

DAIMLERCHRYSLER

XML als Metadaten objektorientierter Programmierung

Mario Jeckle

mario.jeckle@daimlerchrysler.com

mario@jeckle.de

www.jeckle.de

DaimlerChrysler Research

dept. product development process chain (FT3/EK)

Terminologie

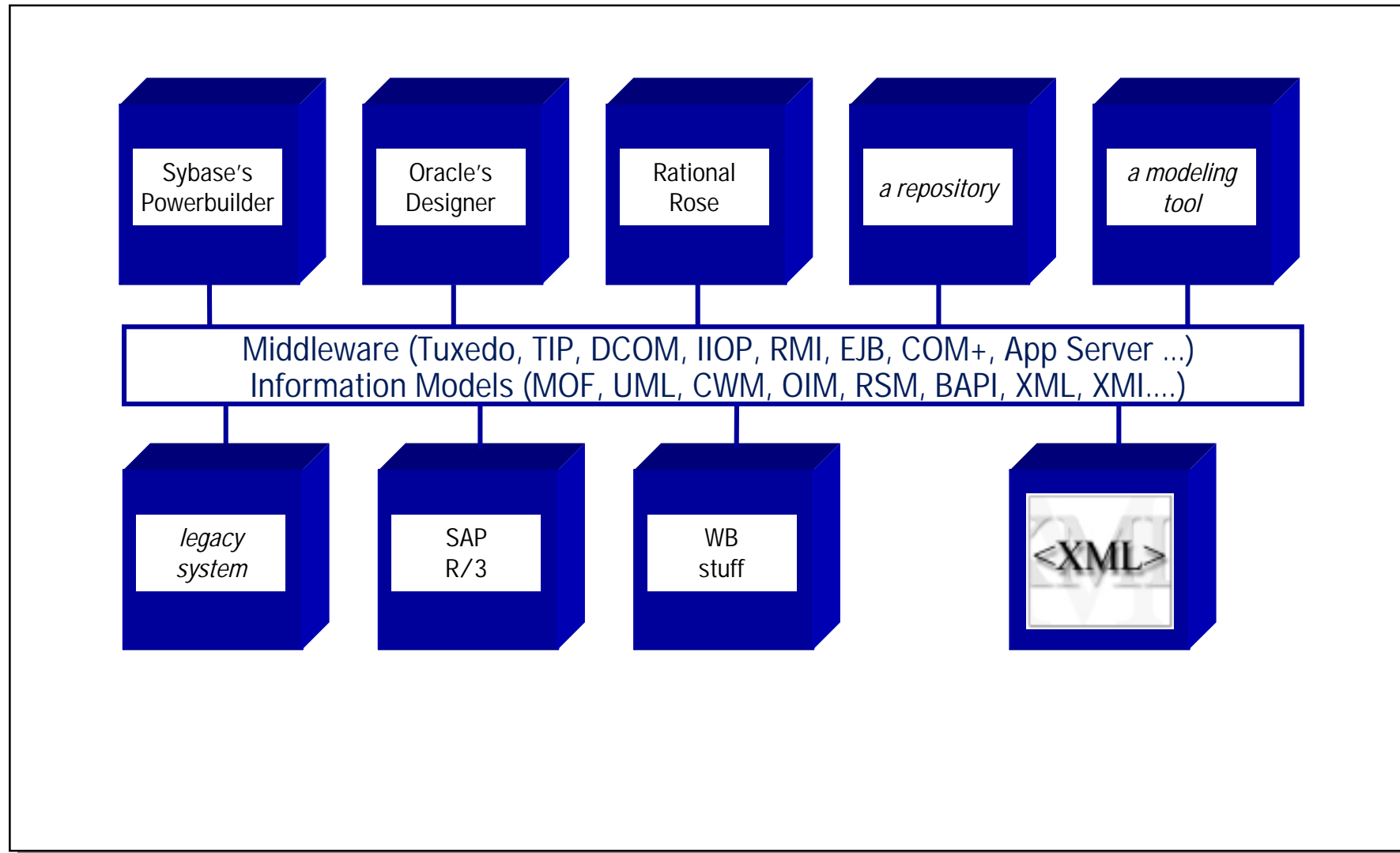
Metadaten:

[griech. meta *inmitten, zwischen, nach, hinter*]

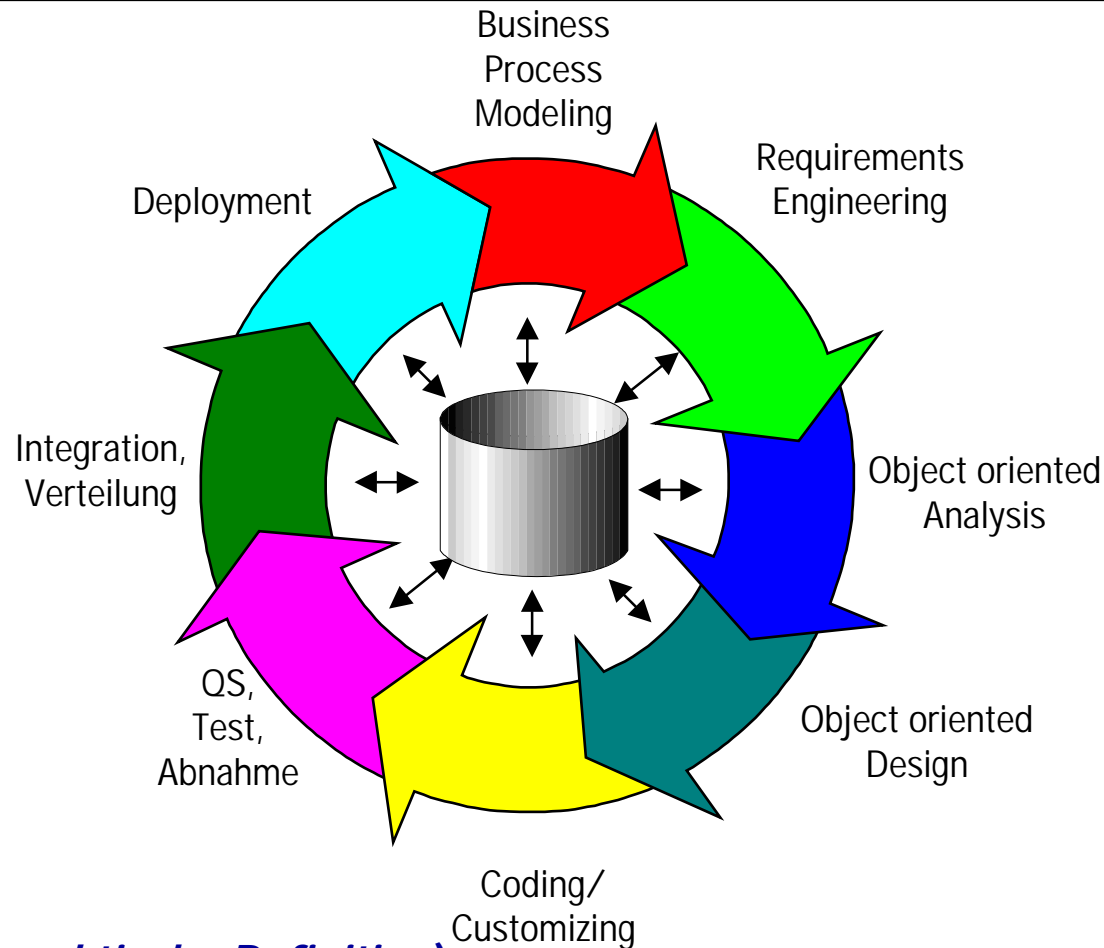
In der Informatik und verwandten Wissensgebieten als Wortbestandteile von *Metamodell, Metaklasse, Metaprogrammierung, Metasprache, Metaknowledge* verwandt.

Allgemein im Sinne einer zusätzlichen höheren Abstraktionsebene über der betrachteten Objektebene.

Muddleware Dilemma



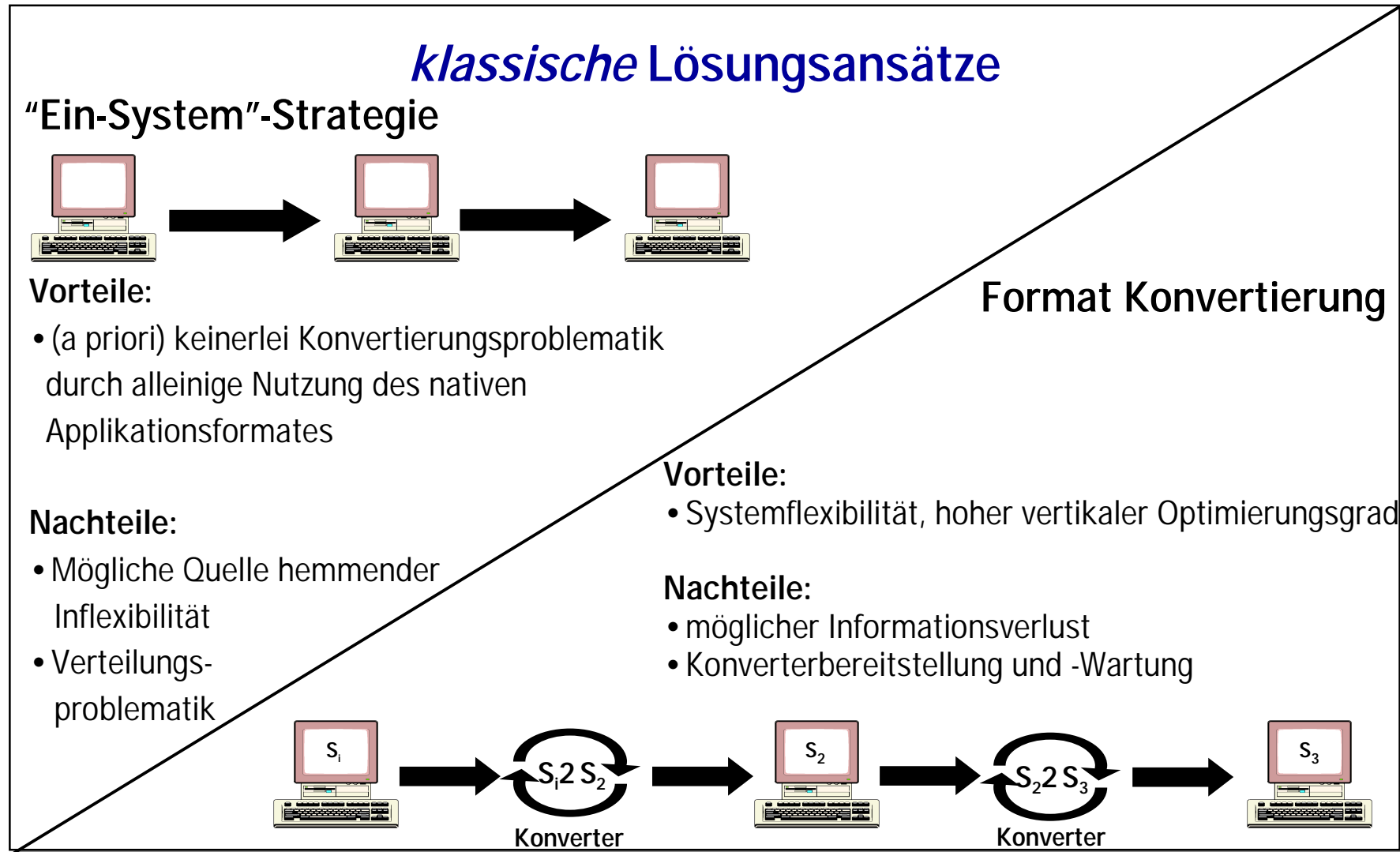
„klassischer“ Entwicklungszyklus



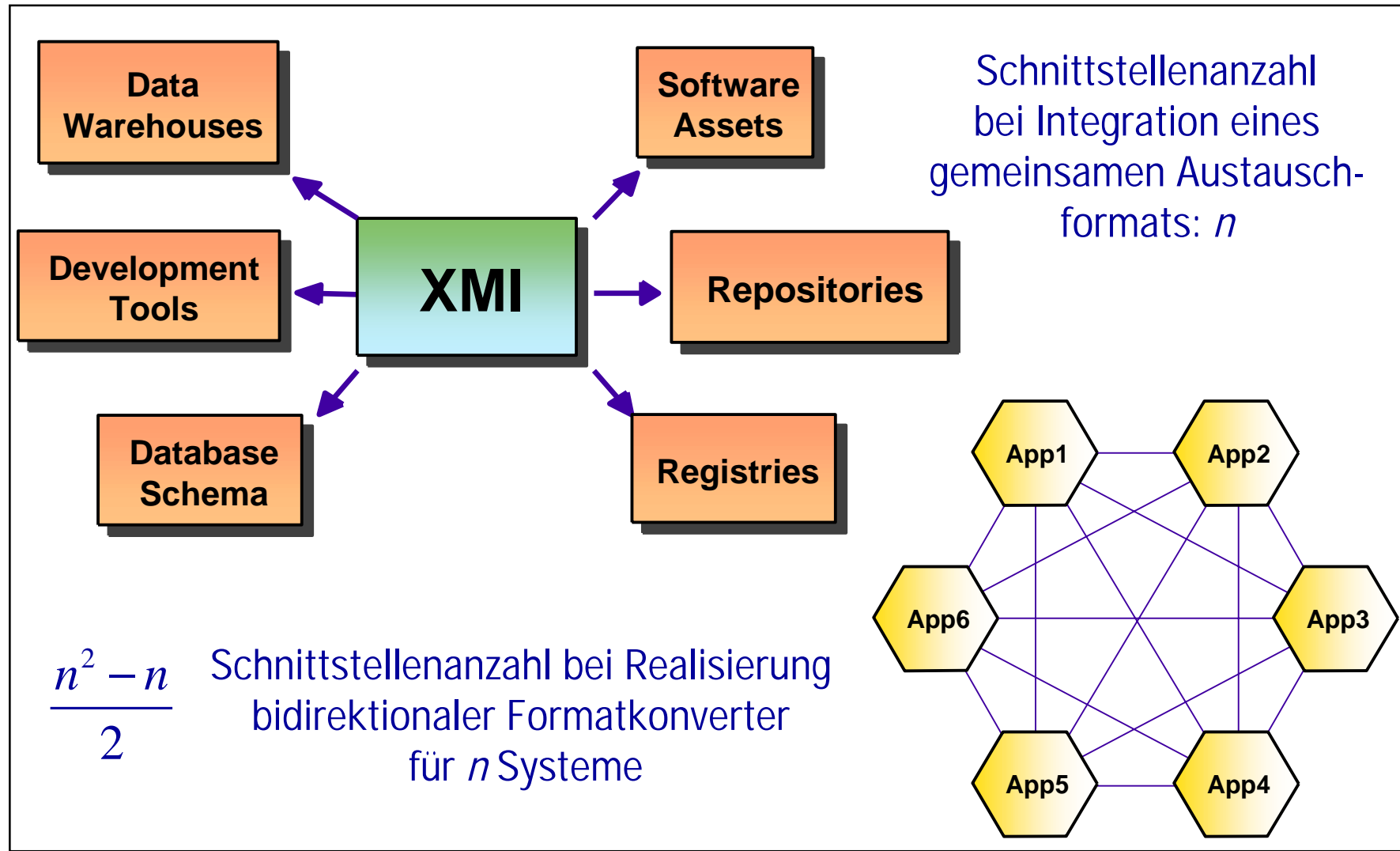
Metadaten (*praktische Definition*):

Alle prozessrelevanten Daten des Entwicklungs- und Einsatzlebenszyklus

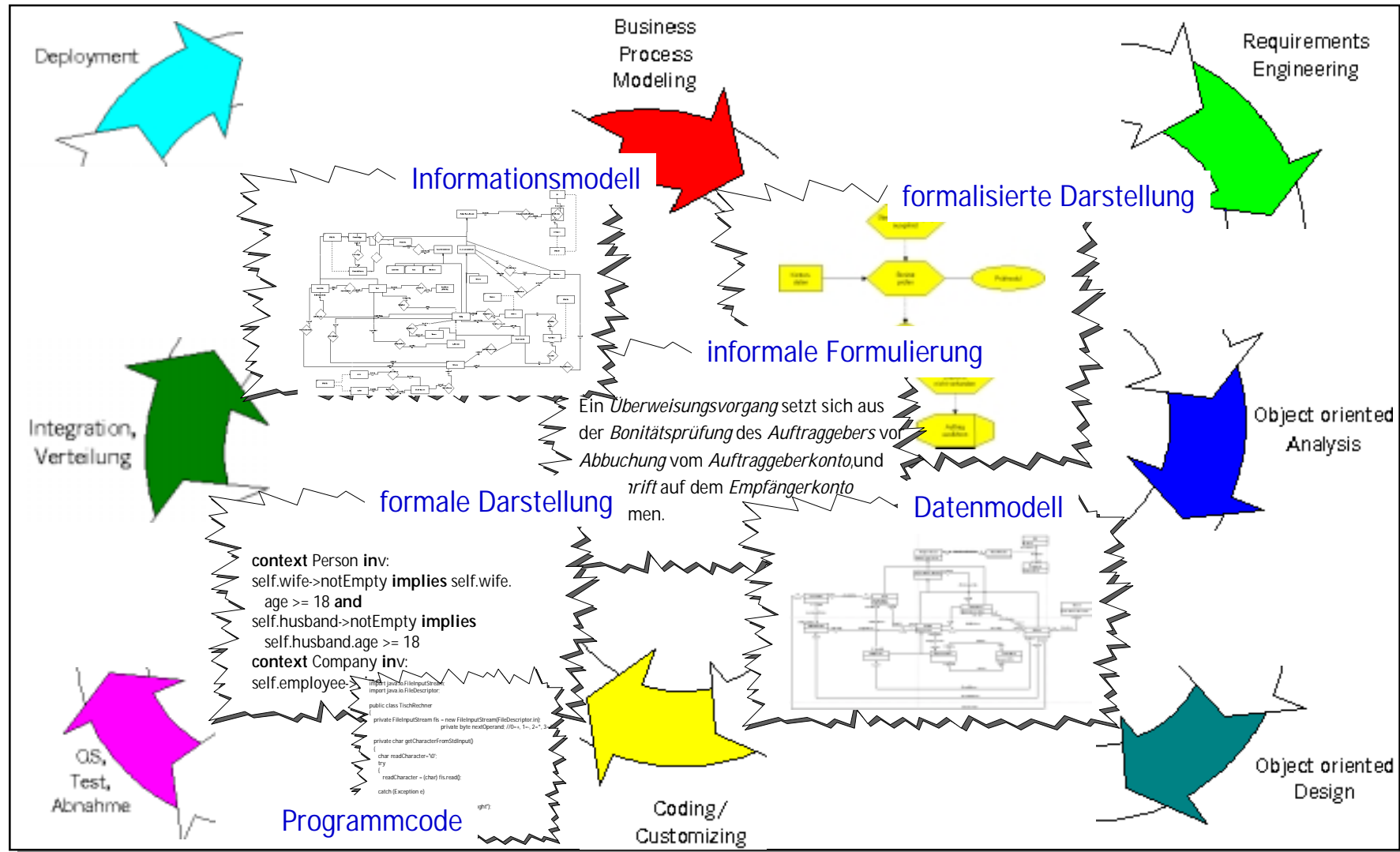
„klassische“ Lösungen der Schnittstellenproblematik



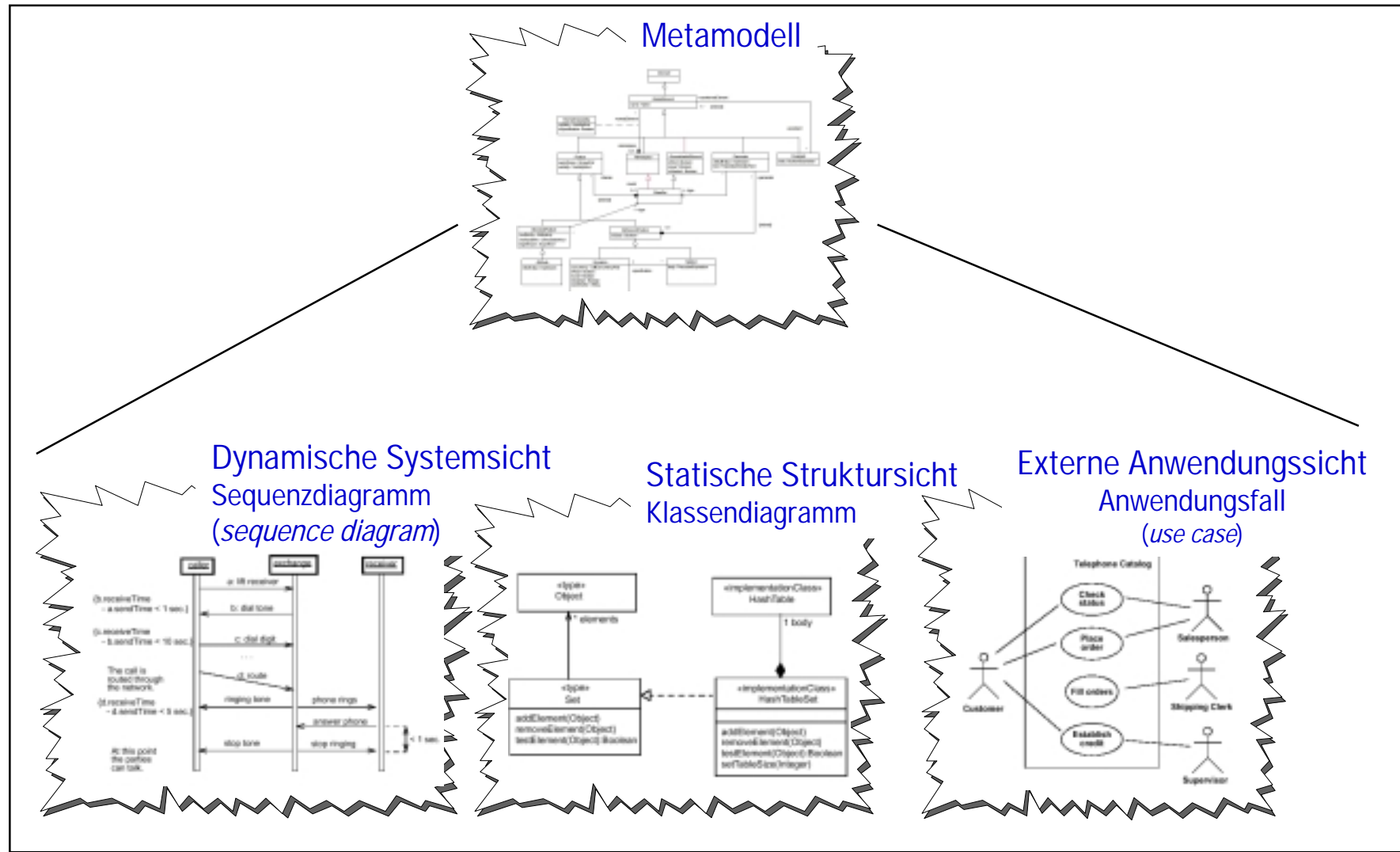
„klassische“ Schnittstellenproblematik



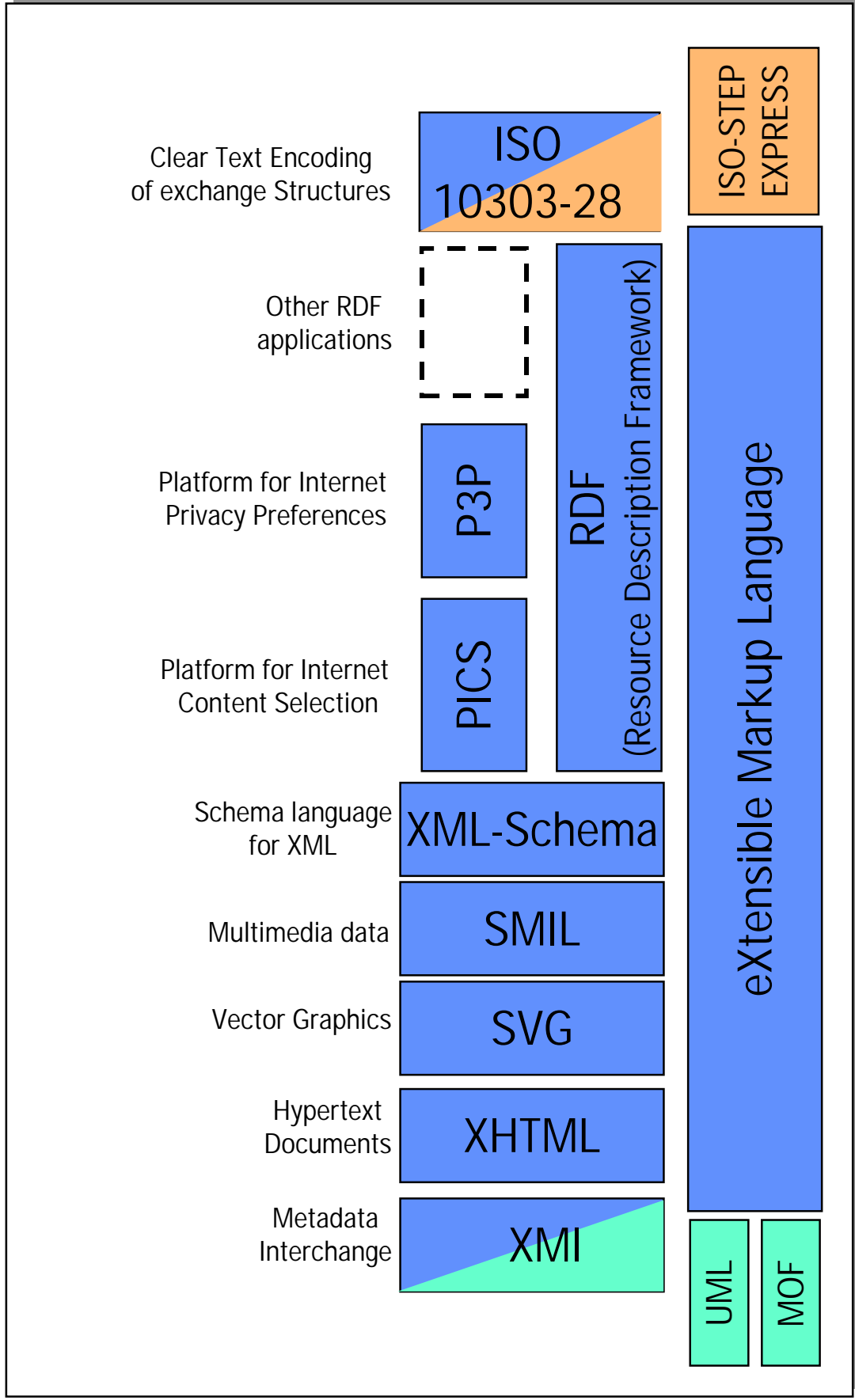
Metadatenquellen und ihre Formate



Metadaten und Metamodelle

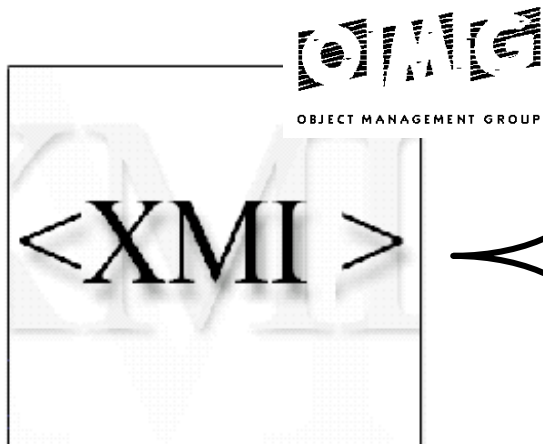


Einordnung von XML in die Familie der XML-Sprachen



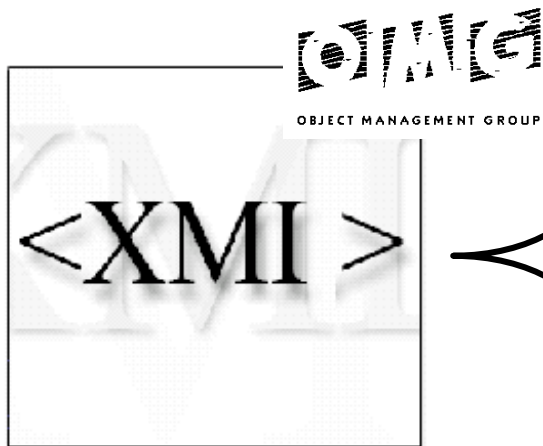
Anwendungsgebiete von XMI

Transfer von...



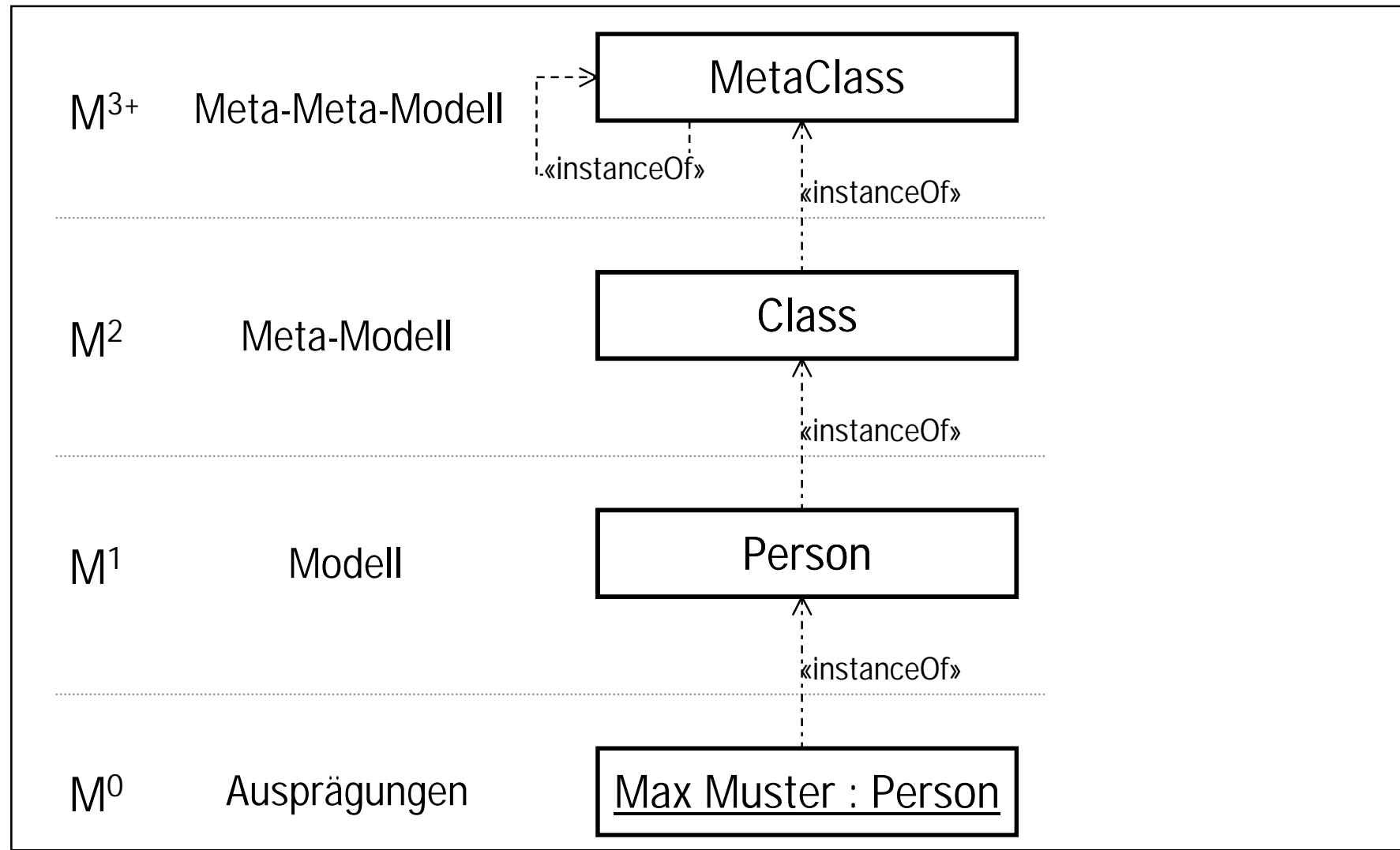
- Meta-Metamodell-Ausprägungen
(vollständige Modellierungssprachen)
- Metamodell-Ausprägungen
(vollständige Datenmodelle)
- Modell-Ausprägungen
(vollständige Datenmodell-Ausprägungen/
Instanzen)

Aufbau von XMI

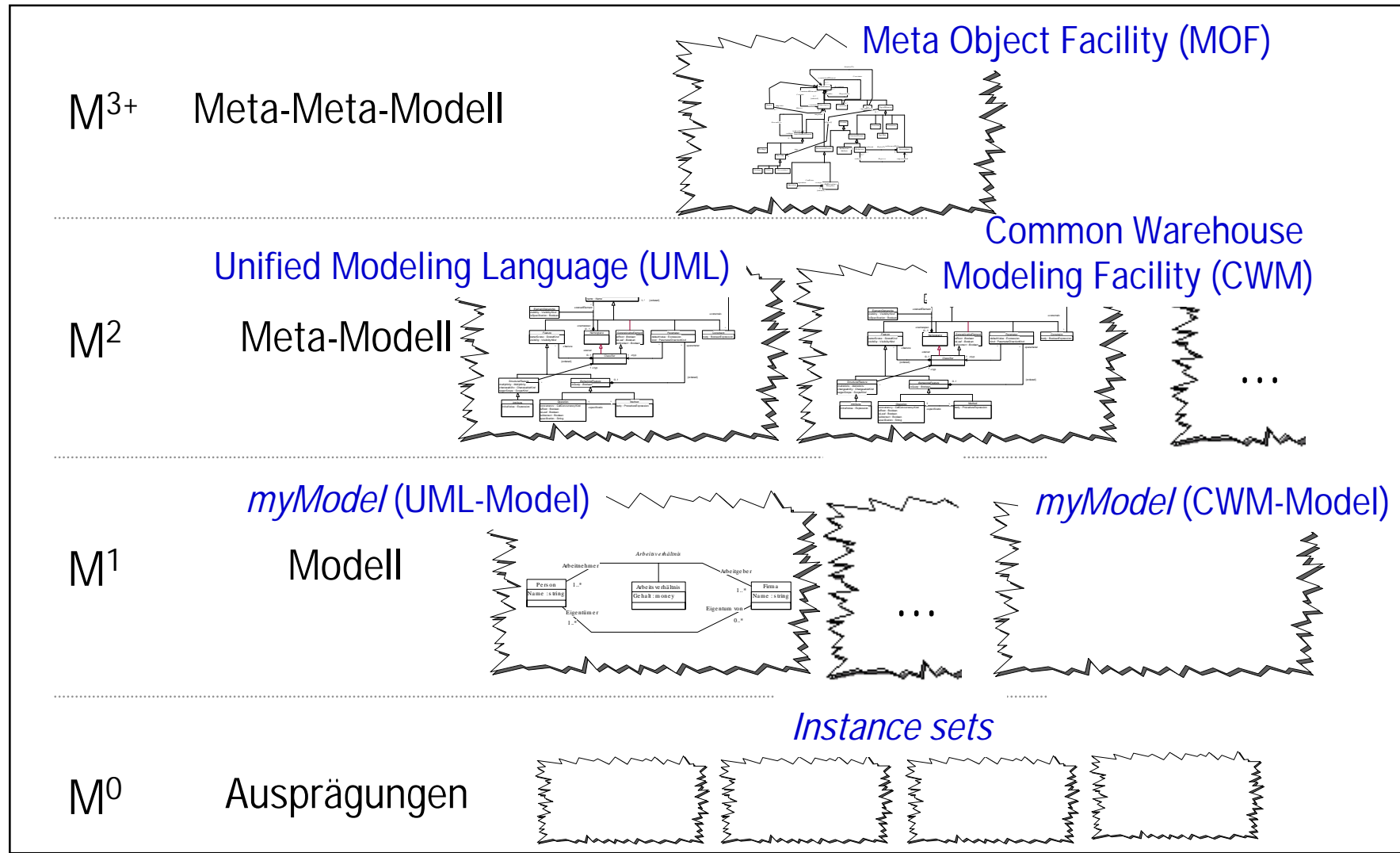


- XML-DTD für UML-Modelle
XMI[UML]
- XML-DTD für MOF-Modelle
XMI[MOF]
- *Generation principles:*
Beschreibung der Herleitung von
XML-DTDs aus UML-basierten
Modellen

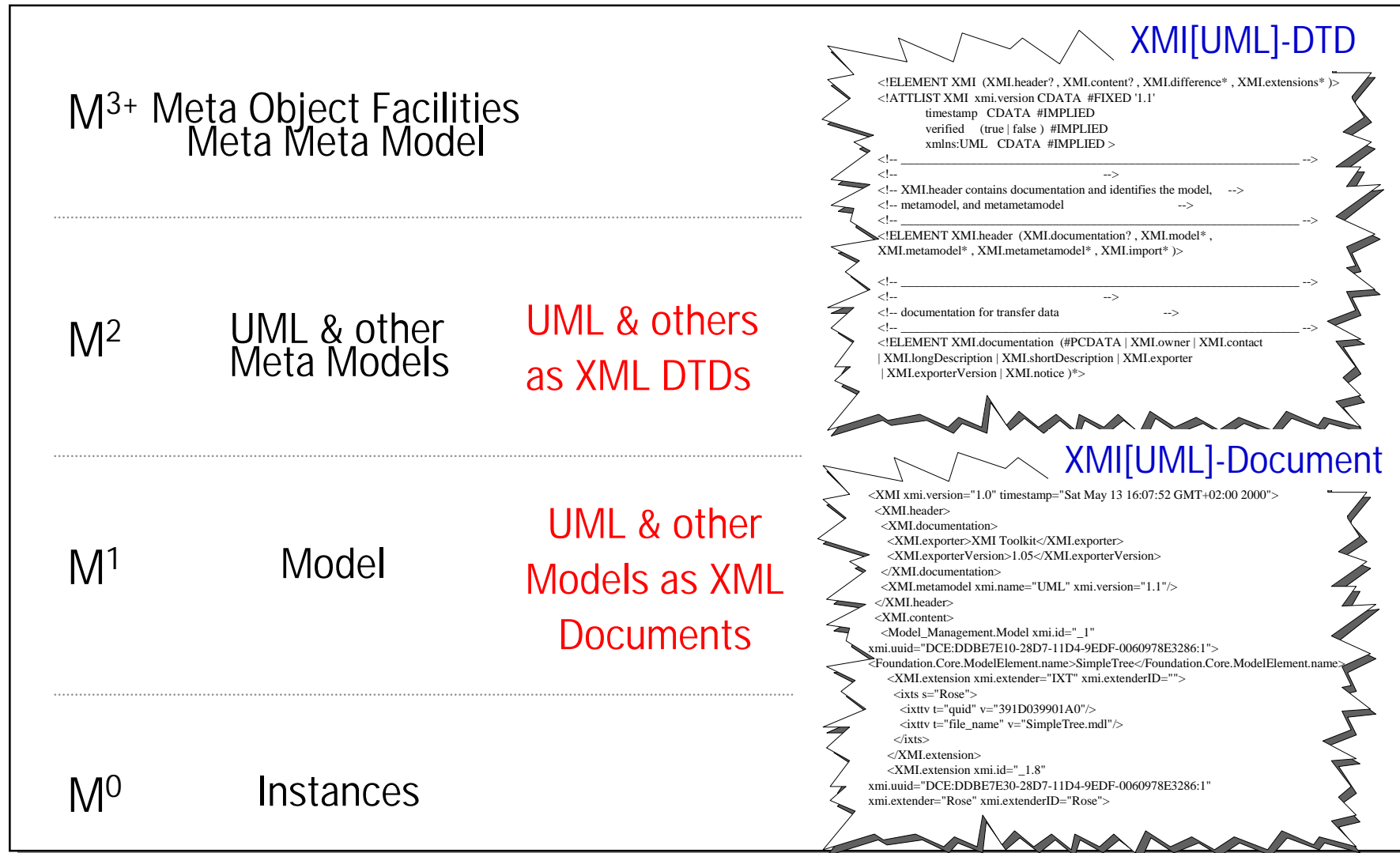
Vier Schichten Metamodellarchitektur der OMG



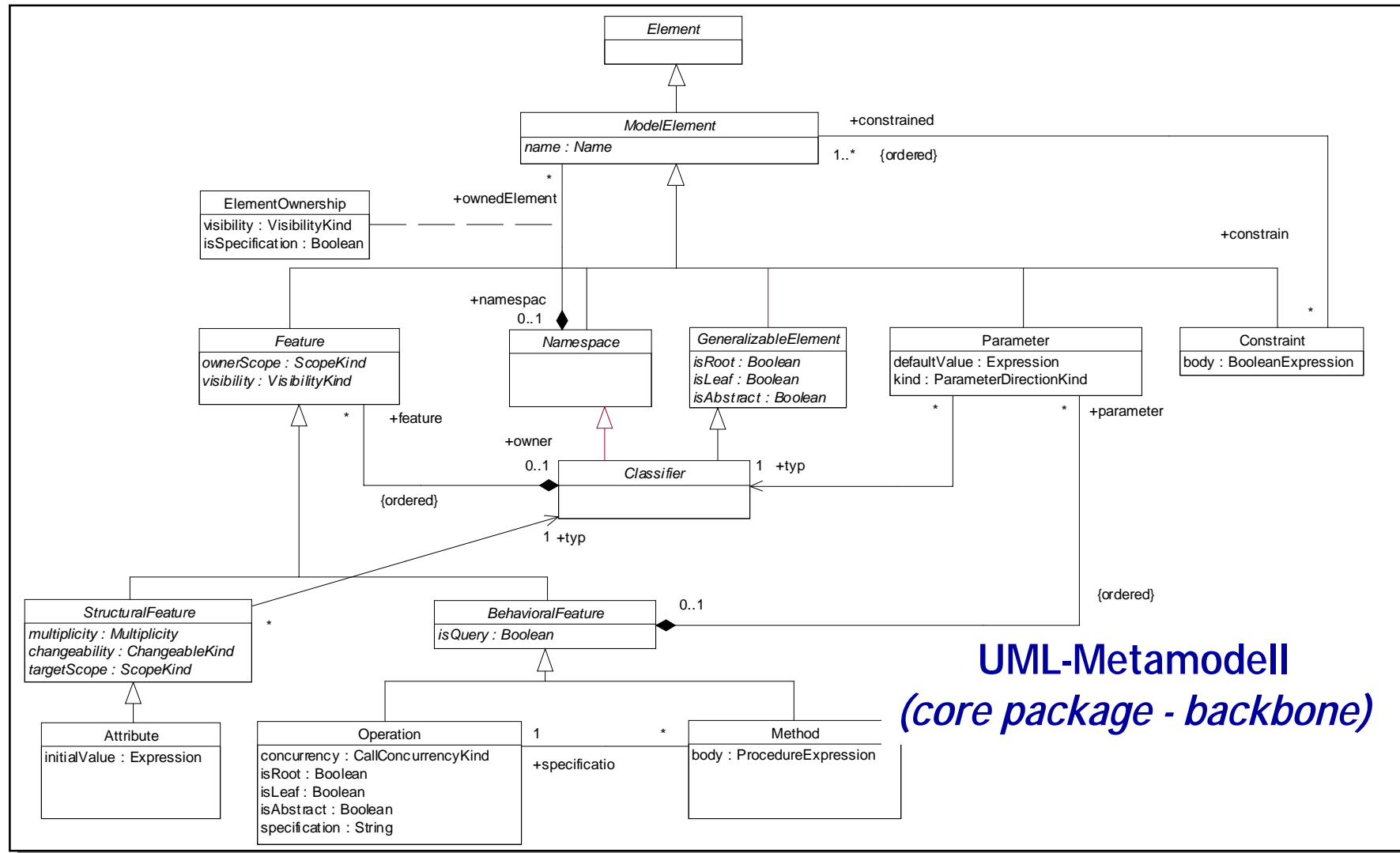
Vier Schichten Metamodellarchitektur der OMG



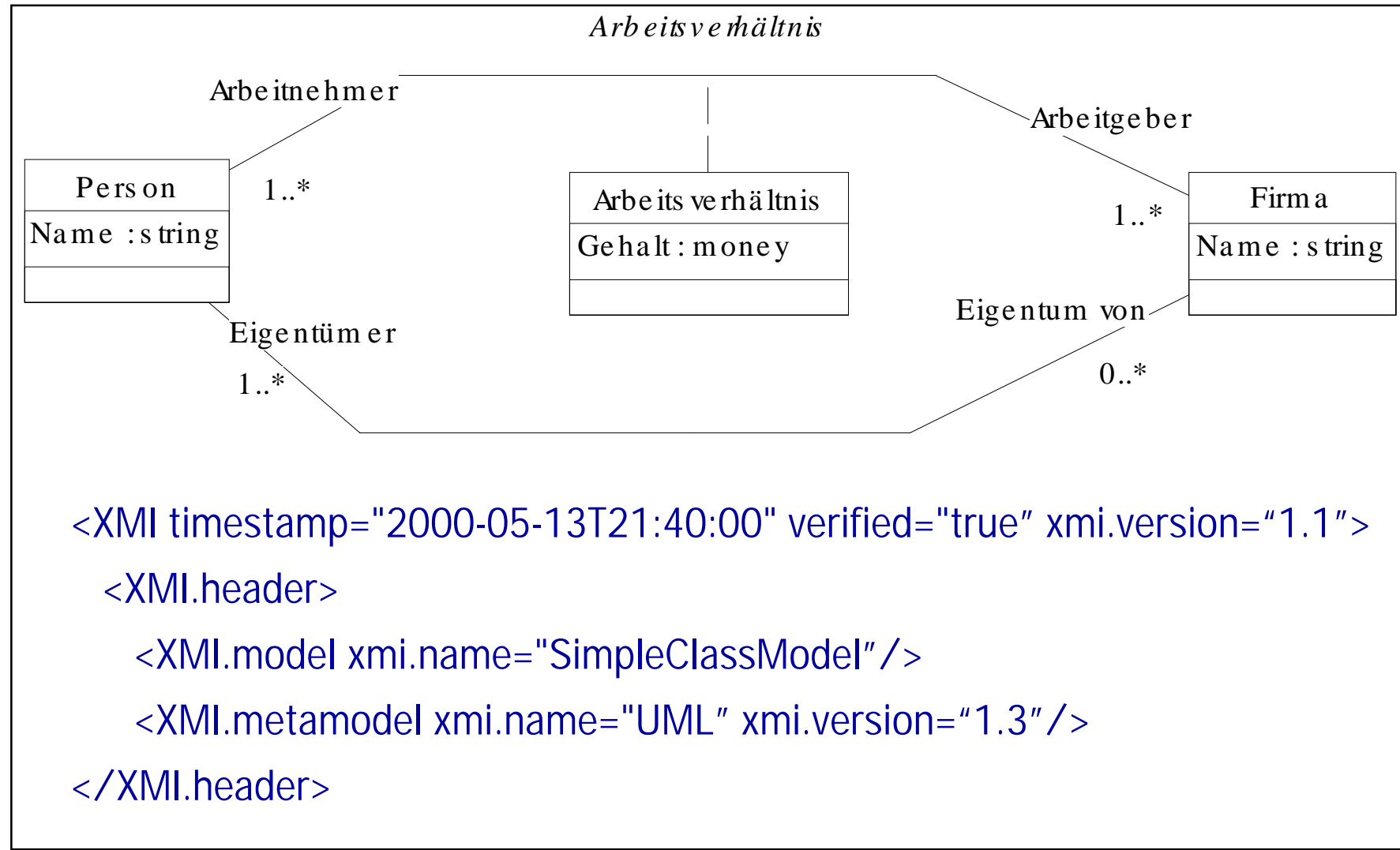
Anwendungsgebiete von XMI -- Transfer von UML-Modellen



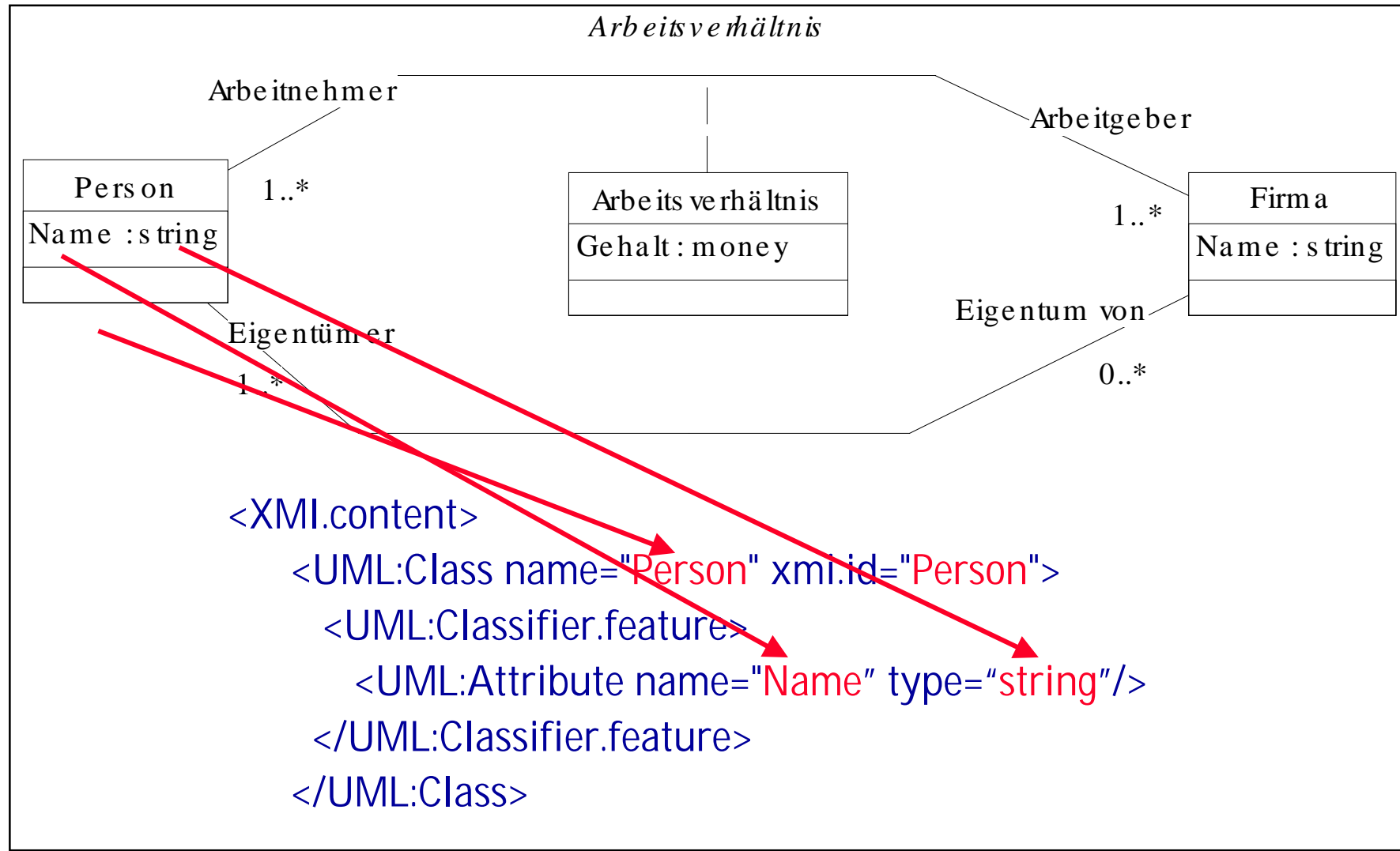
Anwendungsgebiete von XMI -- Transfer von UML-Modellen



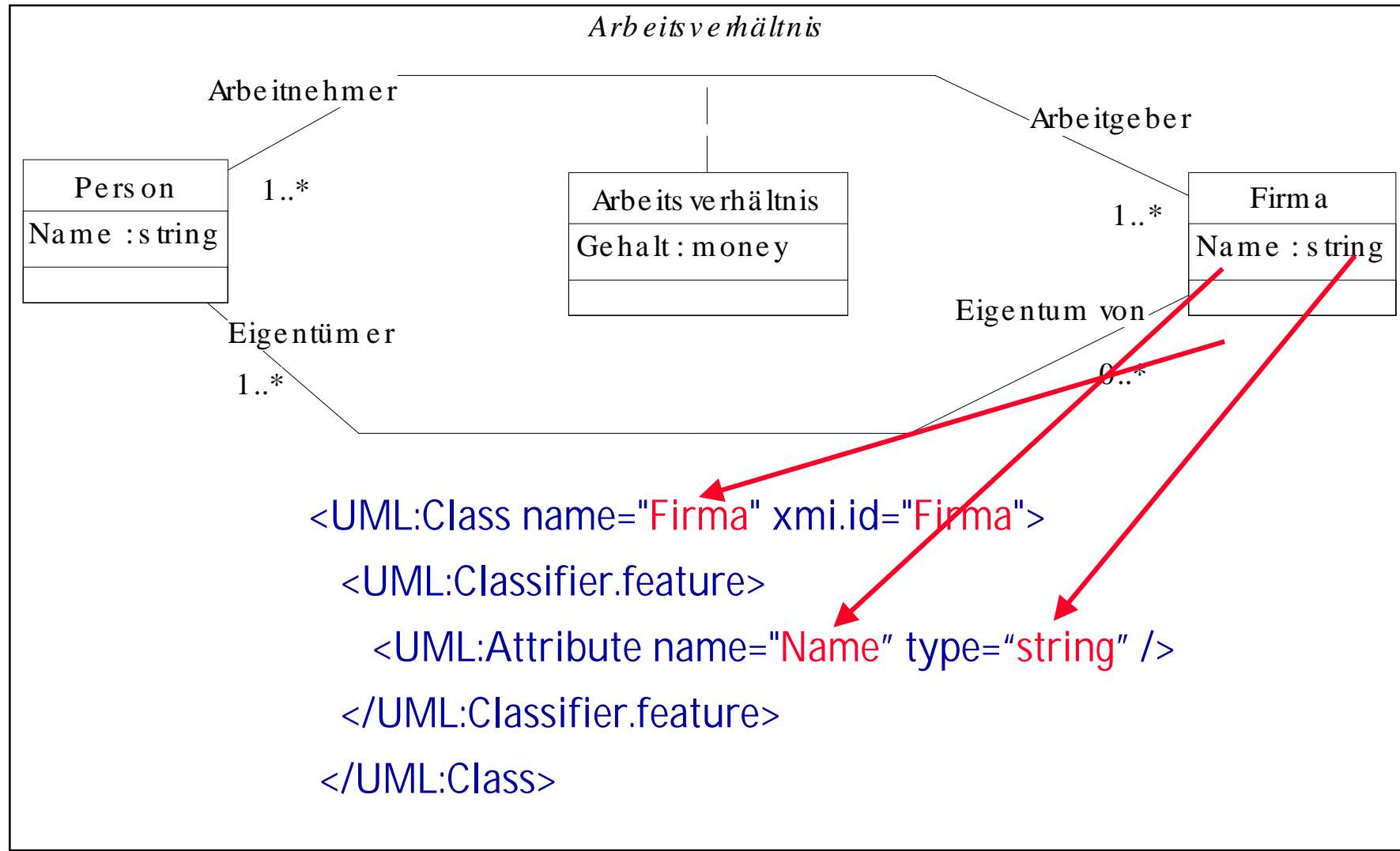
Anwendungsgebiete von XMI -- Transfer von UML-Modellen



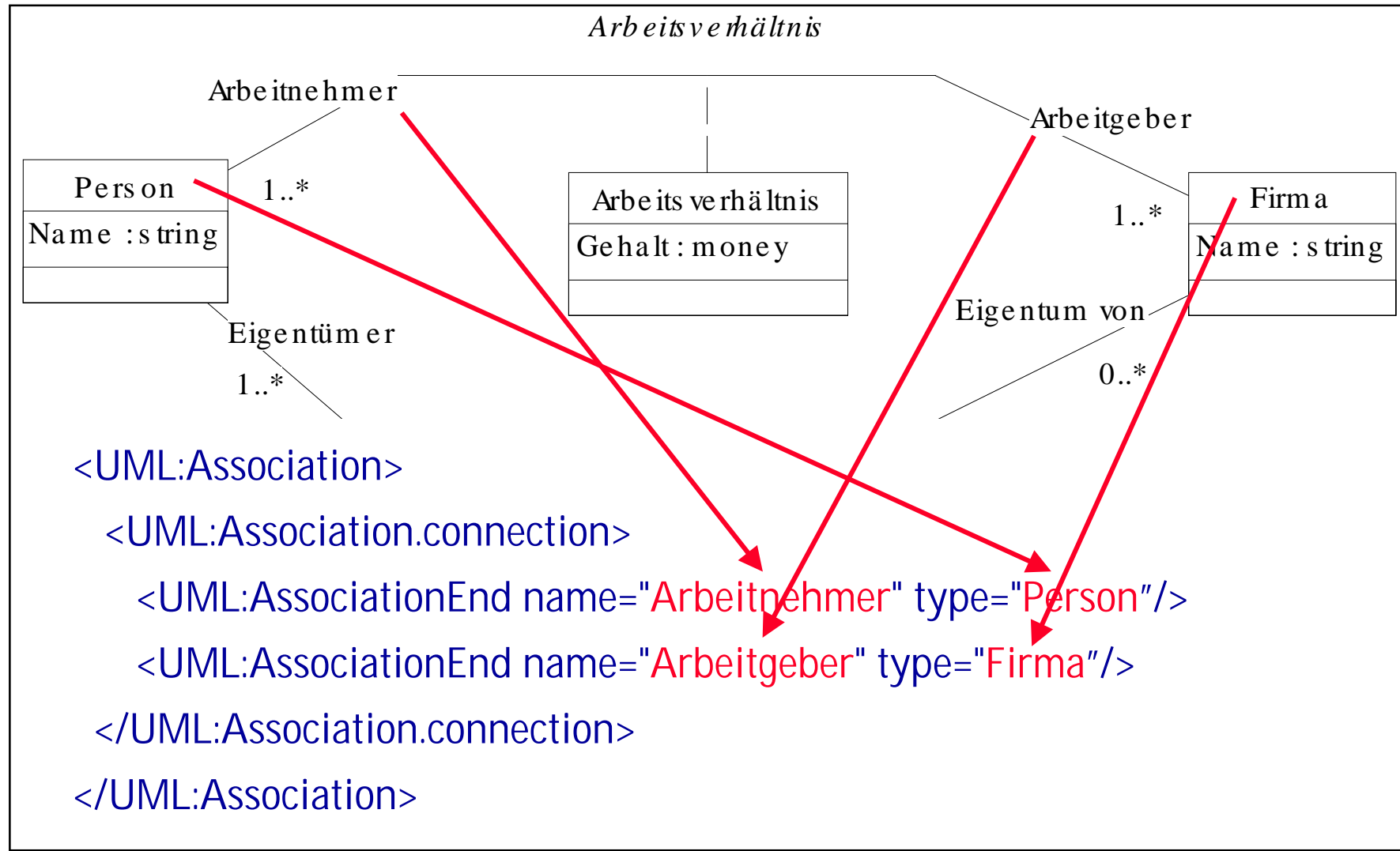
Anwendungsgebiete von XMI -- Transfer von UML-Modellen



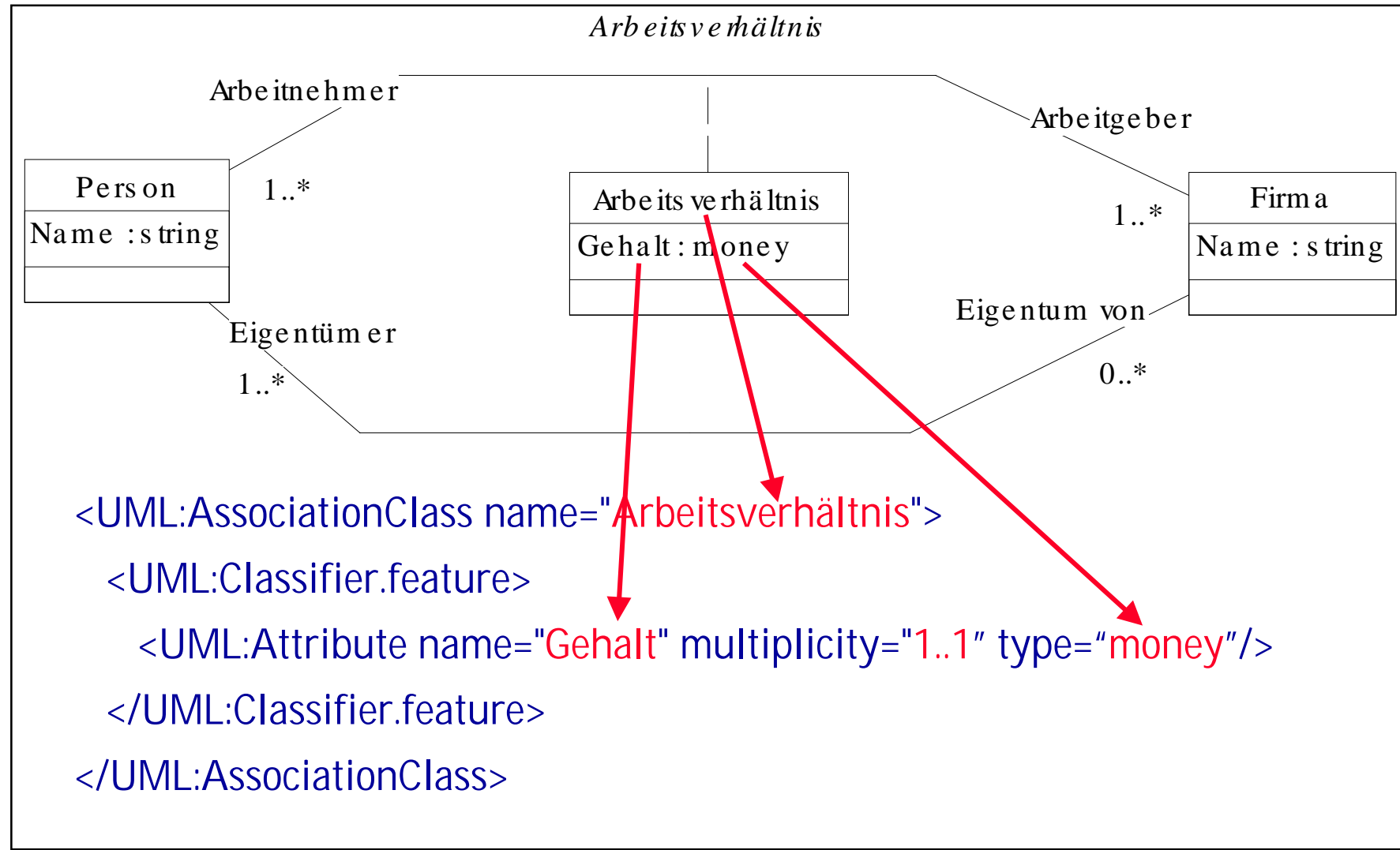
Anwendungsgebiete von XMI -- Transfer von UML-Modellen



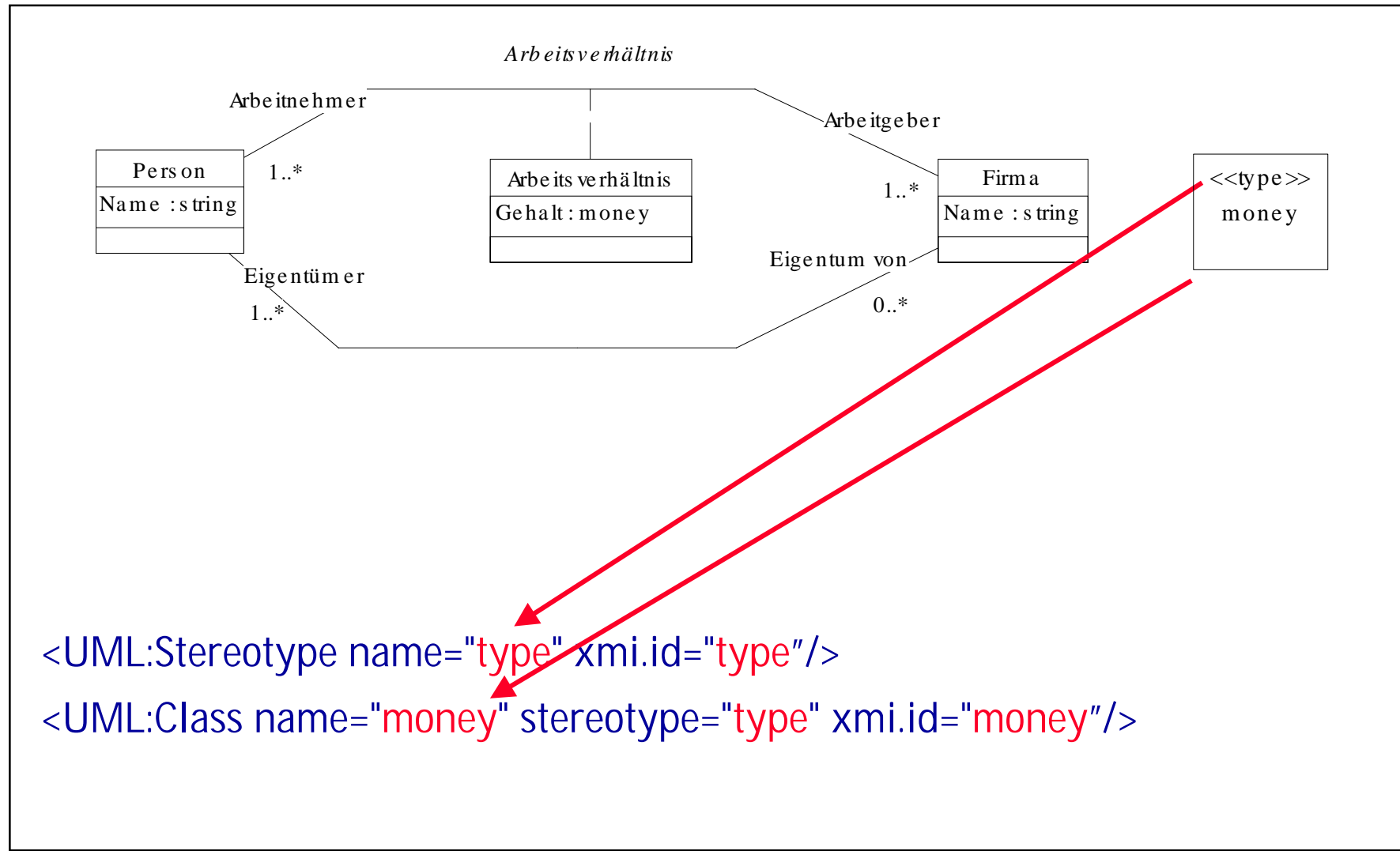
Anwendungsgebiete von XMI -- Transfer von UML-Modellen



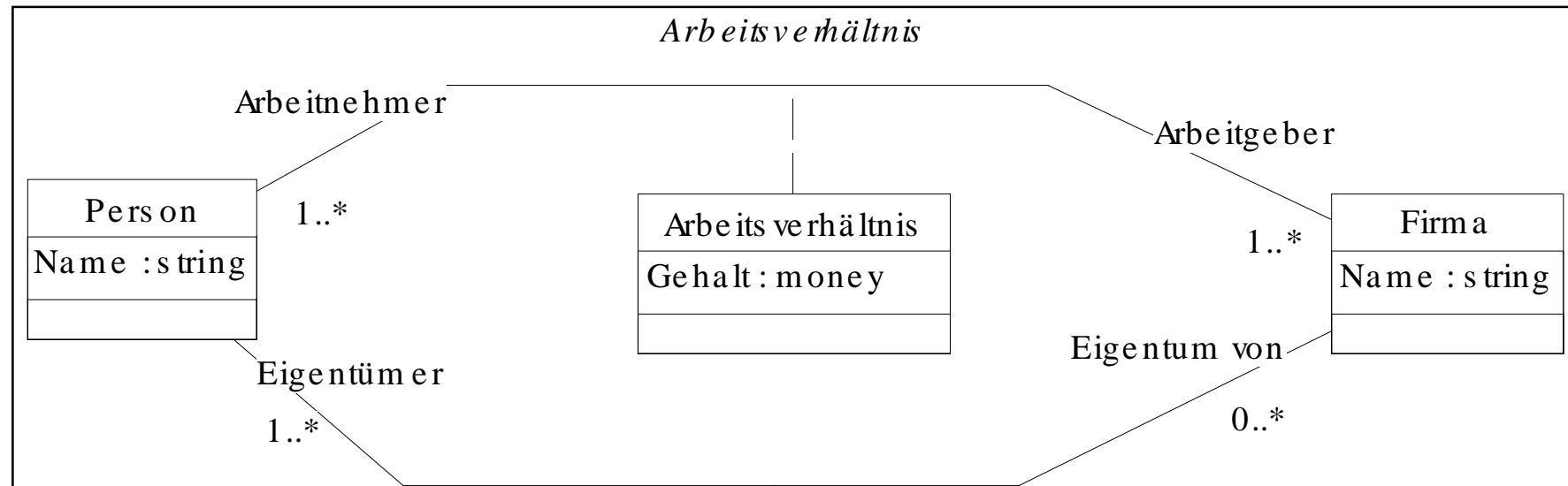
Anwendungsgebiete von XMI -- Transfer von UML-Modellen



Anwendungsgebiete von XMI -- Transfer von UML-Modellen



Anwendungsgebiete von XMI -- Transfer von UML-Modellen



<UML:Association>

<UML:Association.connection>

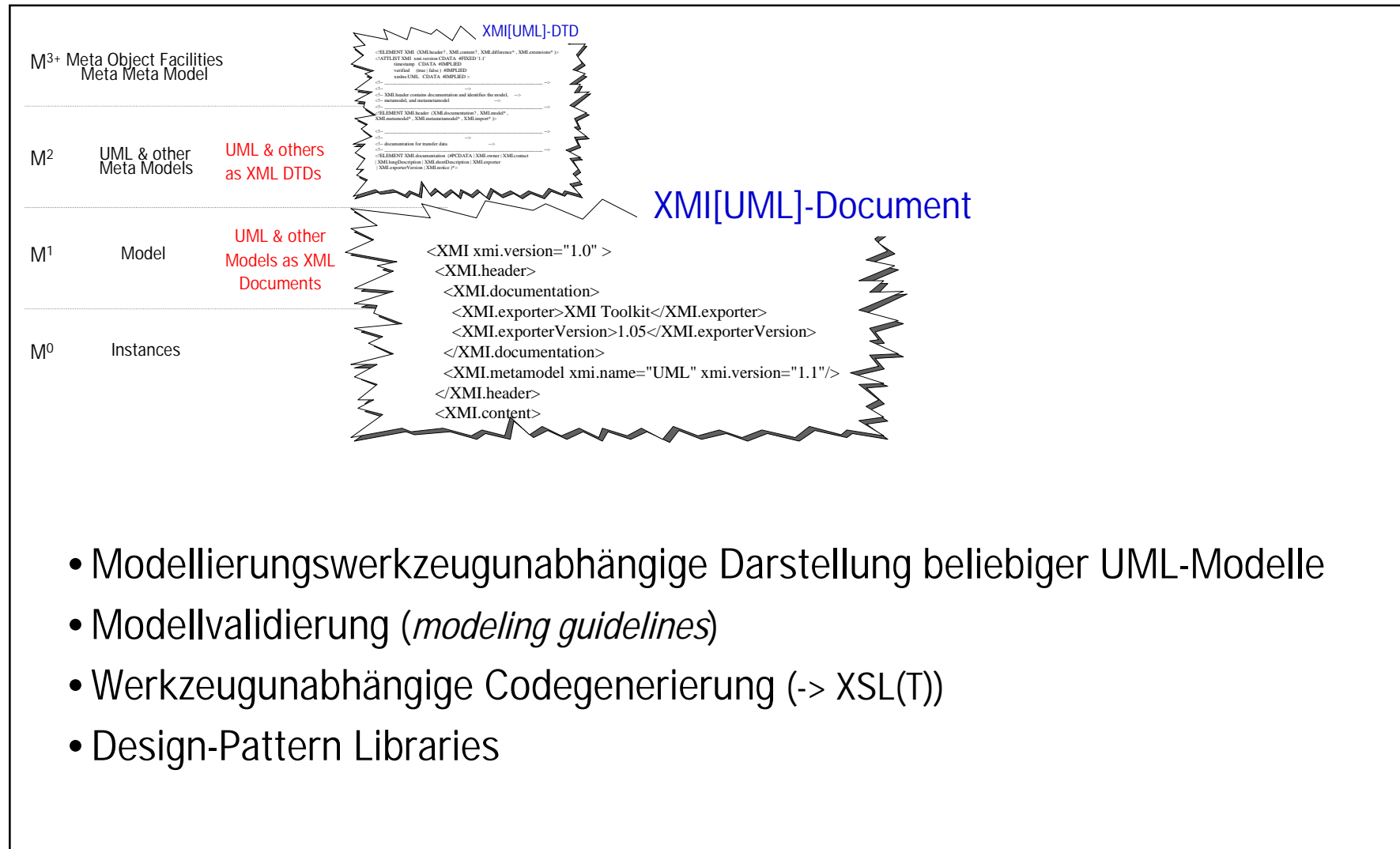
<UML:AssociationEnd name="Eigentümer" multiplicity="1..*" type="Person"/>

<UML:AssociationEnd name="Eigentum_von" multiplicity="1..*" type="Firma"/>

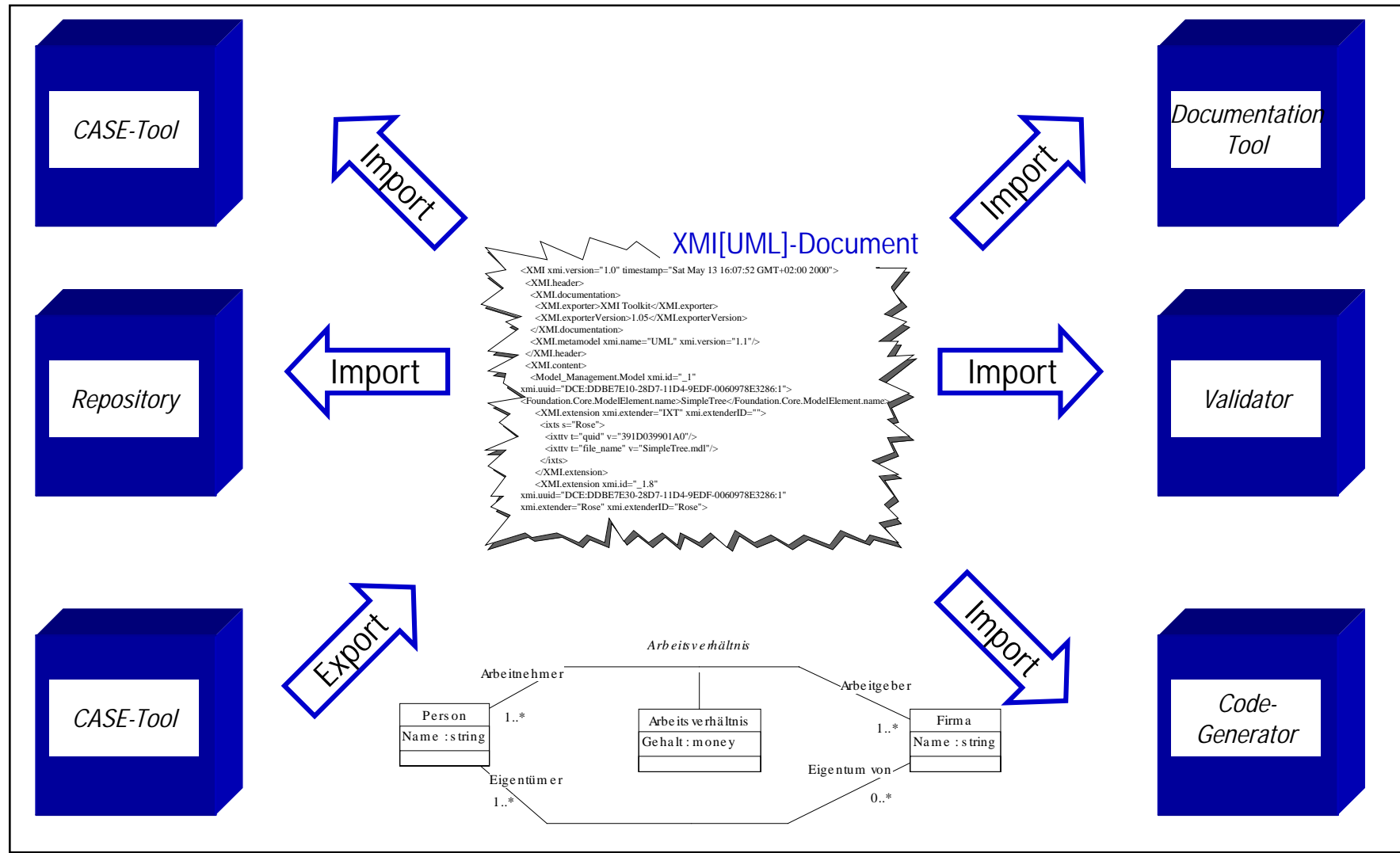
</UML:Association.connection>

</UML:Association>

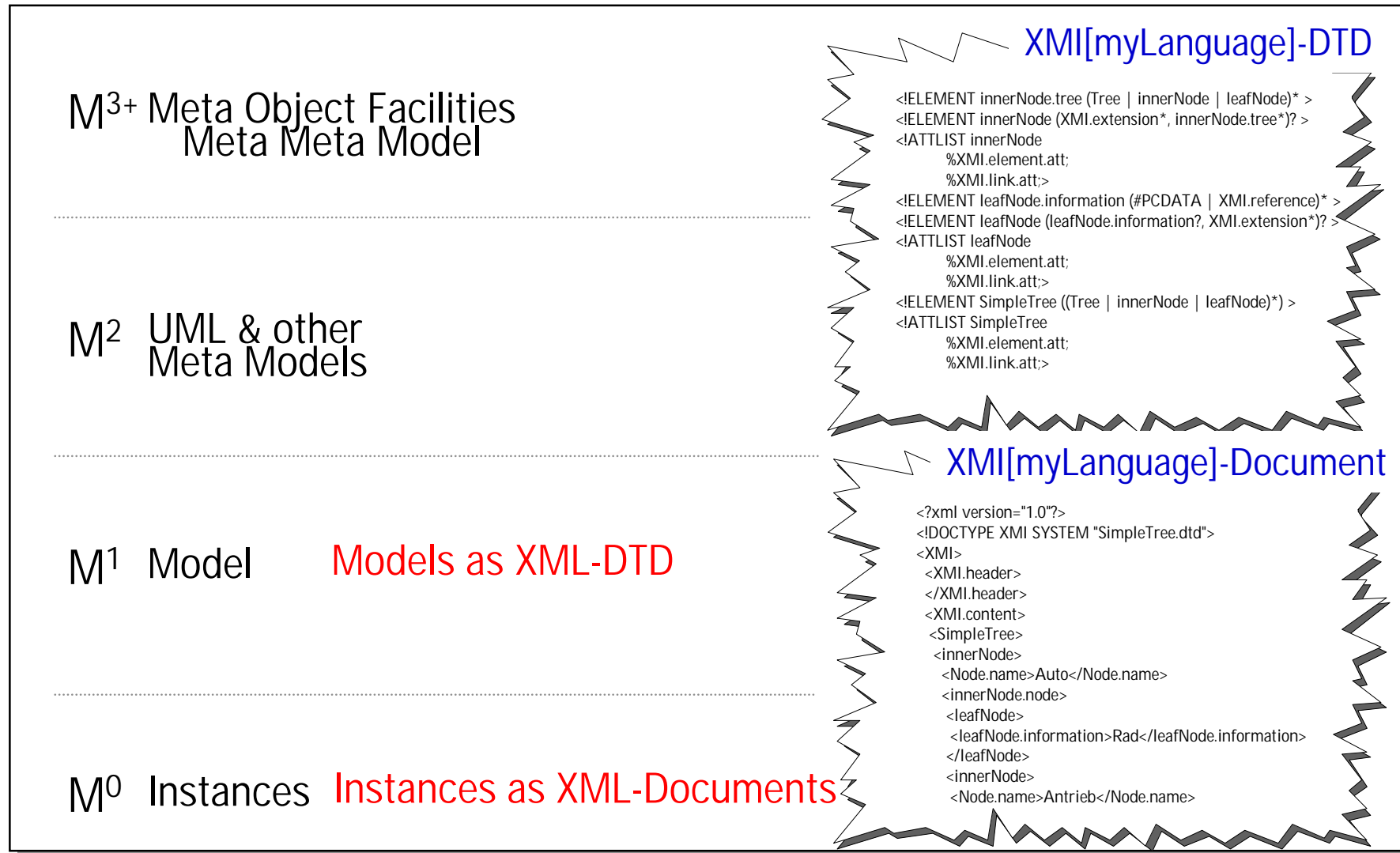
Anwendungsgebiete von XMI -- Transfer von UML-Modellen



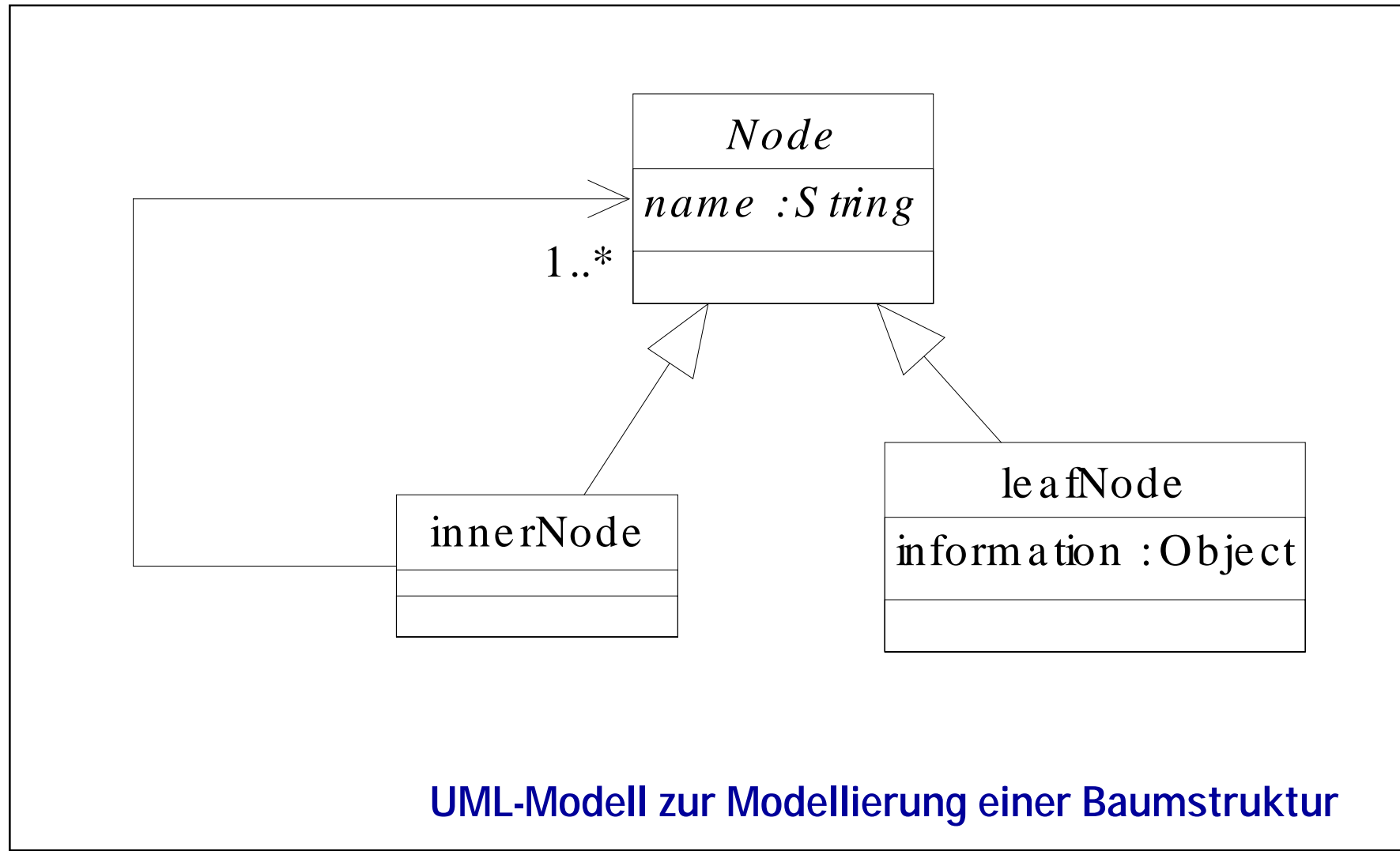
Anwendungsgebiete von XMI -- Transfer von UML-Modellen



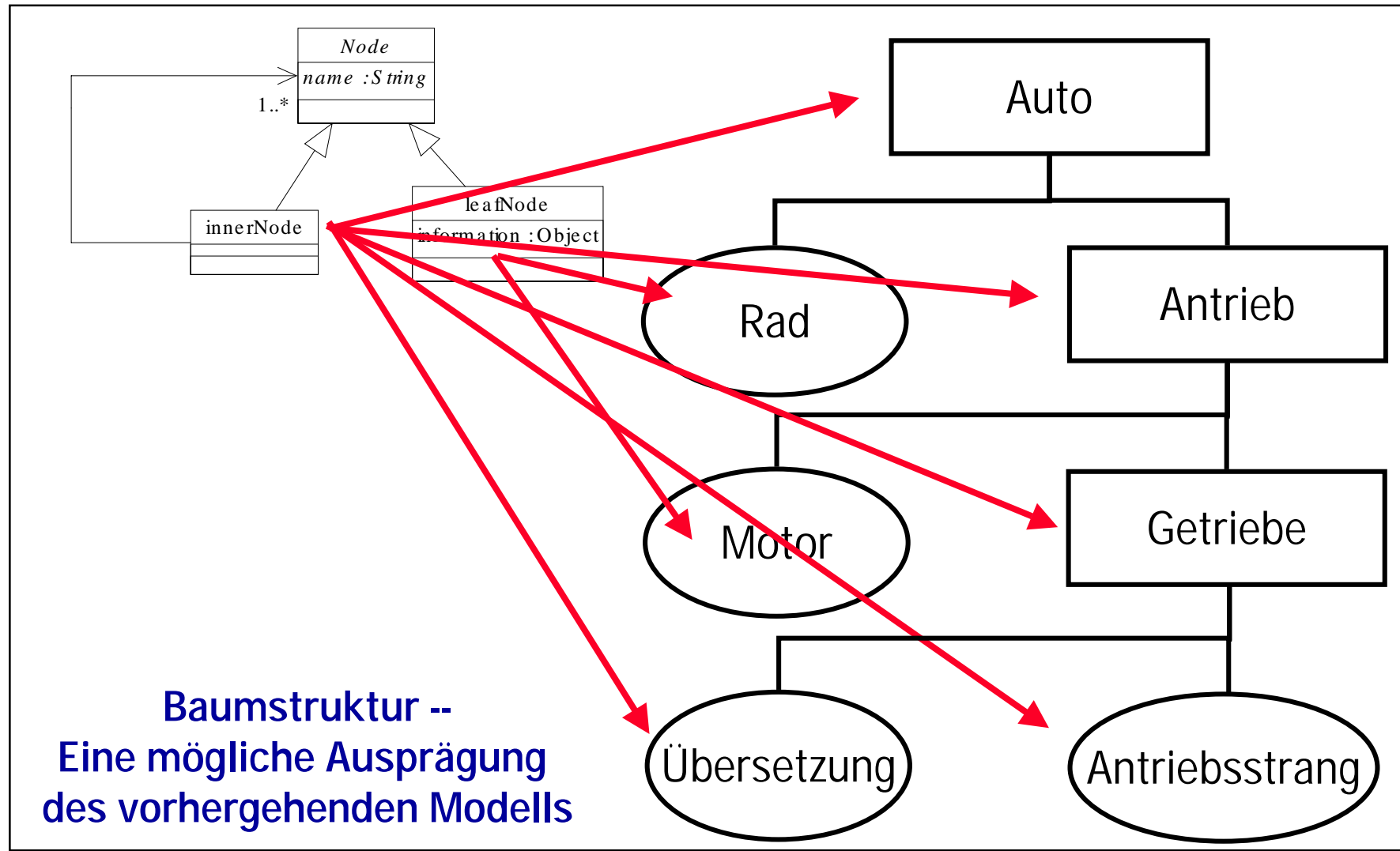
Anwendungsgebiete von XMI -- Transfer von konkreten Daten



Anwendungsgebiete von XMI -- Transfer von konkreten Daten



Anwendungsgebiete von XMI -- Transfer von konkreten Daten

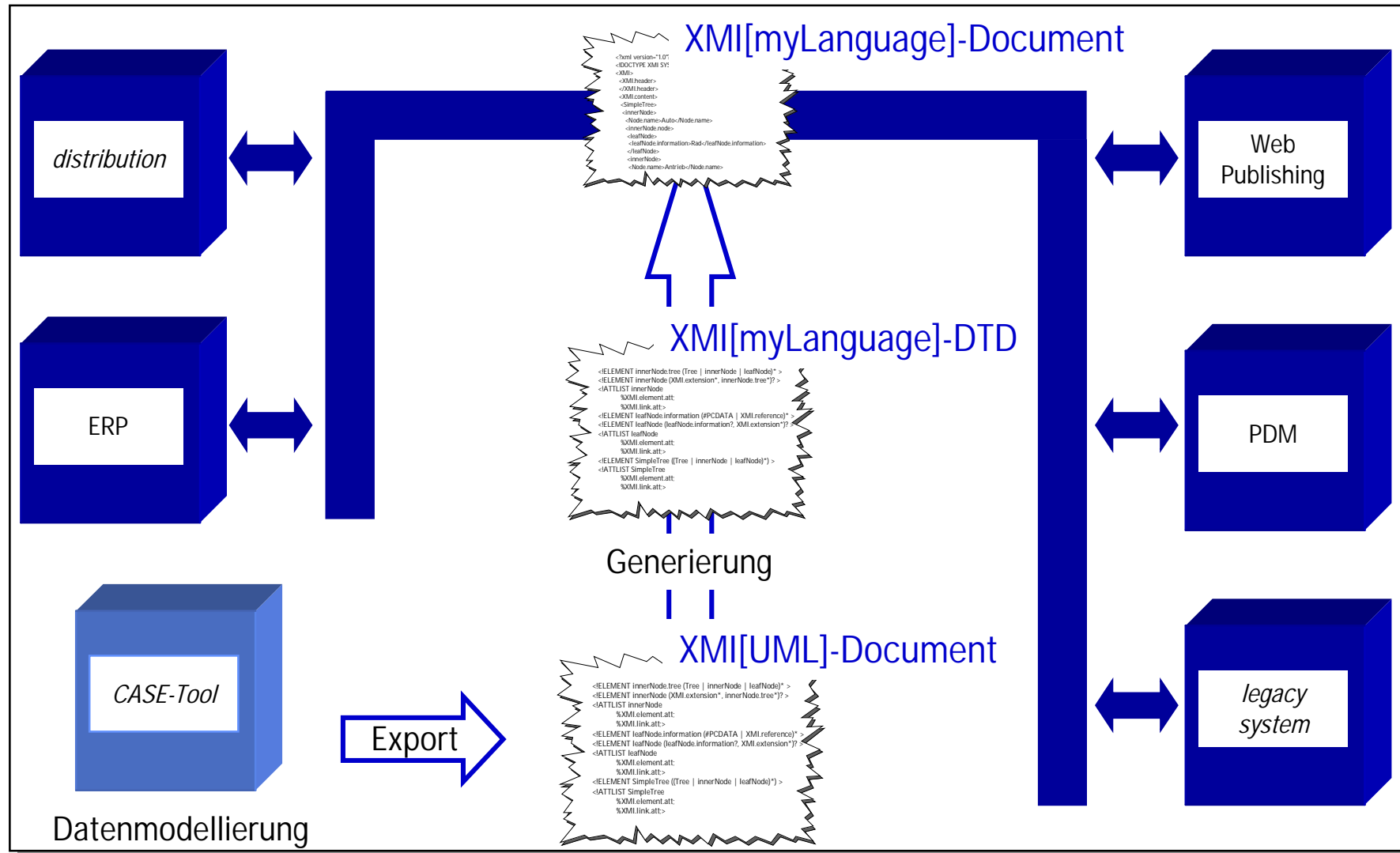


Anwendungsgebiete von XMI -- Transfer von konkreten Daten

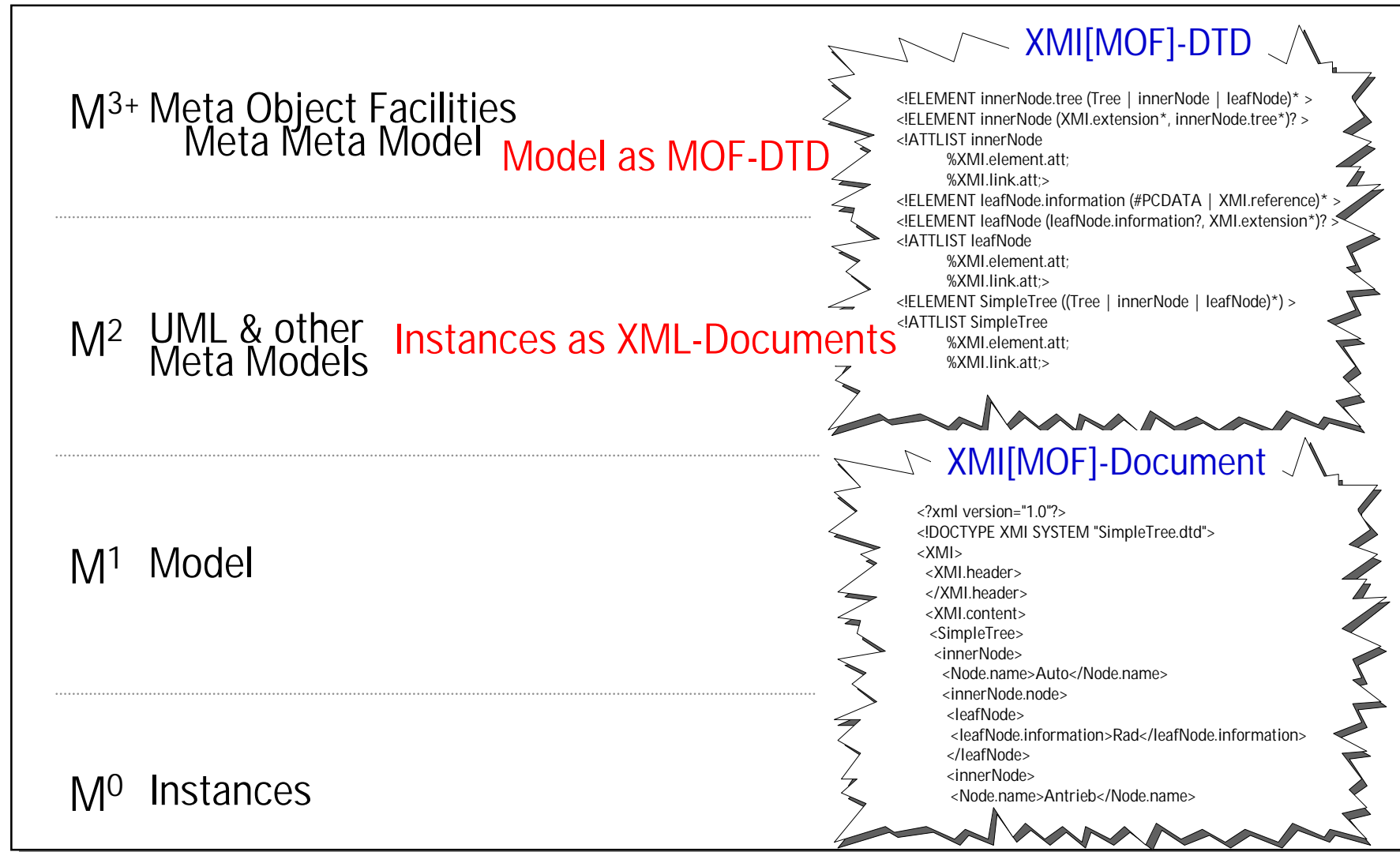
M ³⁺ Meta Object Facilities Meta Meta Model	<p>XMI[myLanguage]-DTD</p> <pre> <ELEMENT innerNode.tree (Tree innerNode leafNode)* > <ELEMENT innerNode (XMI.extension*, innerNode.tree)* > <ATTLIST innerNode %XMI.element.att; %XMI.link.att; <ELEMENT leafNode.information (#PCDATA XMI.reference)* > <ELEMENT leafNode (leafNode.information?, XMI.extension?) > <ATTLIST leafNode %XMI.element.att; %XMI.link.att; <ELEMENT SimpleTree ((Tree innerNode leafNode)* > <ATTLIST SimpleTree %XMI.element.att; %XMI.link.att; </pre>	
M ² UML & other Meta Models		
M ¹ Model Models as XML-DTD	<p>XMI[myLanguage]-Document</p> <pre> <?xml version="1.0"?> <!DOCTYPE XMI SYSTEM "SimpleTree.dtd"> <XMI> <XMI.header> </XMI.header> <XMI.content> <SimpleTree> <innerNode> <Node.name>Auto</Node.name> <innerNode.node> <leafNode> <leafNode.information>Rad </leafNode.information> </leafNode> </pre>	
M ⁰ Instances Instances as XML-Documents		

- Automatisierte Gewinnung von DTDs aus beliebigen Modellen
- Integration von XML-Sprachen in die Prozeßkette mit vergleichsweise geringem Aufwand

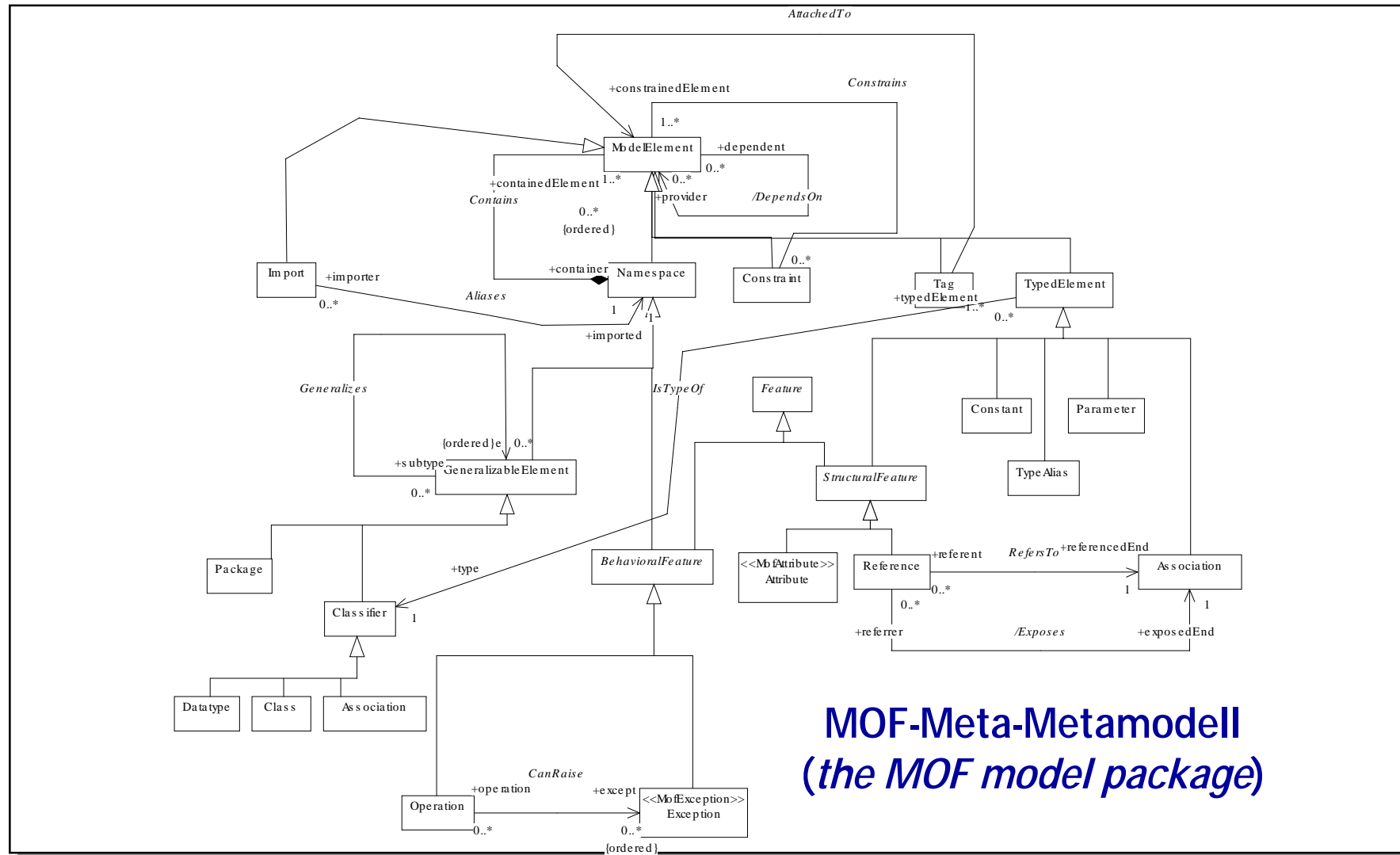
Anwendungsgebiete von XMI -- Transfer von konkreten Daten



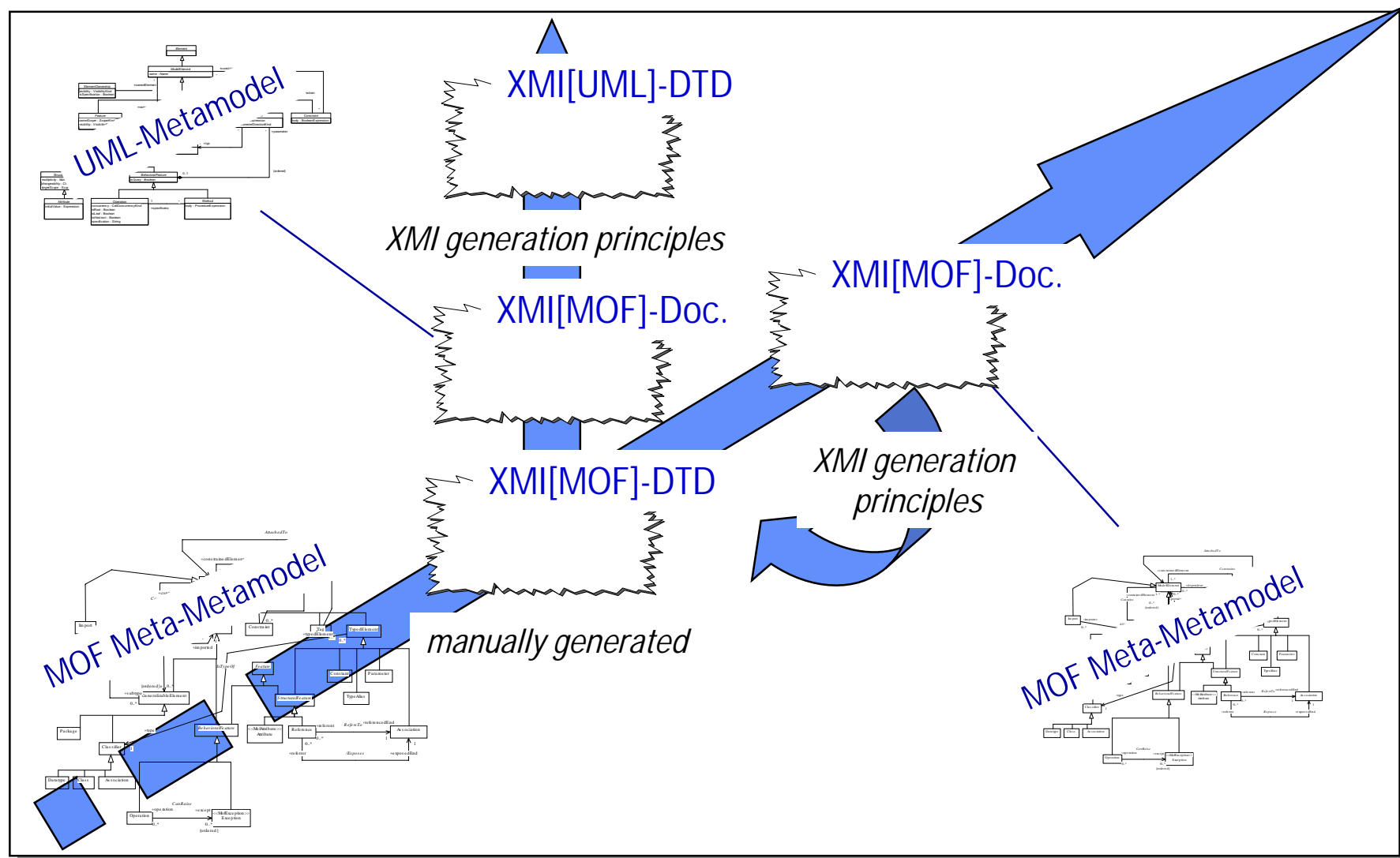
Anwendungsgebiete von XMI -- Transfer von Modellierungssprachen



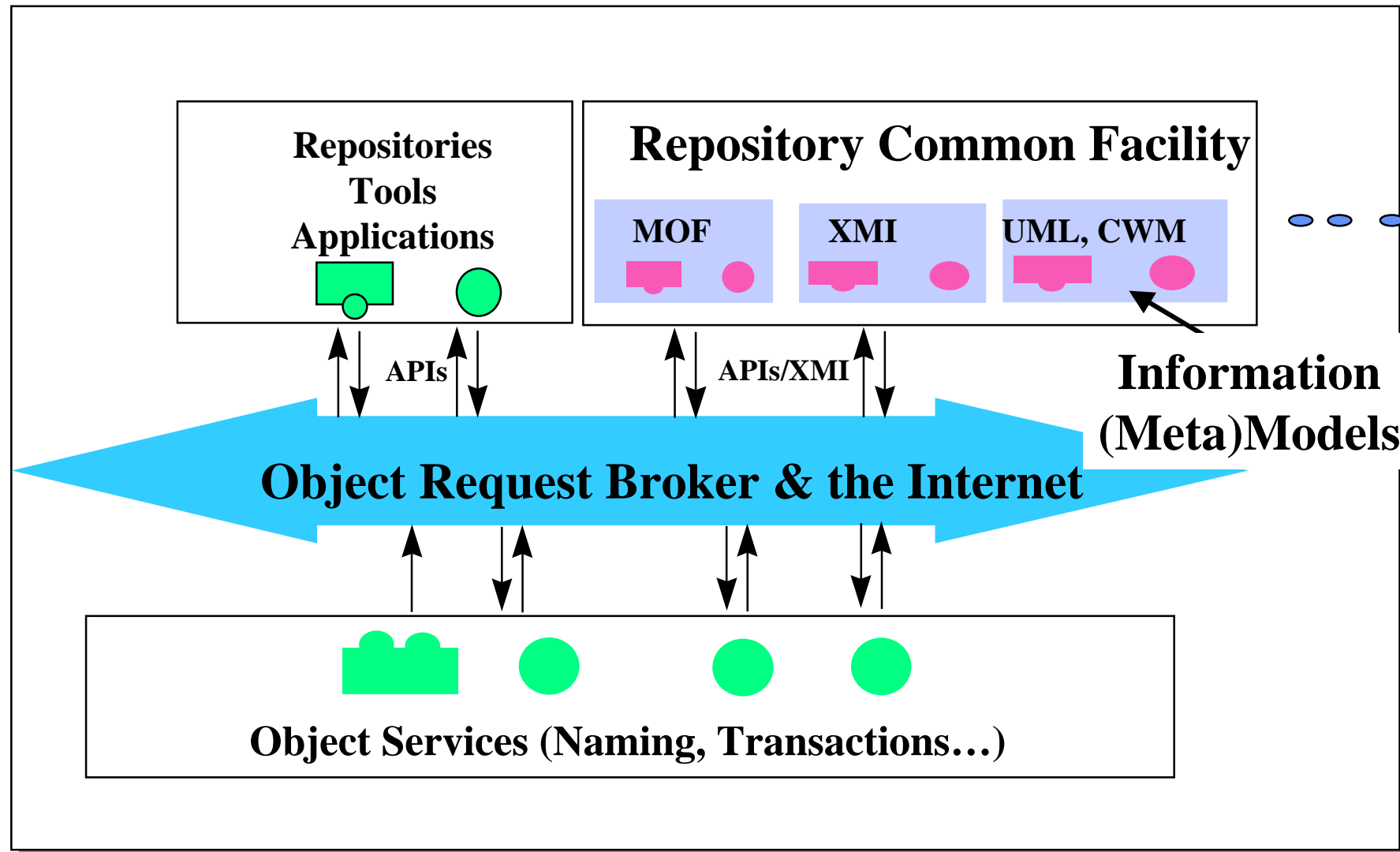
Anwendungsgebiete von XMI -- Transfer von Modellierungssprachen



Anwendungsgebiete von XMI -- Transfer von Modellierungssprachen



XMI Praktischer Einsatz -- *OMG's distributed metadata architecture*



XMI Praktischer Einsatz

- **Einsatzbereiche**

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

- **Vorteile**

- Starke Standardbasiertheit
(OMG's UML, MOF, CWM; W3C's XML-Familie)
- *web enabled*
- (vergleichsweise) leichte Implementierbarkeit
- Architekturneutral
- Überwindet Werkzeuggrenzen

XMI -- wer steckt dahinter?

XMI-Partners:

Unisys, IBM, DSTC, Oracle, Platinum, Fujitsu, Softeam,
Recerca Informatica, DaimlerChrysler

Unterstützer:

Genesis, Inline, Rational, Select, Sprint, Cayenne,
Sybase, Xerox, MCI Systemhouse, Boeing, Ardent,
Aviatis, ICONIX, Integrated Systems, Verilog,
Telefonica I+D, Universitat Politecnica de Catalunya,
NCR, Nihon Unisys, NTT

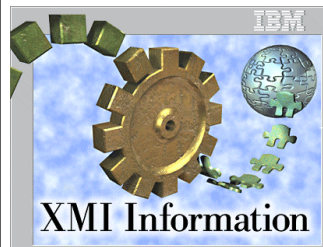
Warum XMI? -- im Überblick

- XMI ist *verabschiedeter OMG-Standard*
 - Die OMG ist mit über 800 Mitgliedern das weltgrößte Software-Konsortium
- XMI integriert die *Unified Modeling Language* (UML)
- XMI basiert auf der *Meta Object Facility* (MOF)
- XMI-Anwendungsgebiete
 - Austausch von UML-Modellen
(Rational Rose, Together, MID's Innovator, Software through Pictures, OTW)
 - Middleware-bezogene Komponenteninformation (CORBA Components)
 - Austausch von Data Warehouse Models (cwm)
 - Vertikale Datenintegration (e.g. clinical information)
- Auf Basis von XMI lassen sich beliebige DTDs aus UML-Datenmodellen generieren
- XMI reagiert auf zukünftige Entwicklungen im Umfeld XML (XML Schema)
- *Meta Data Coalition* (MDC) hat Mapping MDC's OIM auf XMI angekündigt. Eine XMI-DTD für OIM existiert (XMI[OIM])

Erfahrungen aus dem praktischen XMI-Einsatz

- XMI v1.1 ist stabil und einsetzbar
 - DTD ist schlanker als v1.0 => kleinere Dokumente, weniger Overhead
 - Struktur intuitiver und flexibler
- Werkzeugunterstützung wächst stetig
- automatisierte DTD-Generierung beschleunigt XML-Integration in laufenden Entwicklungsprozeß (insbesondere bei Schemaänderungen)
- XMI als XML-Sprache eröffnet Zugang zur XML-Sprachfamilie (XSL(T)!)

References



XMI @ OMG: www.omg.org/news/pr99/xmi_overview.html

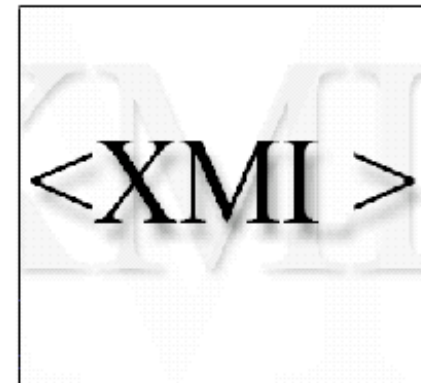
XMI @ IBM: www.software.ibm.com/ad/features/xmi.html

XMI @ XML.COM: www.xml.com/xml/pub/n/New_XML-based_OMG_standard:_XMI

IBM's XMI-Toolkit: www.alphaworks.ibm.com/tech/xmitoolkit

www.jeckle.de/xmi.htm

- Offizielle Spezifikation (UML und XMI)
- DTDs (XMI[MOF], XMI[UML]; XMI[E³R])
- Beispiele
- Hintergrundartikel und Präsentationen

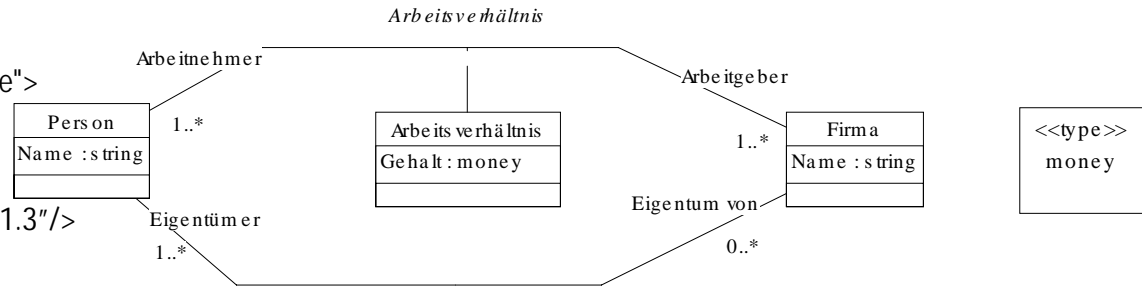


XMI-Code des Beispiels

```

<?xml version="1.0"?>
<!DOCTYPE XMI SYSTEM "xmi/v1.1/99-10-05.dtd">
<XMI timestamp="2000-05-13T21:40:00" verified="true">
  <XMI.header>
    <XMI.model xmi.name="SimpleClassModel"/>
    <XMI.metamodel xmi.name="UML" xmi.version="1.3"/>
  </XMI.header>
  <XMI.content>
    <UML:Class name="Person" visibility="public" isRoot="true" isLeaf="true" isAbstract="false" isActive="true" xmi.id="Person">
      <UML:Classifier.feature>
        <UML:Attribute name="Name" visibility="public" multiplicity="1..1"/>
      </UML:Classifier.feature>
    </UML:Class>
    <UML:Class name="Firma" xmi.id="Firma">
      <UML:Classifier.feature>
        <UML:Attribute name="Name"/>
      </UML:Classifier.feature>
    </UML:Class>
    <UML:Association>
      <UML:Association.connection>
        <UML:AssociationEnd name="Arbeitnehmer" type="Person"/>
        <UML:AssociationEnd name="Arbeitgeber" type="Firma"/>
      </UML:Association.connection>
    </UML:Association>
  </XMI.content>
</XMI>

```



XMI-Code des Beispiels

```
<UML:AssociationClass name="Arbeitsverhältnis" visibility="public">
  <UML:Classifier.feature>
    <UML:Attribute name="Gehalt" type="money" multiplicity="1..1" xmi.id="Gehalt_Arbeitsverhältnis"/>
  </UML:Classifier.feature>
</UML:AssociationClass>
<UML:Association>
  <UML:Association.connection>
    <UML:AssociationEnd name="Eigentümer" multiplicity="1..*" type="Person"/>
    <UML:AssociationEnd name="Eigentum_von" multiplicity="1..*" type="Firma"/>
  </UML:Association.connection>
</UML:Association>
<UML:Class name="money" stereotype="type" xmi.id="money"/>
<UML:Stereotype name="type" xmi.id="type"/>
</XMI.content>
</XMI>
```