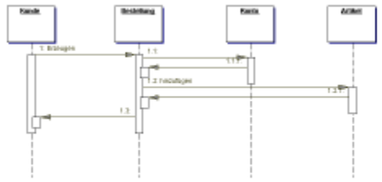
Ein *Bestellvorgang* setzt sich aus
der *Bonitätsprüfung* des *Auftraggebers*,
der *Ausbuchung* aus dem *Warenkatalog*
sowie der *Rechnungserstellung* zusammen

Anwendungsfälle

Object oriented Analysis

Integration, Verteilung

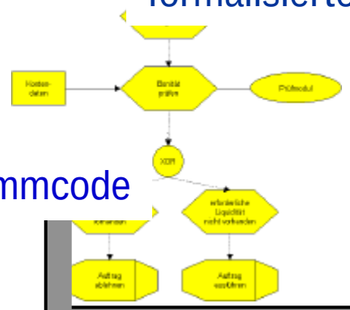
Dynamische Abläufe



formale Darstellung

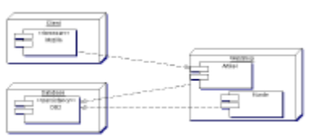
```
context Kunde inv:
self.allInstances -> forAll (k1, k2| k1 <> k2
   implies k1.kundenNr <> k2.kund
context Artikel inv:
self.allInstances -> forAll (a1, a2| a1
                              <> a2.artikel
```

formalisierte Darstellung

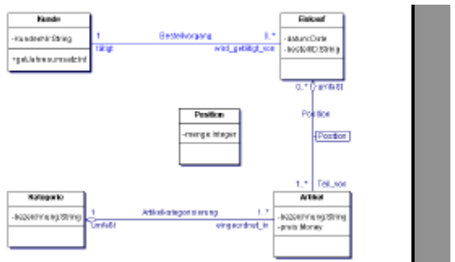


Programmcode

Einsatzszenario



Datenmodell



QS, Test, Abnahme

Coding / Customizing

Object oriented Design

OMG's XML Metadata Interchange

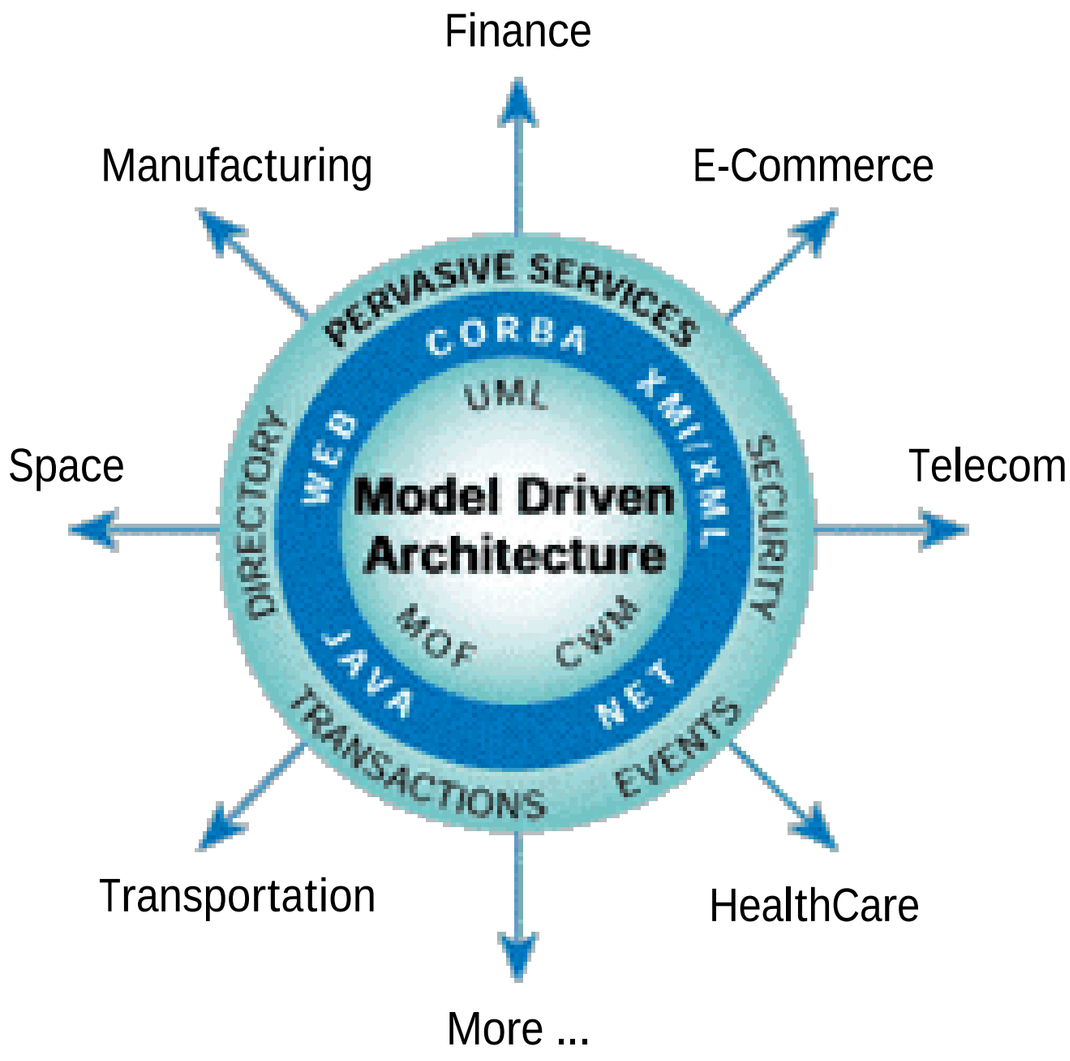
- Integriert
  - *Unified Modeling Language* (UML)
  - *Meta Object Facility* (MOF)
- Teil der *Model Driven Architecture (MDA)*

- Anwendungsgebiete
  - Modellaustausch
  - Metamodellaustausch
    - Middleware-bezogene Komponenteninformation (CORBA Components)
    - Austausch von Data Warehouse Models (CWM)
    - Vertikale Datenintegration (e.g. clinical information)

OMG's XML Metadata Interchange

- Verabschiedeter OMG-Standard
  - erarbeitet durch: Unisys, IBM, DSTC, Oracle, Platinum, Fujitsu, Softeam, Recerca Informatica, DaimlerChrysler
  - unterstützt durch: Genesis, Inline, Rational, Select, Sprint, Cayenne, Sybase, Xerox, Boeing, Ardent, MCI Systemhouse, Aviatis, ICONIX, Integrated Systems, Verilog, Nihon Unisys, NTT, Telefonica I+D, NCR, Universitat Politecnica de Catalunya,

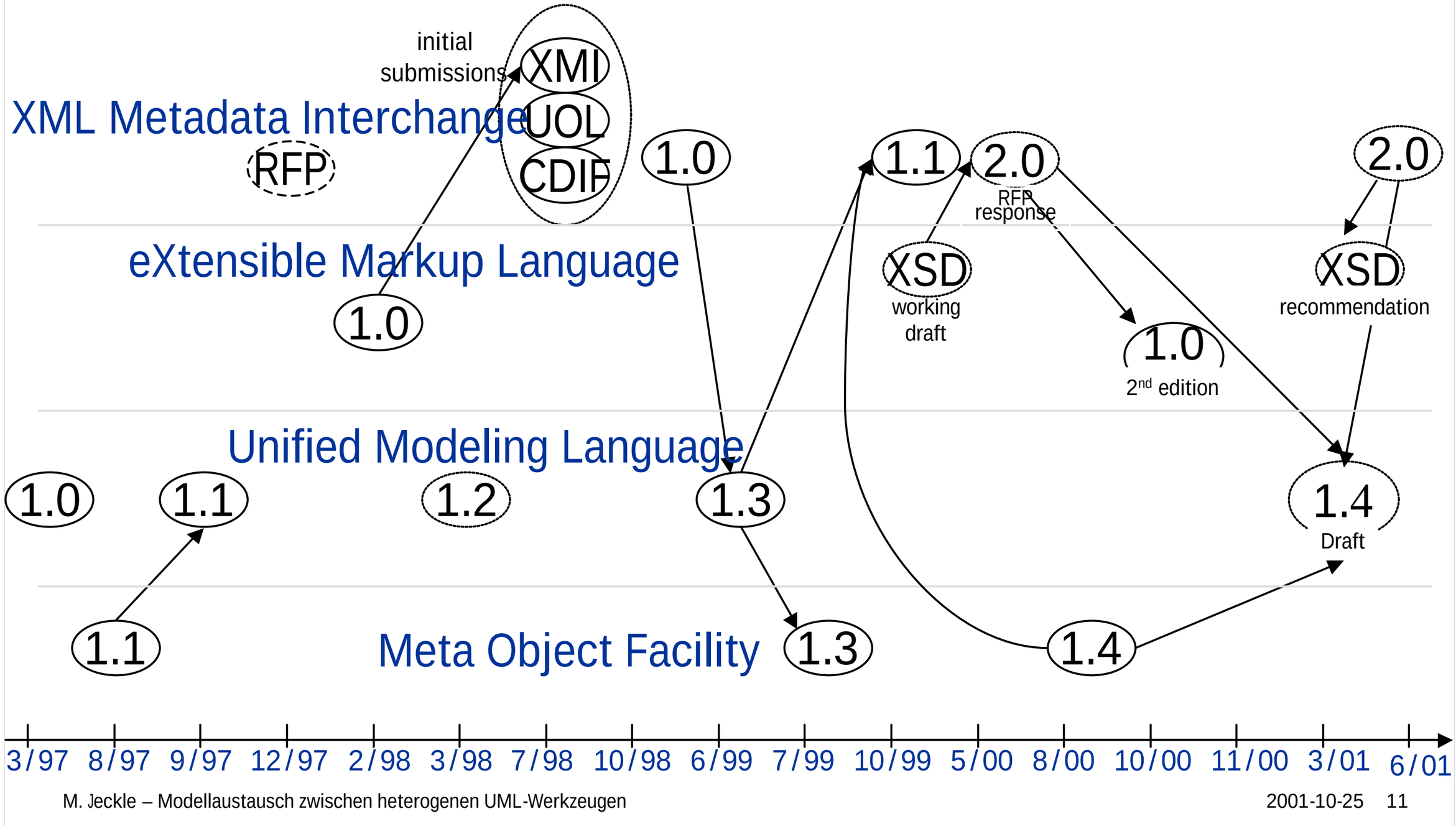
OMG's Model Driven Architecture



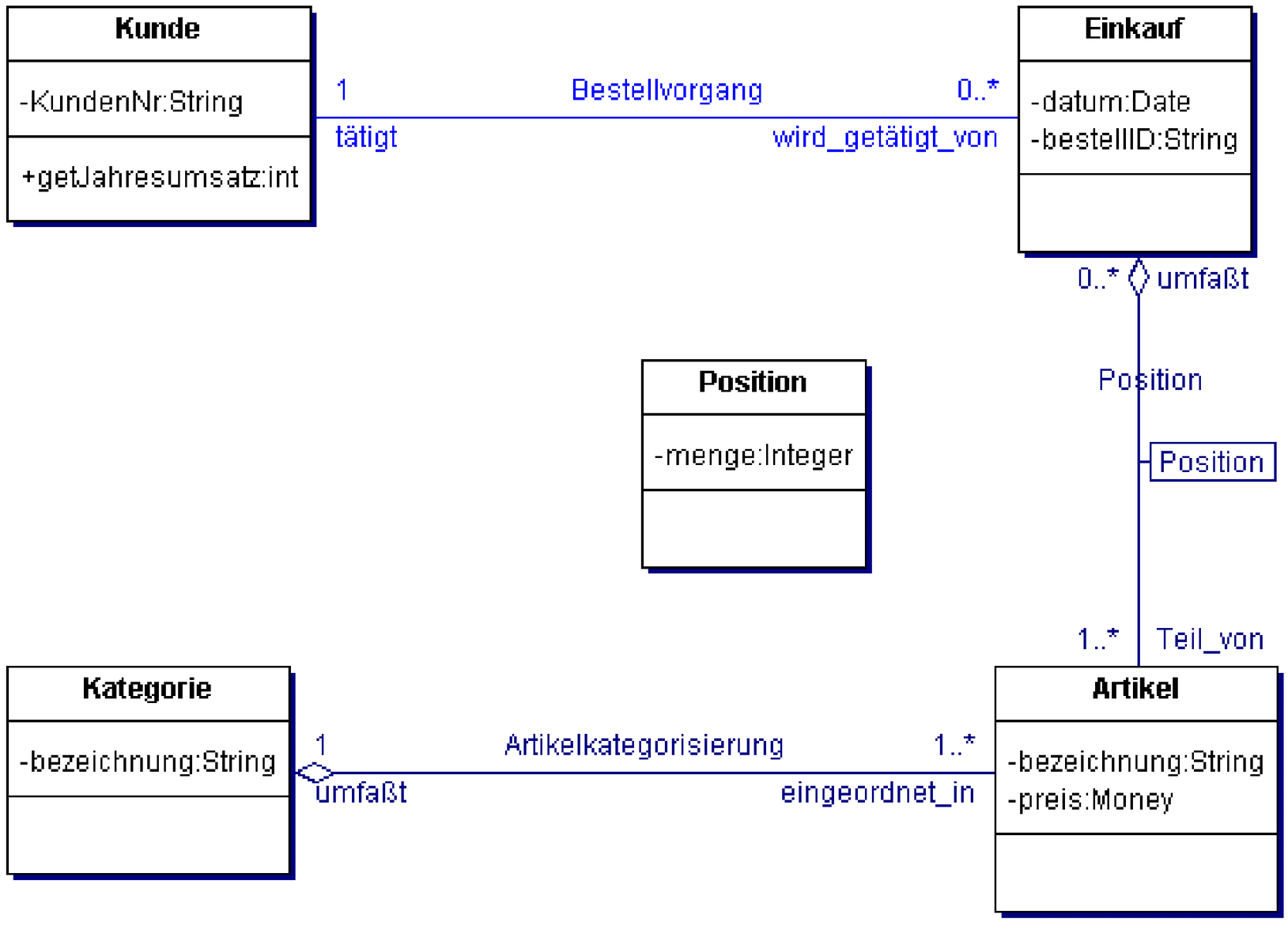
Zielsetzung:

- Schnelle „natürliche“ Entwicklung plattformübergreifender interoperabler Lösungen
- Steigerung der Portierbarkeit auf andere Middleware-Plattformen
- Bereitstellung von standardisierten Diensten für verschiedene Domänen
- Produktivitätssteigerung im Einsatz verschiedener Middleware-Lösungen

XMI – Abhängigkeiten zu anderen Standards

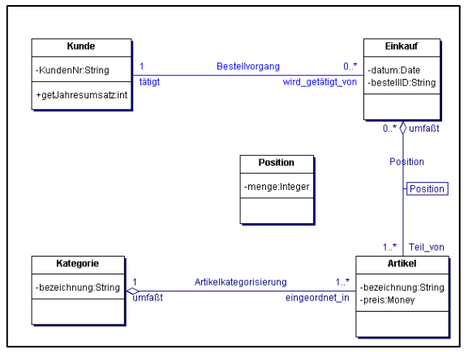


XMI im Einsatz: XML-Darstellung von UML-Modellen



XMI im Einsatz: XML-Darstellung von UML-Modellen

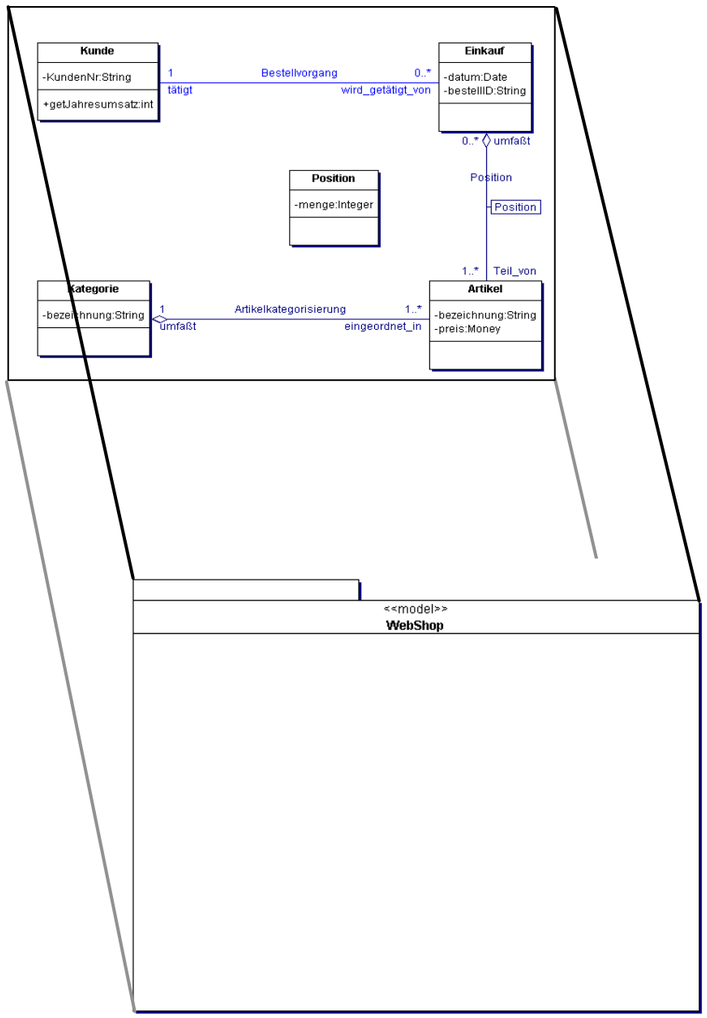
XMI-Header -- Verwaltungsinformation




```
<?xml version = '1.0' encoding = 'ISO-8859-1' ?>
<!DOCTYPE XMI SYSTEM  'uml.dtd'>
<XMI xmi.version = '1.0'>
  <XMI.header>
    <XMI.documentation>
      <XMI.exporter>Together</XMI.exporter>
      <XMI.exporterVersion>5.0</XMI.exporterVersion>
    </XMI.documentation>
    <XMI.metamodel xmi.name = 'UML' xmi.version = '1.1'/>
  </XMI.header>
<XMI.content>
```

XMI im Einsatz: XML-Darstellung von UML-Modellen

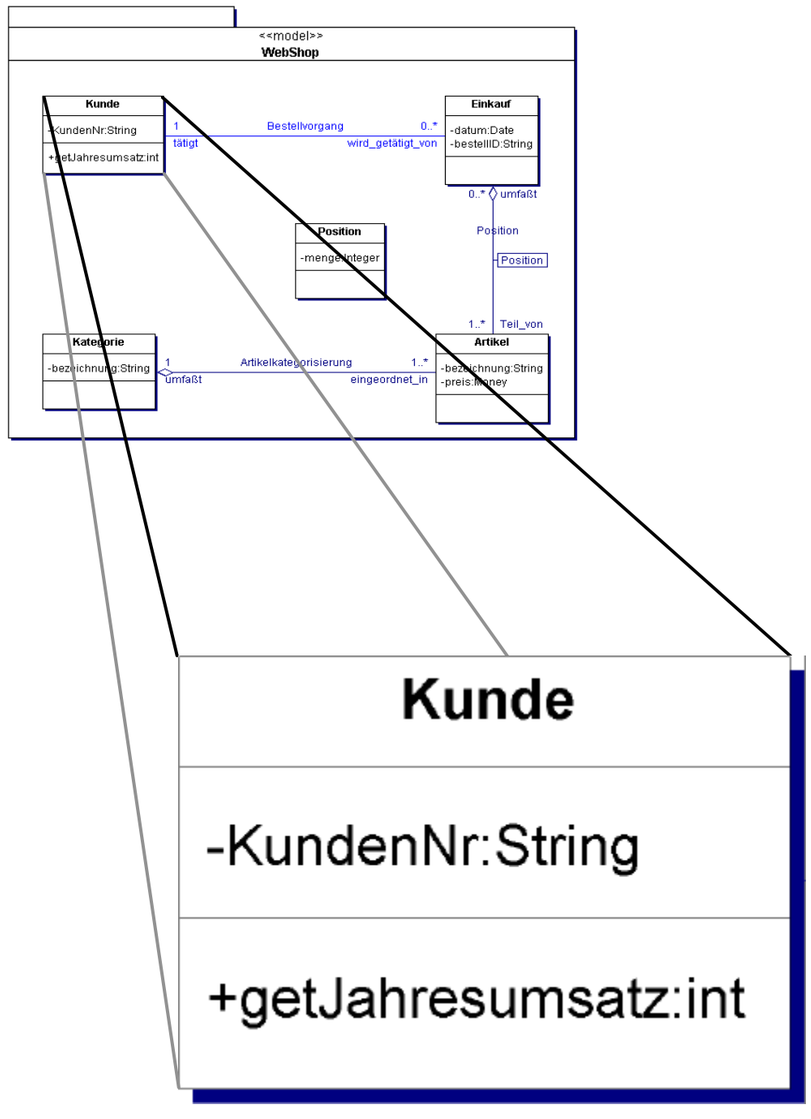
Darstellung des Modells




```
<Model_Management.Model xmi.id = 'txmiid1' >

<Foundation.Core.ModelElement.name>
  WebShop
</Foundation.Core.ModelElement.name>

 <Foundation.Core.ModelElement.visibility
    xmi.value = 'private'/>

<Foundation.Core.GeneralizableElement.isRoot
   xmi.value = 'false'/>

<Foundation.Core.GeneralizableElement.isLeaf
   xmi.value = 'false'/>

<Foundation.Core.GeneralizableElement.isAbstract
   xmi.value = 'false'/>
```

XMI im Einsatz: XML-Darstellung von UML-Modellen

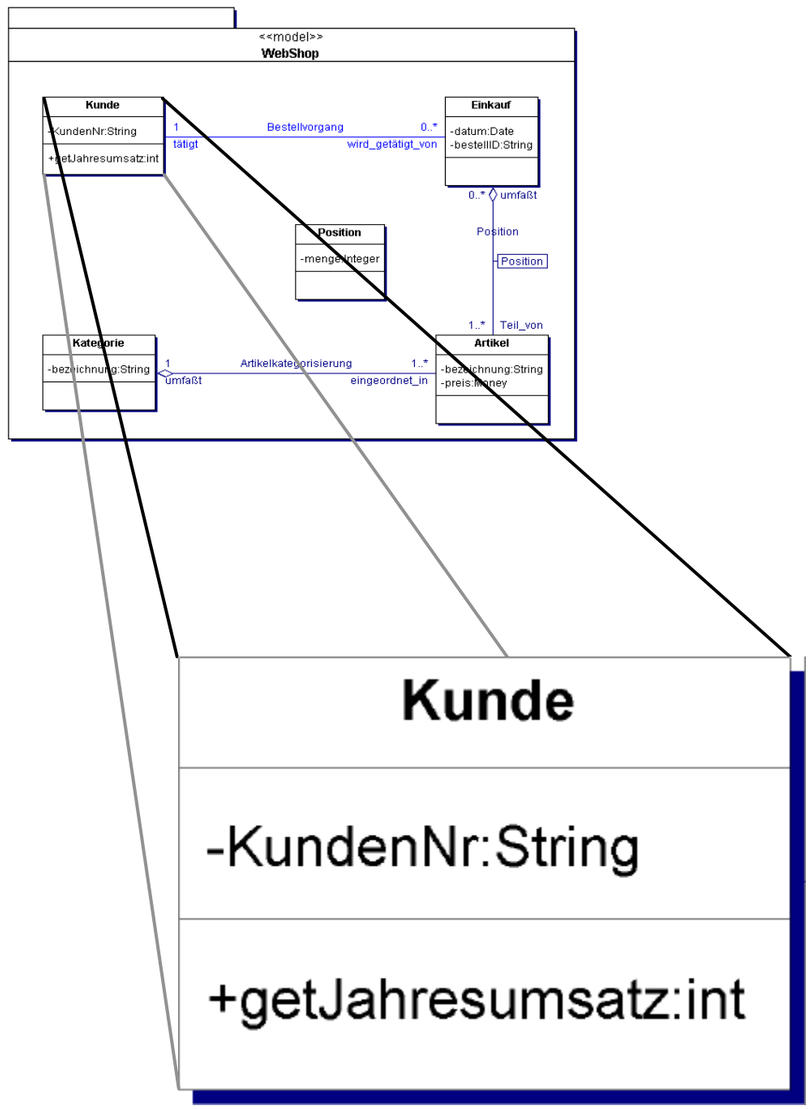
Darstellung der Klassen




```
<Foundation.Core.Class xmi.id = 'txmiid12' >

 <Foundation.Core.ModelElement.name>
   Kunde
</Foundation.Core.ModelElement.name>

  <Foundation.Core.ModelElement.visibility
   xmi.value = 'public' />

  <Foundation.Core.GeneralizableElement.isRoot
   xmi.value = 'false' />

<Foundation.Core.GeneralizableElement.isLeaf
 xmi.value = 'false' />

<Foundation.Core.GeneralizableElement.isAbstract
 xmi.value = 'false' />

<Foundation.Core.Class.isActive xmi.value = 'false' />

<Foundation.Core.ModelElement.namespace>

   <Model_Management.Package xmi.idref = 'txmiid1' />

</Foundation.Core.ModelElement.namespace>
```

XMI im Einsatz: XML-Darstellung von UML-Modellen

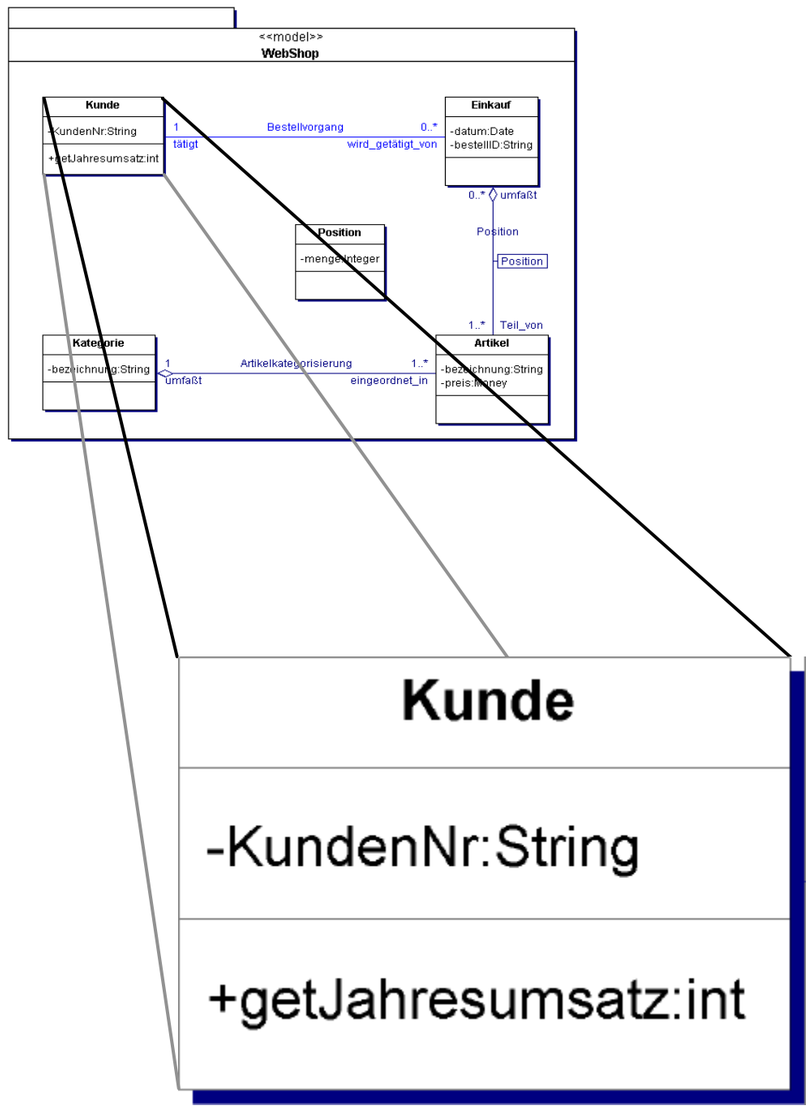
Darstellung der Attribute




```
<Foundation.Core.Attribute xmi.id = 'txmiid14.a' >
  <Foundation.Core.ModelElement.name>
    KundenNr
  </Foundation.Core.ModelElement.name>
  <Foundation.Core.ModelElement.visibility
    xmi.value = 'private' />
  <Foundation.Core.Feature.ownerScope
    xmi.value = 'classifier'/>
  <Foundation.Core.StructuralFeature.multiplicity>
    1..1   default
  </Foundation.Core.StructuralFeature.multiplicity>
  <Foundation.Core.StructuralFeature.changeable
    xmi.value = 'none'/>
  <Foundation.Core.StructuralFeature.targetScope
    xmi.value = 'instance'/>   default
```

XMI im Einsatz: XML-Darstellung von UML-Modellen

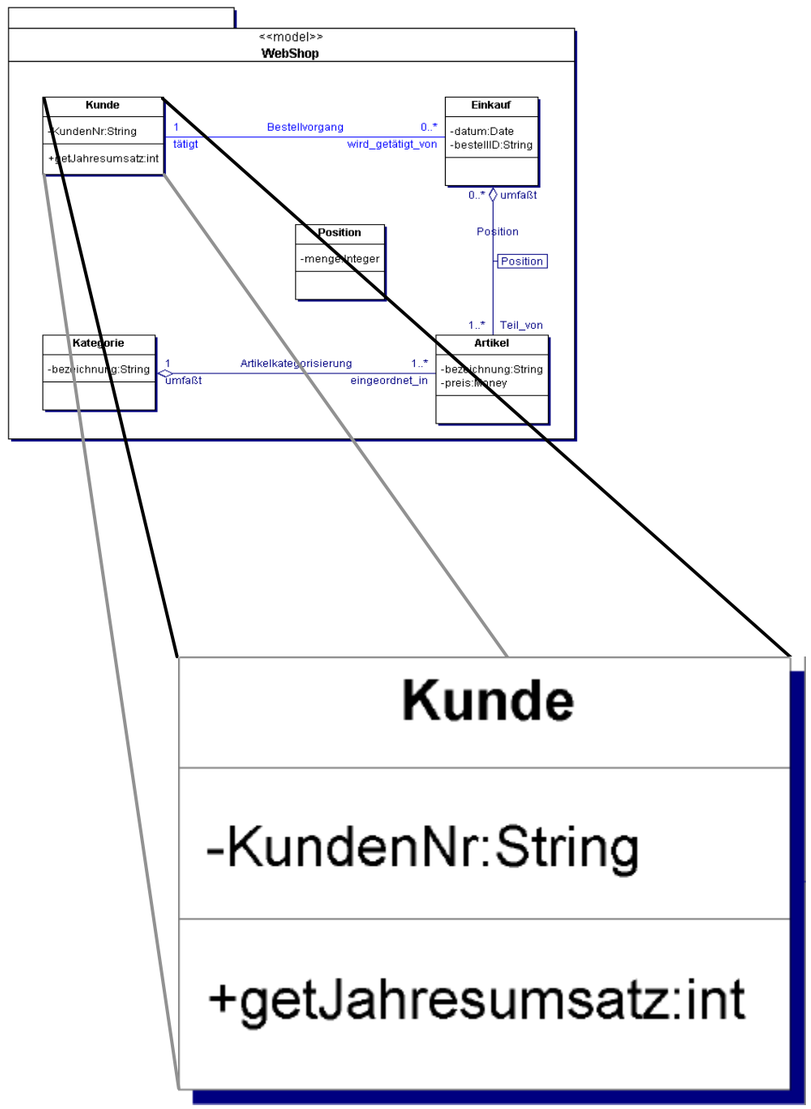
Darstellung der Operationen




```
<Foundation.Core.Operation xmi.id =
'txmiid17.O' >
<Foundation.Core.ModelElement.name>
   getJahresumsatz
</Foundation.Core.ModelElement.name>

...

<Foundation.Core.BehavioralFeature.isQuery
   xmi.value = 'false'/>

...

<Foundation.Core.Operation.concurrency
   xmi.value = 'sequential'/>

...
```

XMI im Einsatz: XML-Darstellung von UML-Modellen

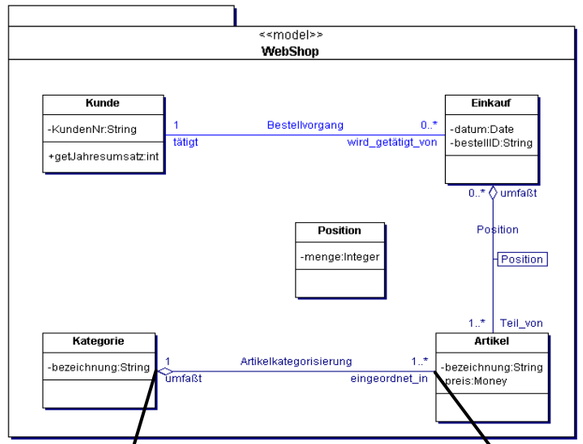
Darstellung der Datentypen




```
<Foundation.Core.DataType xmi.idref = 'txmiid4' />

<Foundation.Core.DataType xmi.id = 'txmiid4' >
  <Foundation.Core.ModelElement.name>
    String
  </Foundation.Core.ModelElement.name>
  <Foundation.Core.ModelElement.visibility
    xmi.value = 'private'/>
  <Foundation.Core.GeneralizableElement.isRoot
    xmi.value = 'false'/>
  <Foundation.Core.GeneralizableElement.isLeaf
    xmi.value = 'false'/>
  <Foundation.Core.GeneralizableElement.isAbstract
    xmi.value = 'false'/>
</Foundation.Core.DataType>
```

XMI im Einsatz: XML-Darstellung von UML-Modellen

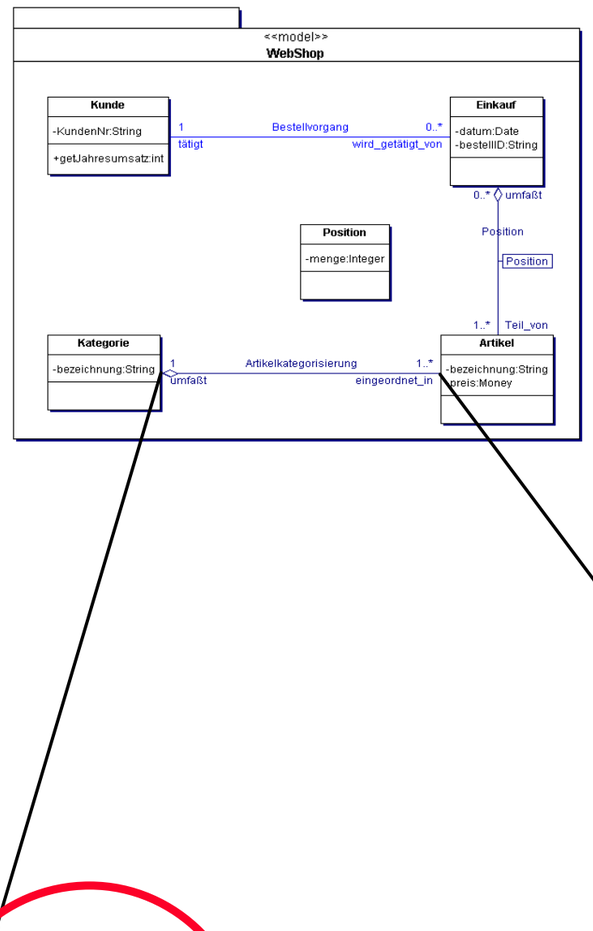
Darstellung der Assoziationen




```
<Foundation.Core.Association xmi.id = 'txmiidf.ASS'>

<Foundation.Core.ModelElement.name>
  Artikelkategorisierung
</Foundation.Core.ModelElement.name>

<Foundation.Core.ModelElement.visibility
  xmi.value = 'private' />

<Foundation.Core.GeneralizableElement.isRoot
  xmi.value = 'false'/>

<Foundation.Core.GeneralizableElement.isLeaf
  xmi.value = 'false'/>

<Foundation.Core.GeneralizableElement.isAbstract
  xmi.value = 'false'/>
```

XMI im Einsatz: XML-Darstellung von UML-Modellen

Darstellung der Assoziationen




```
<Foundation.Core.AssociationEnd xmi.id = 'txmiid12.CE' >

<Foundation.Core.ModelElement.name>
umfaßt
</Foundation.Core.ModelElement.name>

<Foundation.Core.ModelElement.visibility
   xmi.value = 'public' />

...

<Foundation.Core.AssociationEnd.aggregation xmi.value =
'aggregate' />
```

CASE-Werkzeuge mit XMI-Unterstützung zum Austausch von UML-Modellen

- Verfügbare Werkzeuge mit XMI-Unterstützung

IDEOGRAMIC APS

Das innovative Systemhaus MID

ObjeXion Software

Poseidon for UML

Rhapsody®

FCJABA

TogetherSoft

magicdraw

objectiF®
Für die Entwicklung objektorientierter und komponentenbasierter Software-systeme

Visual Object Modelers

SOFTEAM
Think Object

Telelogic

Together 5.5 -- example

File Edit Object Search View Options Tools Help

Documentation
Quality Assurance
Inspector Property Builder...
Code Template Expert...
Database Import/Export
Import
Export
DTD/XSD Import/Export
J2EE Deployment Expert...
Web Services Expert...
J2EE Module Import...
Format Project Source...
Synchronize with external changes

Export to XML...

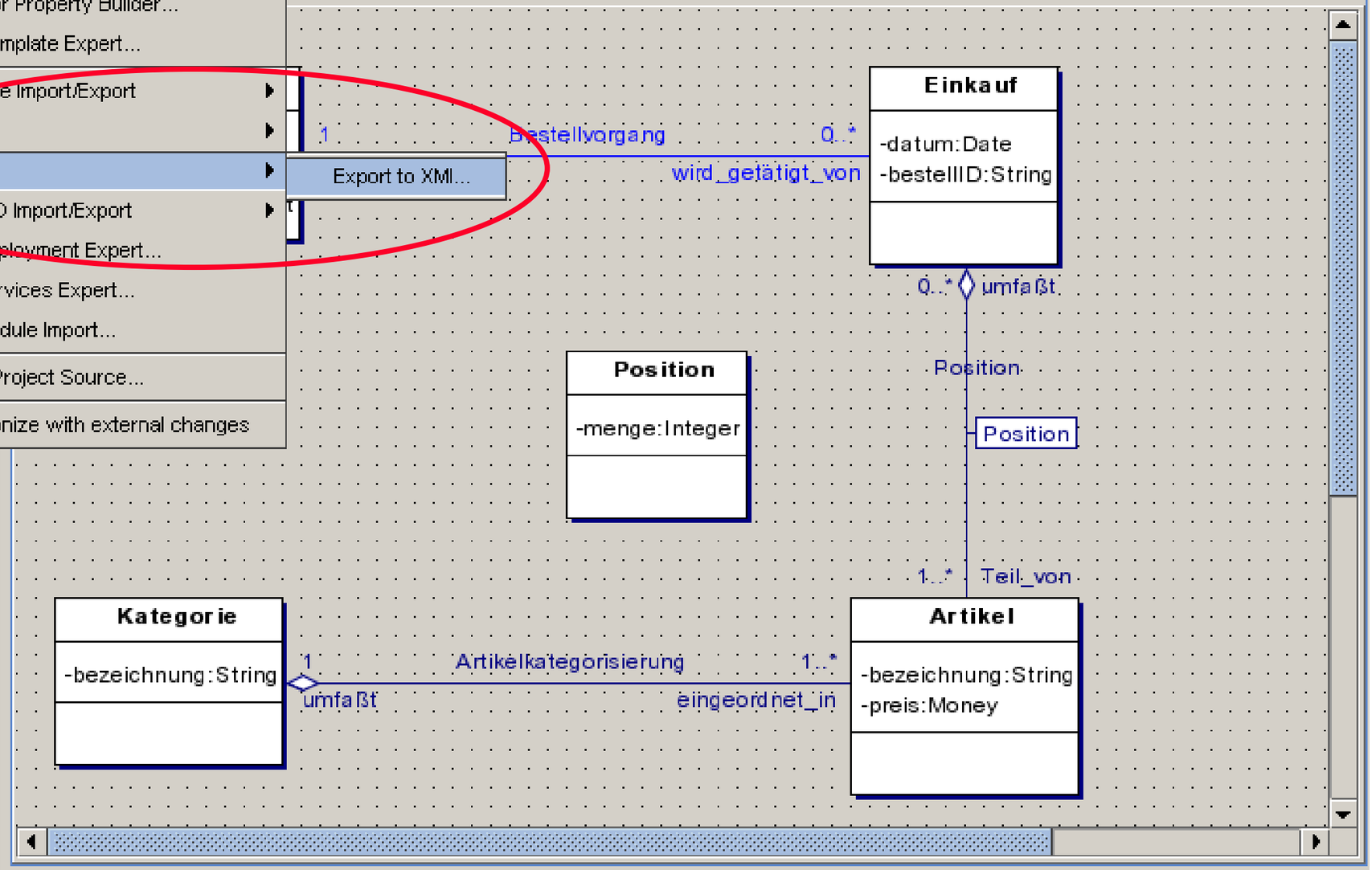
example
<default>
Artikel
Einkauf
Kategorie
Position

Properties of <default>

Description HTMLdoc Requirements
Properties Hyperlink

Name	Value
diagram type	Class Diagram
name	<default>
package	<default>
stereotype	
alias	

Press Ctrl+Enter to finish editing and close Inspector




```
public class Kunde {
    public int getJahresumsatz(int jahr) {
    }

    String KundenNr;
```

Kategorie.cs Kunde.cs

Progress Insert Ln: 5 Col: 1

Together 5.5 -- example

File Edit Object Search View Options Tools Help

Documentation
Quality Assurance
Inspector Property Builder...
Code Template Expert...
Database Import/Export
Import
Export
DTD/XSD Import/Export
J2EE Deployment Expert...
Web Services Expert...
J2EE Module Import...
Format Project Source...
Synchronize with external changes

Export to XMI...

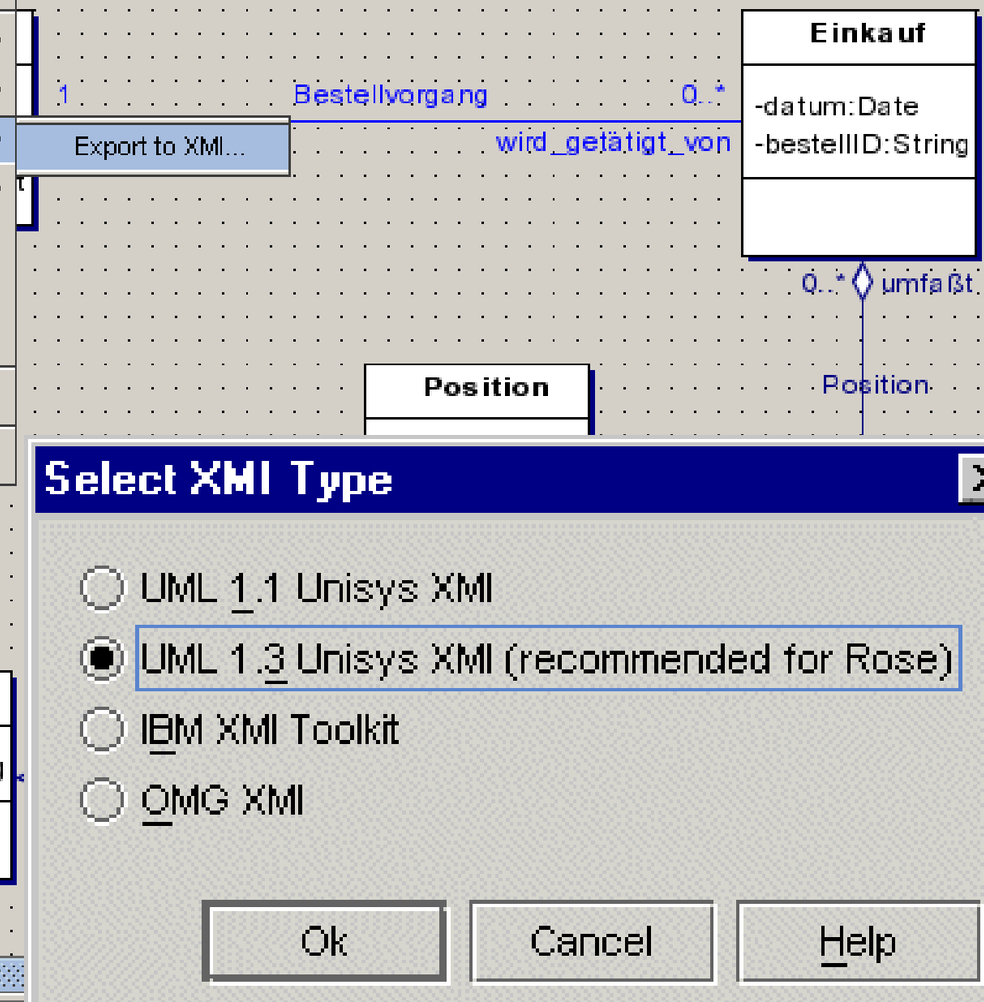
example
<default>
Artikel
Einkauf
Kategorie
Position

Properties of <default>

Description HTMLdoc Requirements
Properties Hyperlink

Name	Value
diagram type	Class Diagram
name	<default>
package	<default>
stereotype	
alias	

Press Ctrl+Enter to finish editing and close Inspector



Select XMI Type

- UML 1.1 Unisys XMI
- UML 1.3 Unisys XMI (recommended for Rose)
- IBM XMI Toolkit
- OMG XMI

Ok Cancel Help

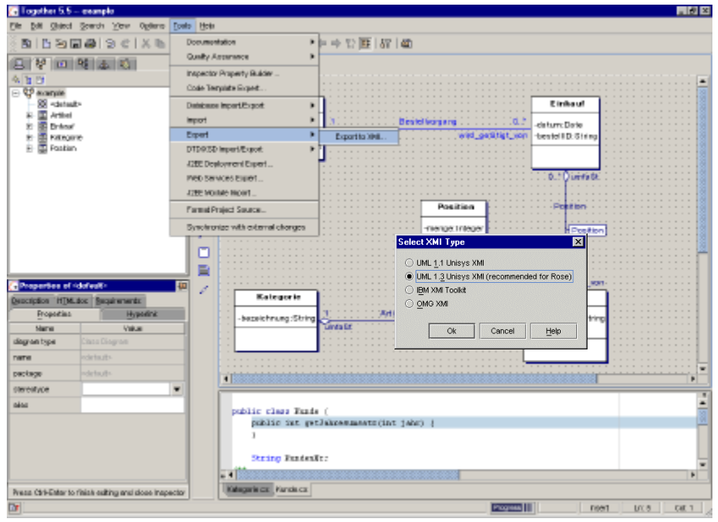
```
public class Kunde {
    public int getJahresumsatz(int jahr) {
    }

    String KundenNr;
```

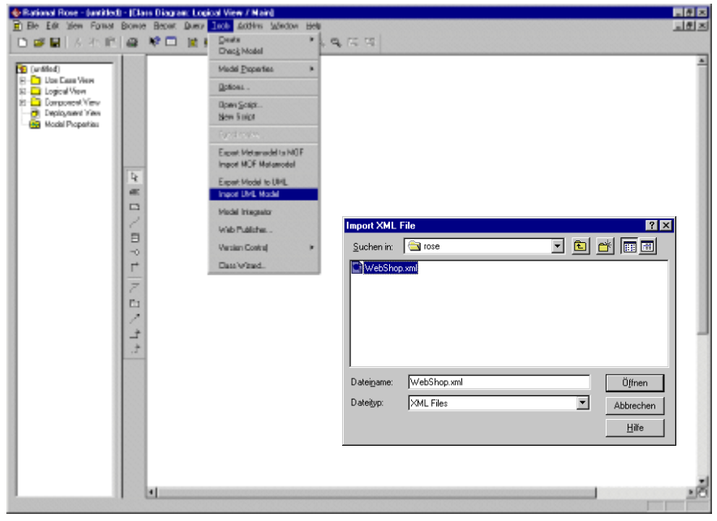
Kategorie.cs Kunde.cs

Progress Insert Ln: 5 Col: 1

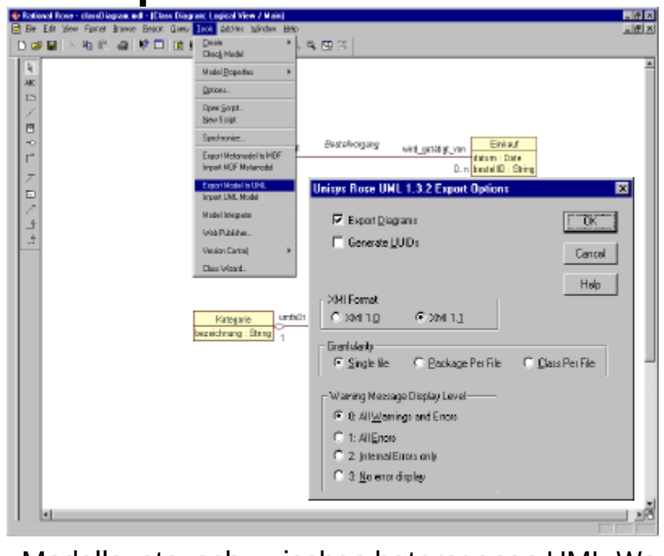
XMI – Interoperabiltität von CASE-Werkzeugen

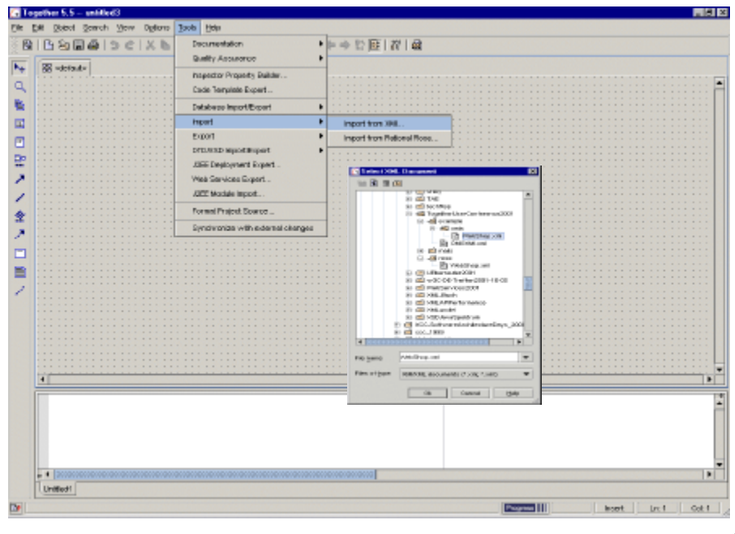
- Export Together → Import Rational Rose





- Export Rational Rose → Import Together

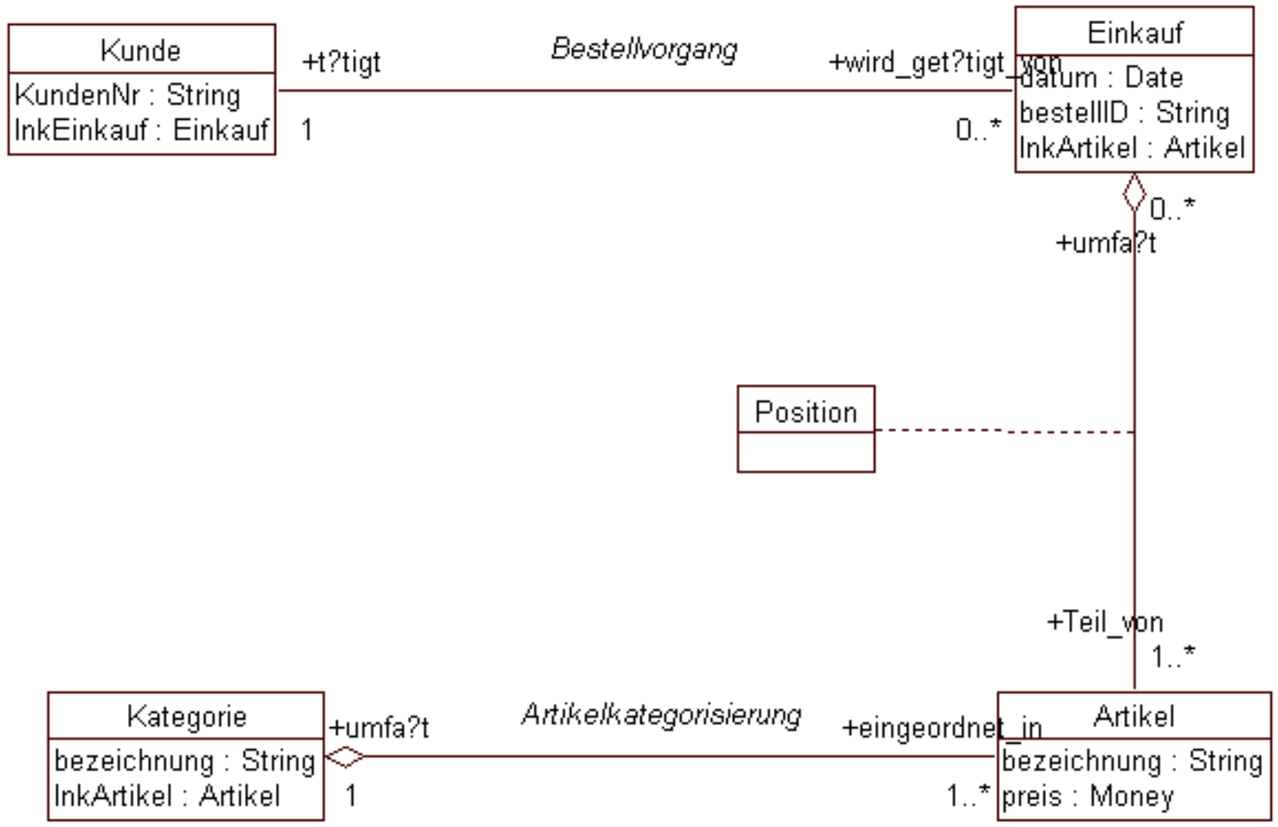




Rational Rose - (untitled) - [Class Diagram: Logical View / Main]

File Edit View Format Browse Report Query Tools Add-Ins Window Help

(untitled)
- Use Case View
- Logical View
  - xmis
  - Main
  - <<default>>
  - Artikel
  - Einkauf
  - Kategorie
  - Kunde
  - Position
  - Associations
- Component View
- Deployment View
- Model Properties



For Help, press F1

Together 5.5 -- untitled2

File Edit Object Search View Options Tools Help

untitled2
- <default>
- Artikel
- Einkauf
- Kategorie
- Kunde
- Position

Properties of <default>

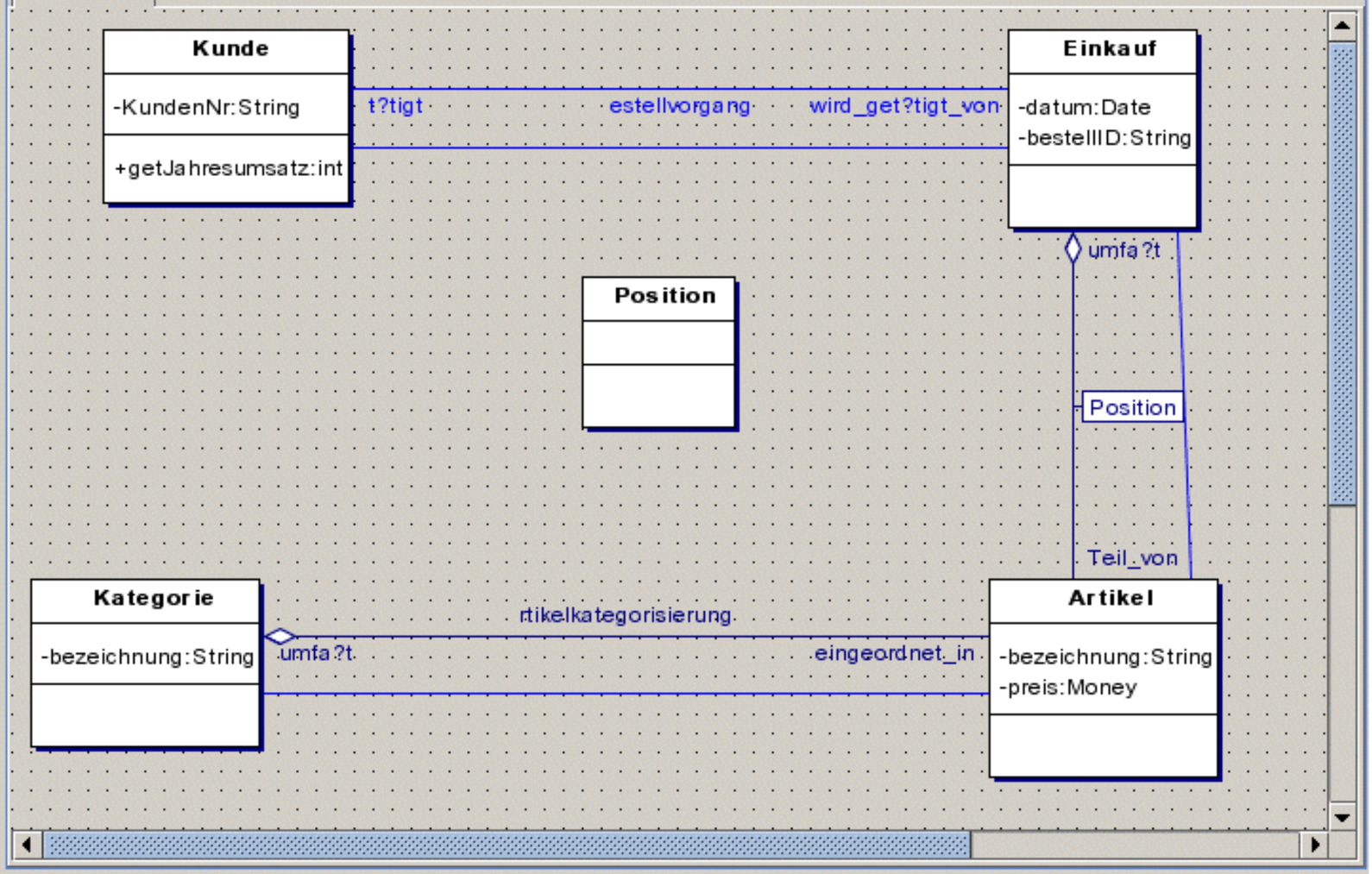
Description HTMLdoc Requirements

Properties Hyperlink

Name	Value
diagram type	Class Diagram
name	<default>
package	<default>
stereotype	
alias	

Press Ctrl+Enter to finish editing and close Inspector

<default>



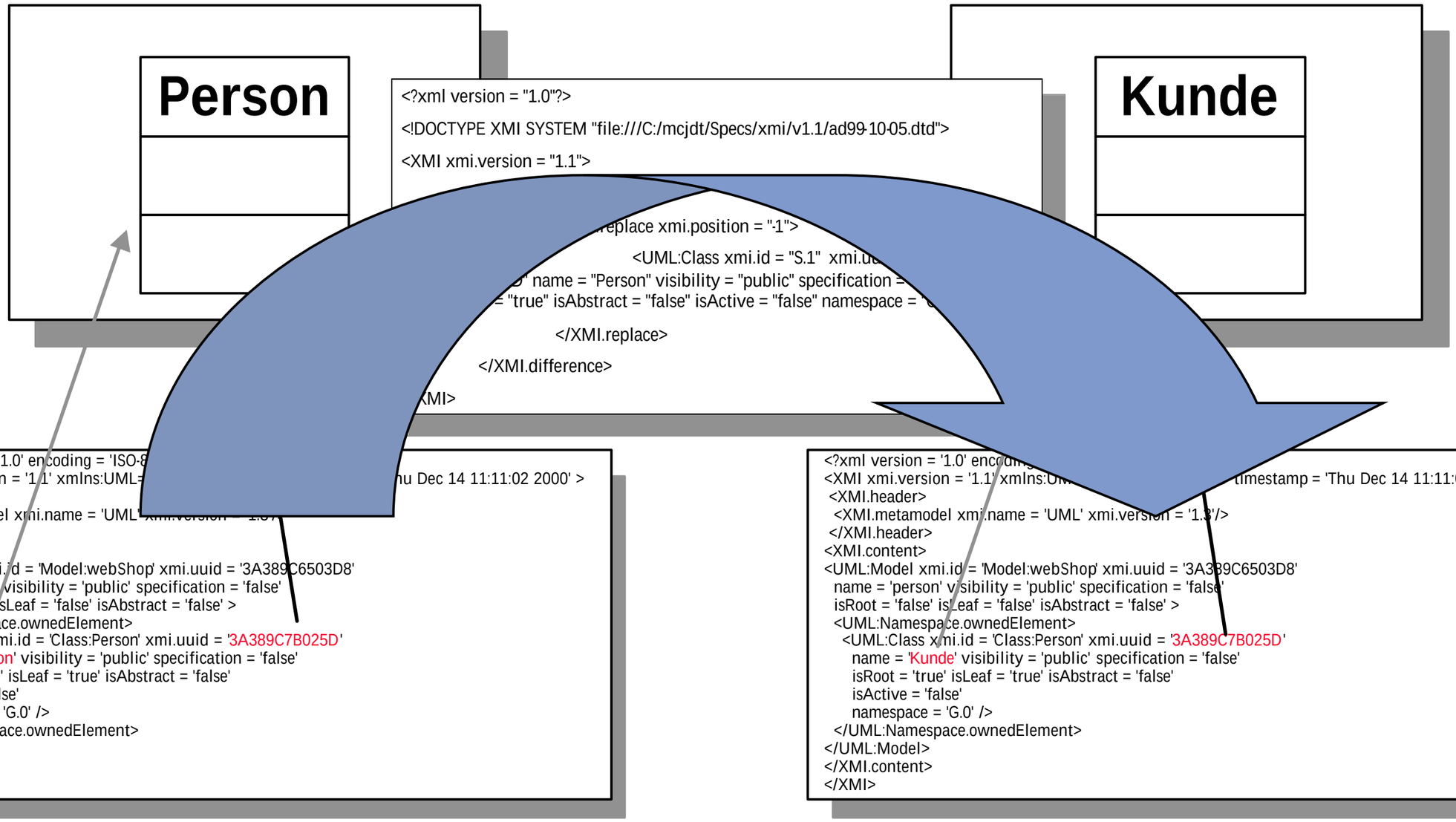

```
class Kategorie {
    String bezeichnung;
    Artikel lnkArtikel;
/**
 * @label rtikelkategorisierung
 * @link aggregation
```

Kategorie.cs

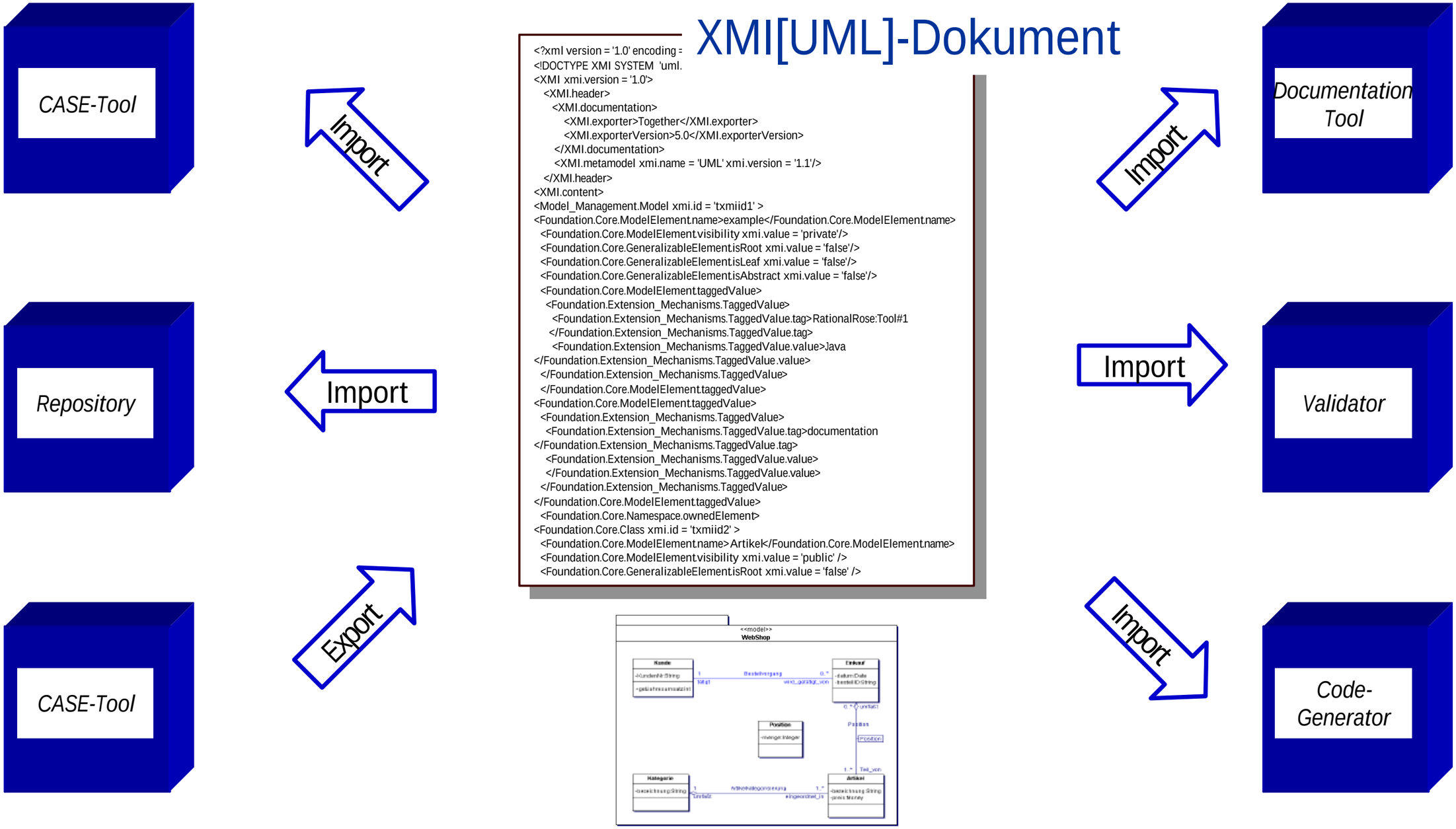
Progress Insert Ln: 1 Col: 1

Onlinekopplung von CASE-Werkzeugen

- Hinzufügen, Löschen und Ändern von Modellelementen

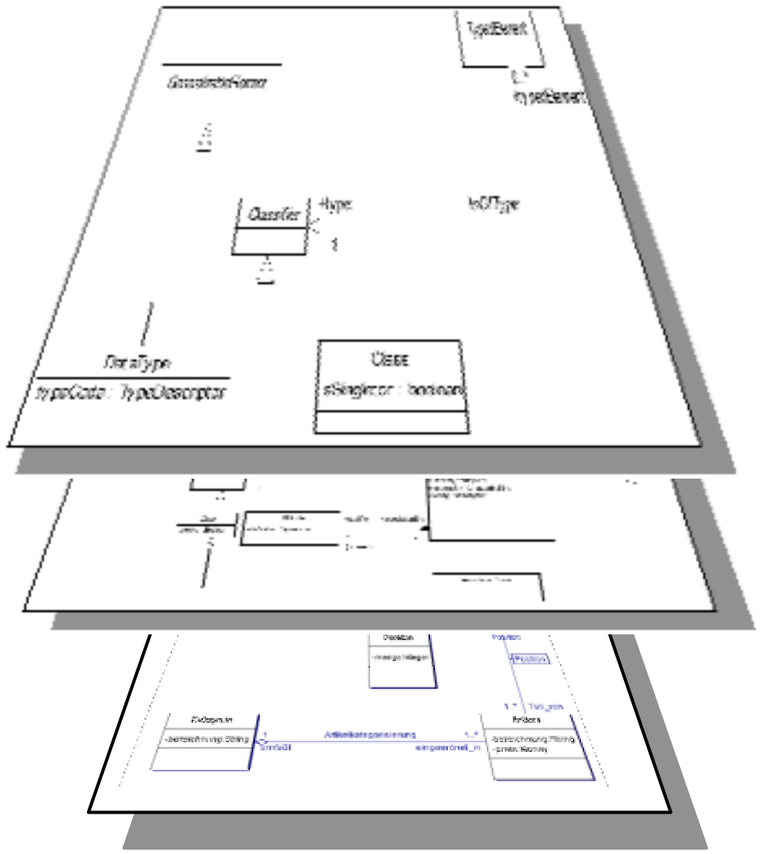


XMI im Einsatz: Anwendungsgebiete



XMI im Einsatz: Austausch vollständiger Modellierungssprachen

- Voraussetzung: Zur Modellierungssprache existiert ein explizites Metamodell
- Metamodell verfügt über MOF als Meta-Metamodell

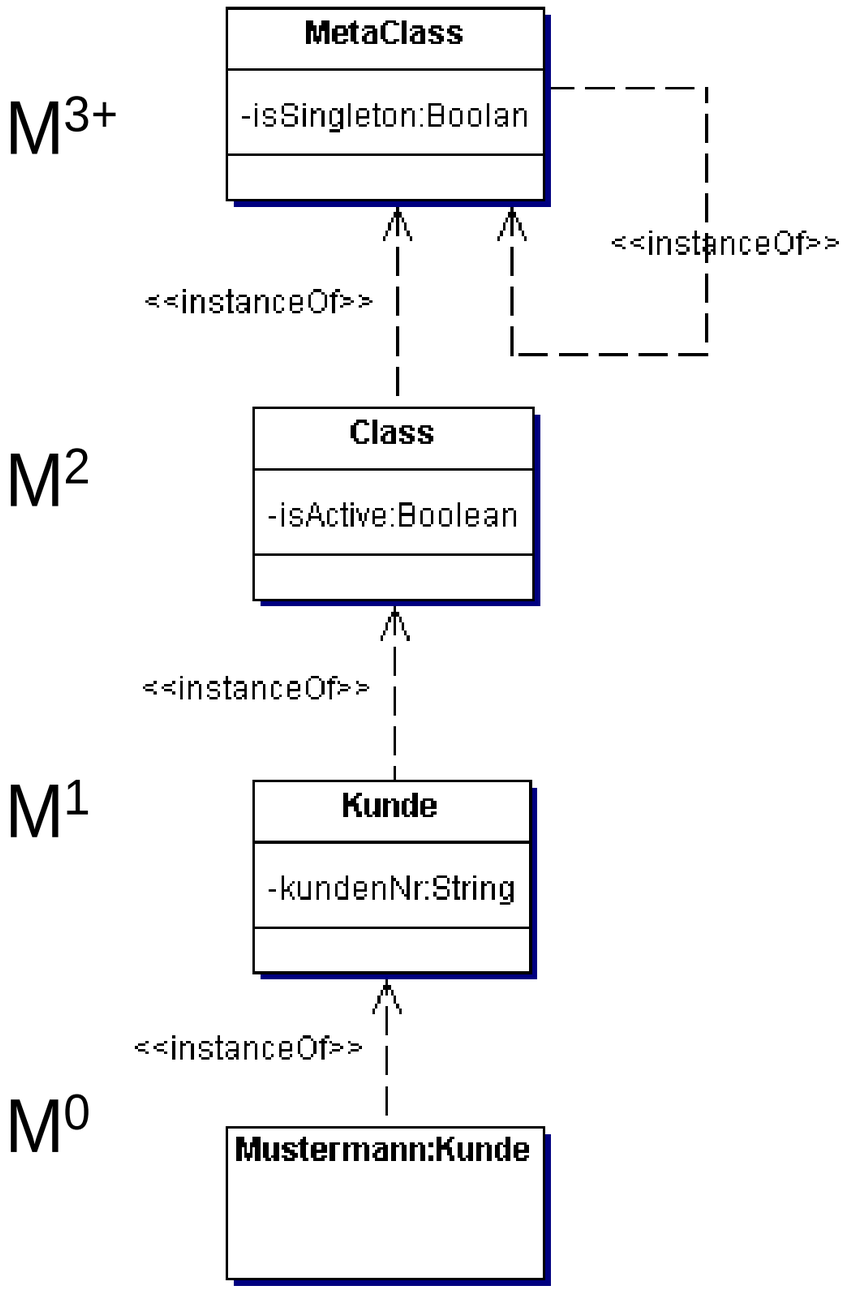


Meta Object Facility

Unified Modeling Language

Web Shop

Die vier-Schichten Metamodellarchitektur der OMG



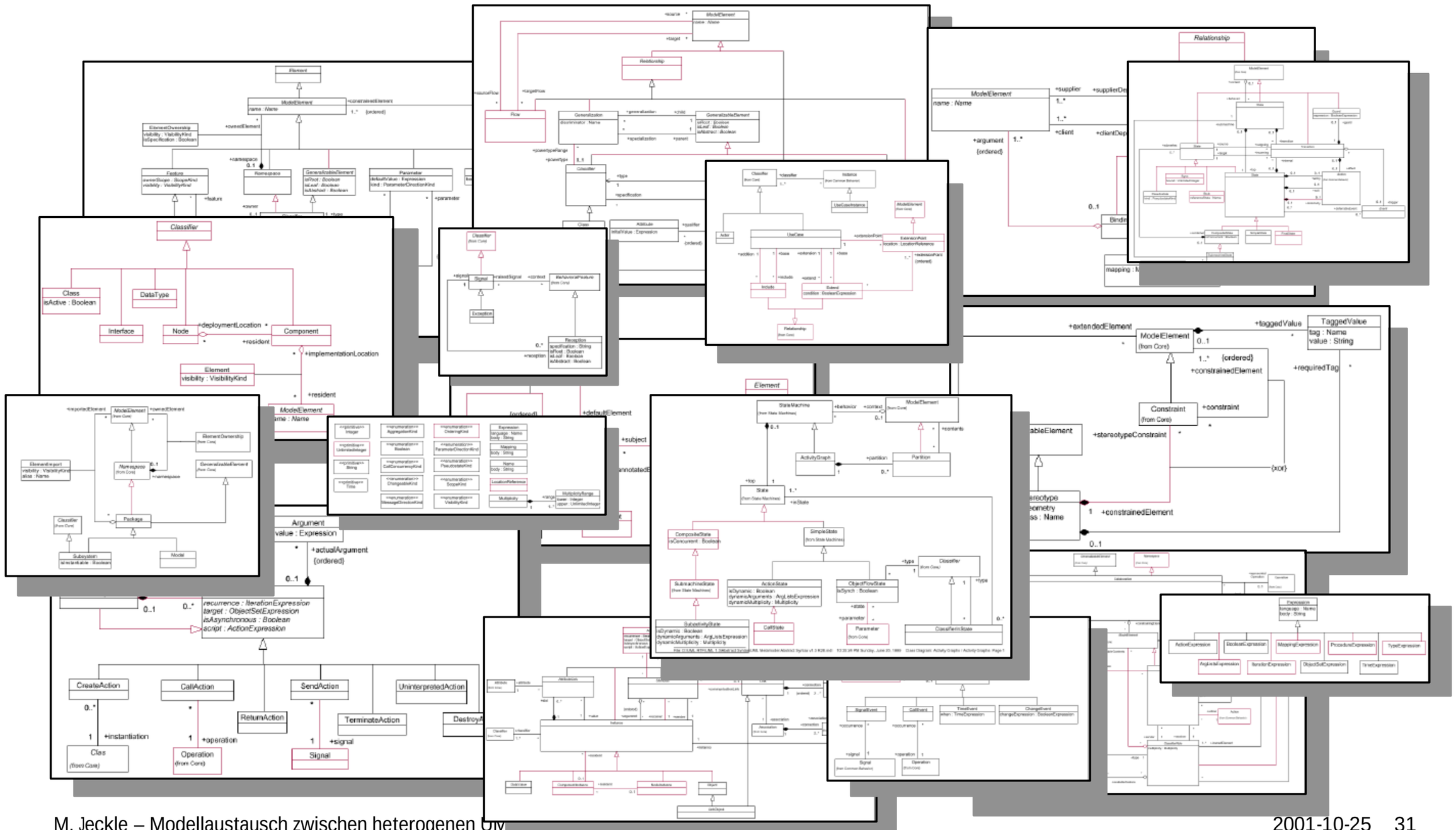
Meta-Metaklassenebene
Meta-Metamodell

Metaklassenebene
Metamodell

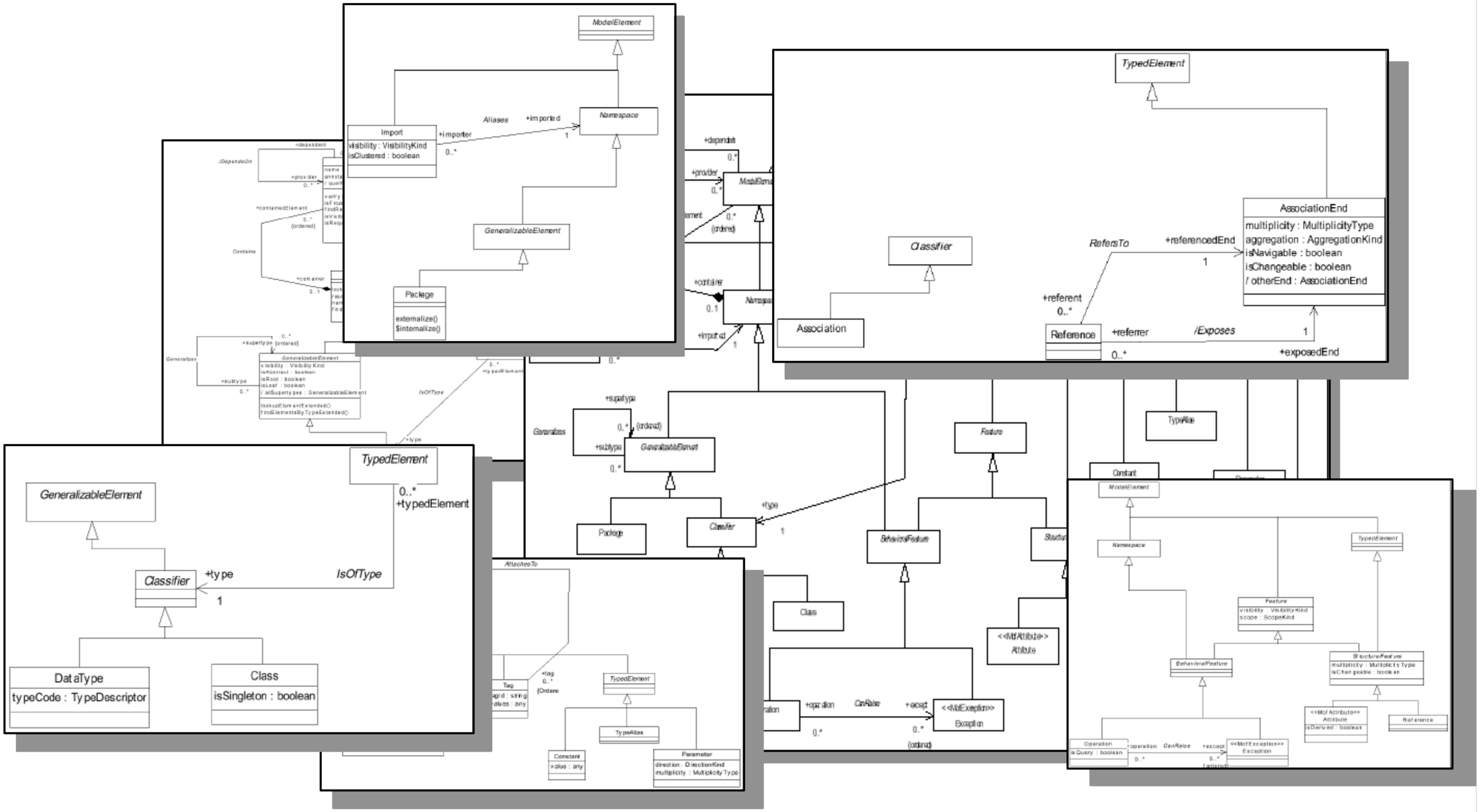
Modellebene

Ausprägungs- oder
Instanzenebene

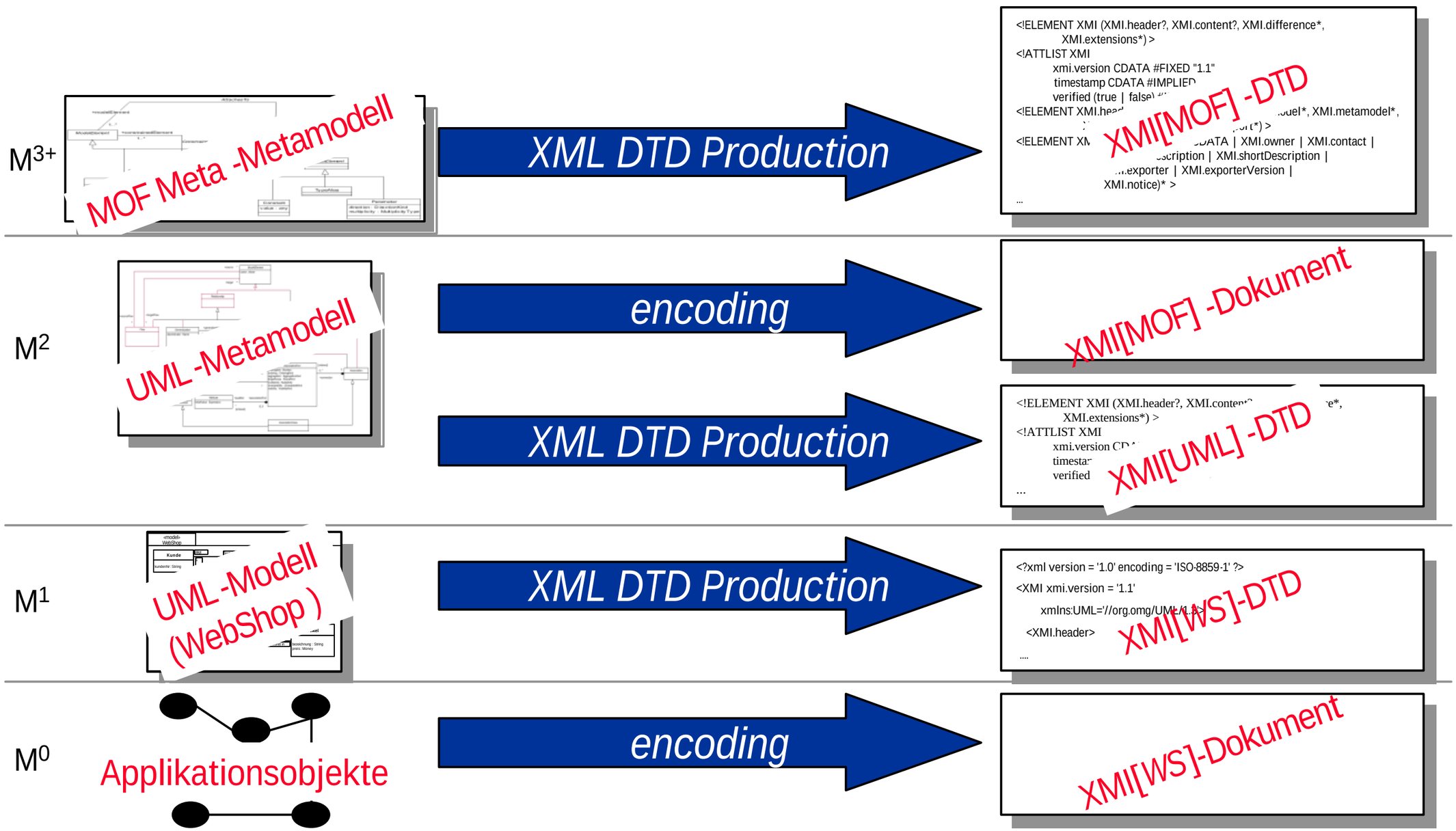
UML v1.4-Metamodell



MOF v1.3-Meta-Metamodell



Entwicklung eigener XML / XMI-Vokabulare (Stand XMI v1.1)

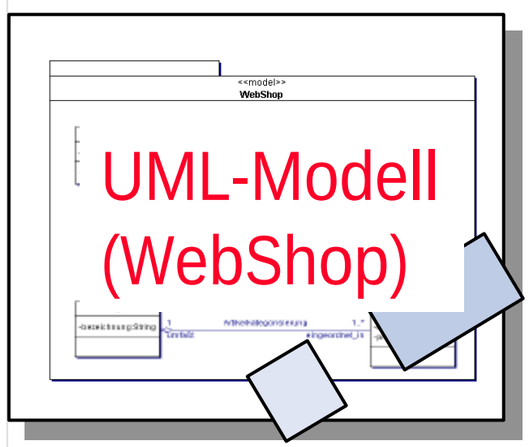


XMI im Einsatz: Erzeugung eigener XML-Vokabulare

<?xml version = '1.0' encoding = 'ISO-8859-1' ?>
<XMI xmi.version = '1.1' xmlns:UML='//org.omg/UML/1.3'>
 <XMI.header>
  <XMI.documentation>
   <XMI.exporter>Mario Jeckle</XMI.exporter>
   <XMI.exporterVersion>1.0 ;)</XMI.exporterVersion>
  </XMI.documentation>
  <XMI.metamodel xmi.name = 'UML' xmi.ve
 </XMI.header>
<XMI.content>
<UML:Model xmi.id='
 name='WebShop' visibility='public' specification='false' isRoot='false'
  isLeaf = 'false'
  isAbst
pace.ownedElement>
L:Class xmi.id = 'Class:Kunde'
name = 'Kunde' visibility = 'public' specification = 'false'
.Root = 'true' isLeaf = 'true' isAbstract = 'false'

Web Shop als UML-Modell

XMI generation principles



Zusammenfassung: XMI-Einsatzgebiete

- (Meta-)Modellaustausch
- XML-Spracherzeugung
- Langzeitspeicherung von Modelldaten
- Dokumentationsgenerierung (XSLT)
- Modellvalidierung (Qualitätssicherung; Metriken, etc.)
- Codegenerierung
- Prototypengenerierung
- Versionsverwaltung mit textbasierten Standardwerkzeugen
- ...

XMI -- Resumé

- **Einsatzbereiche**
  - Hersteller- und Middleware-neutrales Austauschformat
  - Ideal für modellierungsgetriebene Entwicklung
  - inkrementelle iterative Entwicklungsprozesse

- **Vorteile**
  - basiert auf anerkannten und erprobten Standards
  - Kompatibilität zu verschiedenen Standards
    (OMG's UML, MOF, CWM; W3C's XML-Familie (XSLT, SOAP / XP, XSD))
  - *web enabled*
  - (vergleichsweise) leichte Implementierbarkeit
  - Architektur- und Prozeßneutral
  - überwindet Werkzeuggrenzen

Ausblick: XMI v2.0

- Berücksichtigung von UML v2.0 (Präsentationsinformation)
- Berücksichtigung der XML-Standards der zweiten Generation
  - XML Linking für inter-Dokumentreferenzen
  - W3C‘s XML Schema
- Integration von W3C‘s XML Schema
  - XMI[UML] in XML Schema
  - XMI[MOF] in XML Schema
  - Generation Principles auf XML Schema erweitert

Referenzen

XMI @ OMG:

http://www.omg.org/xml

XMI @ IBM:

http://www.software.ibm.com/ad/features/xmi.html

XMI @ XML.COM:

http.//www.xml.com/xml/pub/n/New_XML-based_OMG_standard:_XMI

IBMs XMI-Toolkit:

http://www.alphaworks.ibm.com/tech/xmitoolkit

Dieser Vortrag und weiterführende Information zum Thema:

http://www.jeckle.de/