

# XML-basierter Metadatenaustausch

Der Standard

*Stream-based Model Interchange Format*  
der Object Management Group

Mario Jeckle

mario.jeckle@daimlerchrysler.com

<http://welcome.to/marioj>

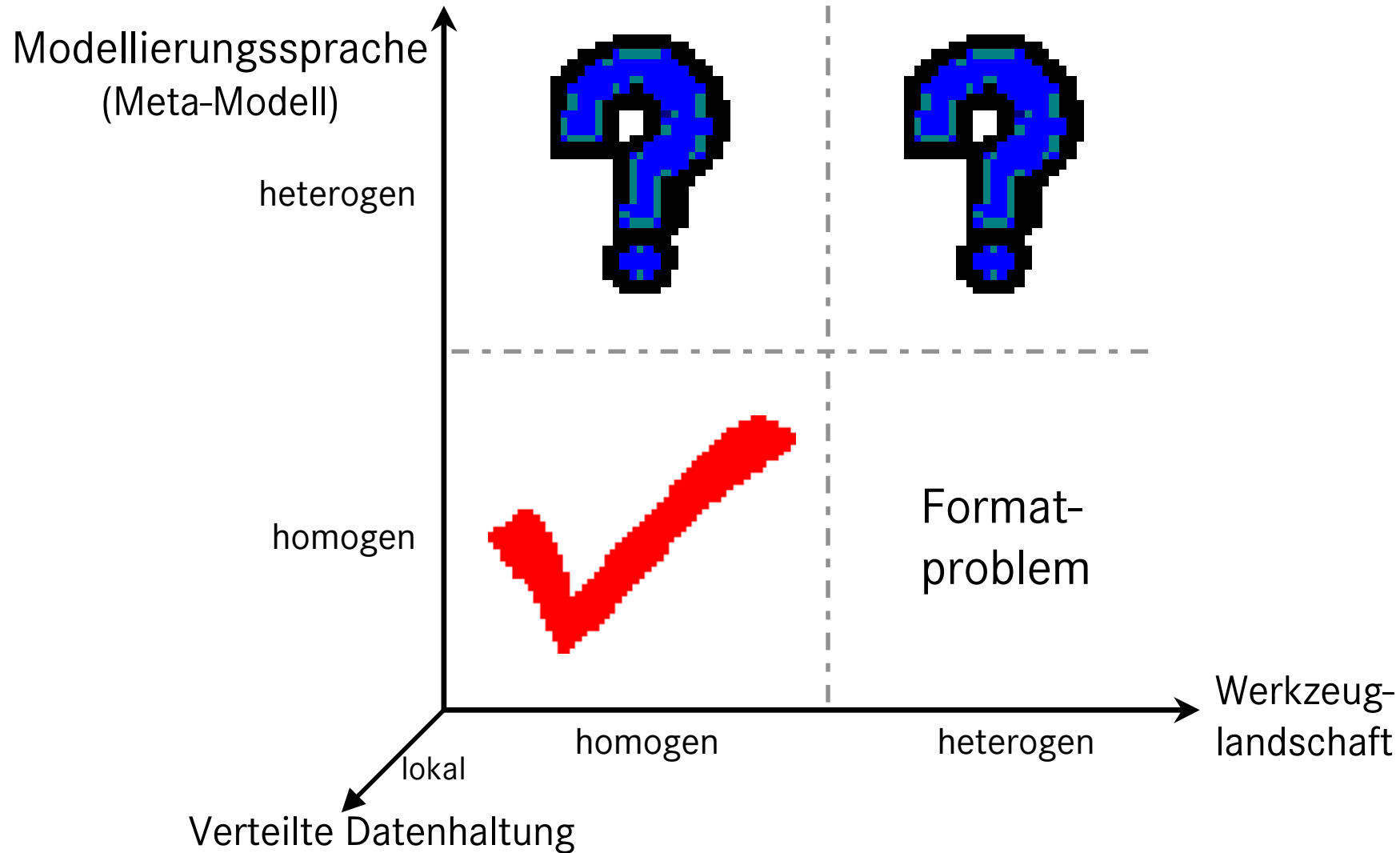
Daimler-Chrysler Forschung und Technologie Ulm  
Abt. Prozeßkette Produktentwicklung (FT3/EK)

Fachgruppentreffen (FG 2.1.9) *Objekt-Orientierte Softwareentwicklung (OOSE)*  
München, 27. Januar 1999

# Übersicht

- Herausforderung *Metadatenaustausch*
- Metamodell Architektur der OMG
- Request for Proposals für ein *Stream-based Model Interchange Format*
- Initial Proposals
  
- Die *Extensible Markup Language (XML)* als Ausgangspunkt von XMI
- XMI als Meta-Meta-(Auszeichnungs)-Sprache
- Beispiel
- Kritik

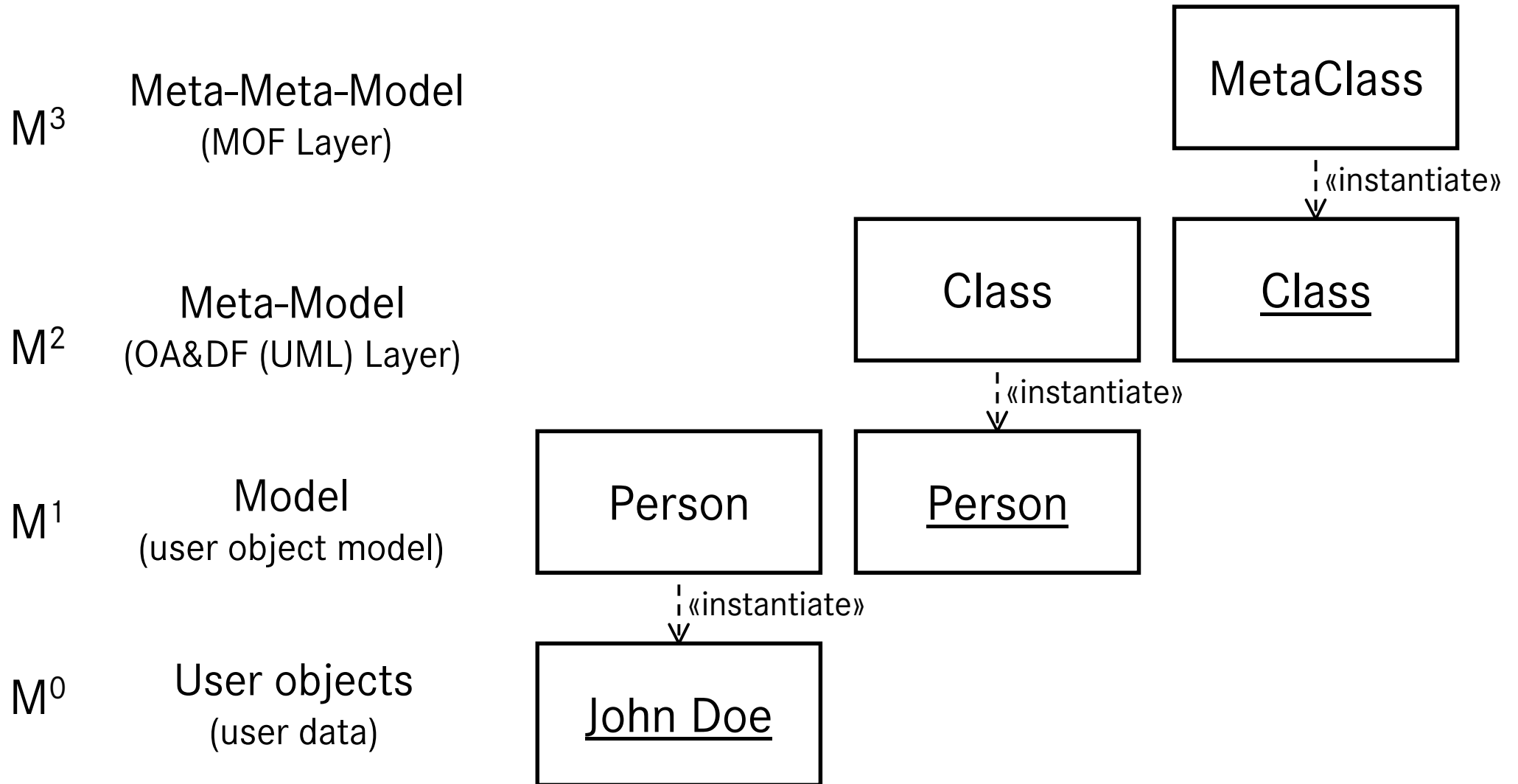
# Herausforderung Metadatenaustausch



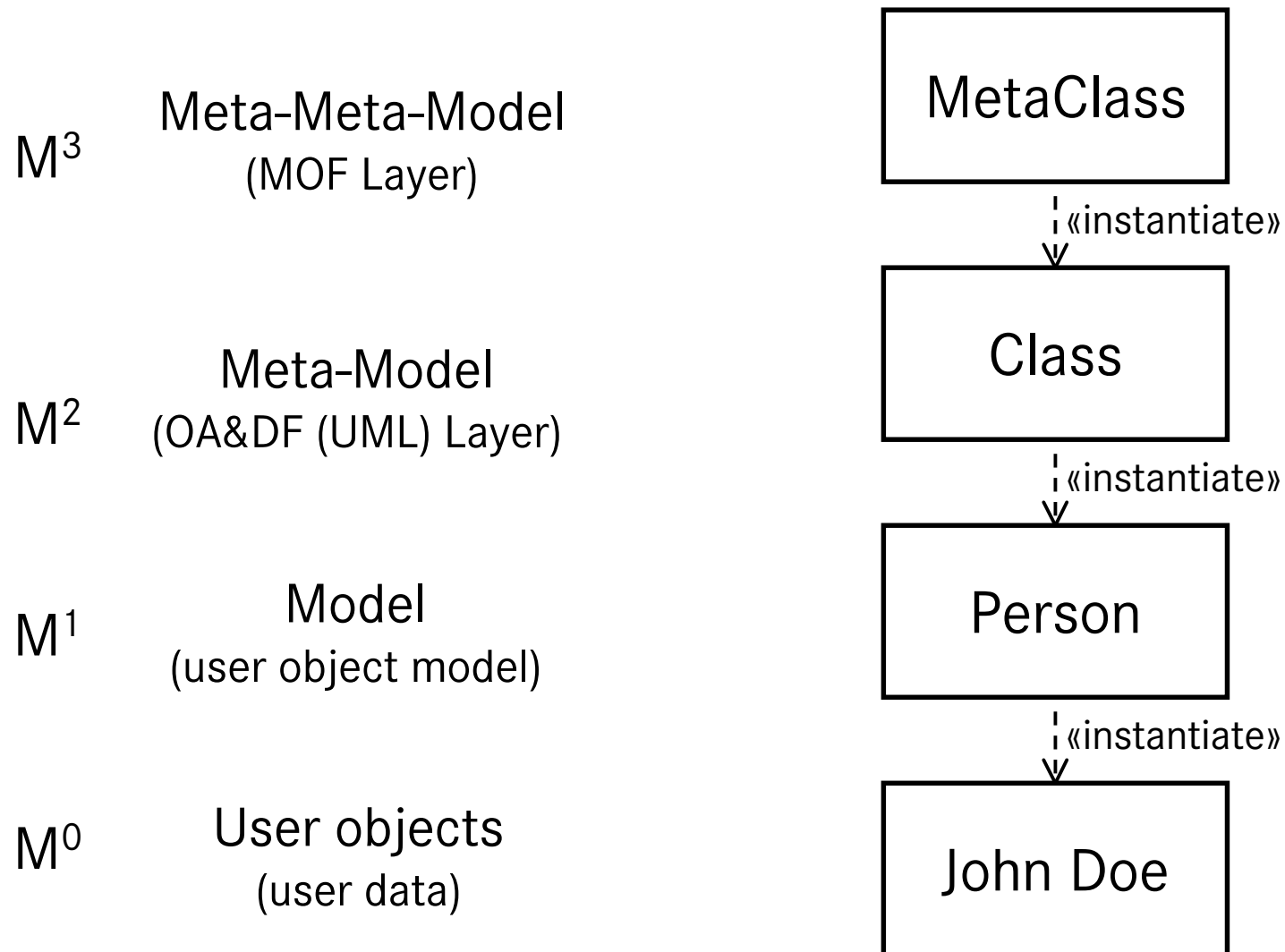
# Einordnung des Stream-based Model Interchange Formats OMG's four-layer Metamodel Architecture

	<i>Layer</i>	<i>Beschreibung</i>	<i>Beispiel</i>
M <sup>3+</sup>	Meta-Meta-Model (MOF Layer)	Sprache zur Beschreibung von Metamodellen	MetaClass
M <sup>2</sup>	Meta-Model (OA&DF (UML) Layer)	Meta-Metamodell- ausprägung	Class
M <sup>1</sup>	Model (user object model)	Metamodell- Ausprägung	StockQuoteServer
M <sup>0</sup>	User objects (user data)	Model- Ausprägung	SQServer_1234

# Modellhierarchien: Beispiel



## Modellhierarchien: Beispiel



# RFP für ein Stream-based Model Interchange Format

## Ziele

- Herstellerunabhängiges Transferformat
- Industriestandard-Formatfestlegung
- generisches Format
- Modellaustausch

## Lösung für

- Transferspezifikation
  - MOF basierter Modelle
  - CDIF codierter Modelle
  - STEP/EXPRESS Modelle
- Metamodell-Versionierung

# Initial Proposals

## Normungsvorschläge und Schwerpunkte

### CDIF

- Mapping zwischen CDIF-Metamodell und MOF
- Meta-Identifizierer zur Meta-Modell-Versionierung

### UOL

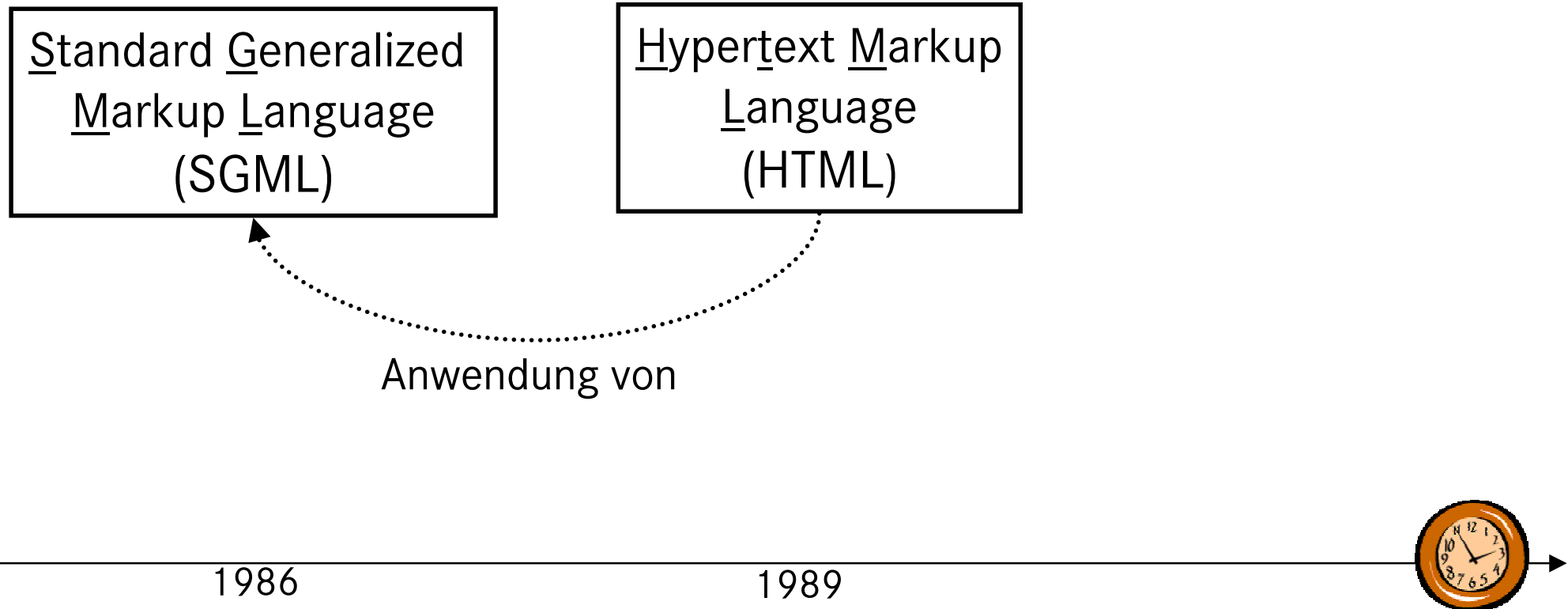
- Human readability
- vollständiges Mapping zwischen STEP/EXPRESS und MOF

### XMI

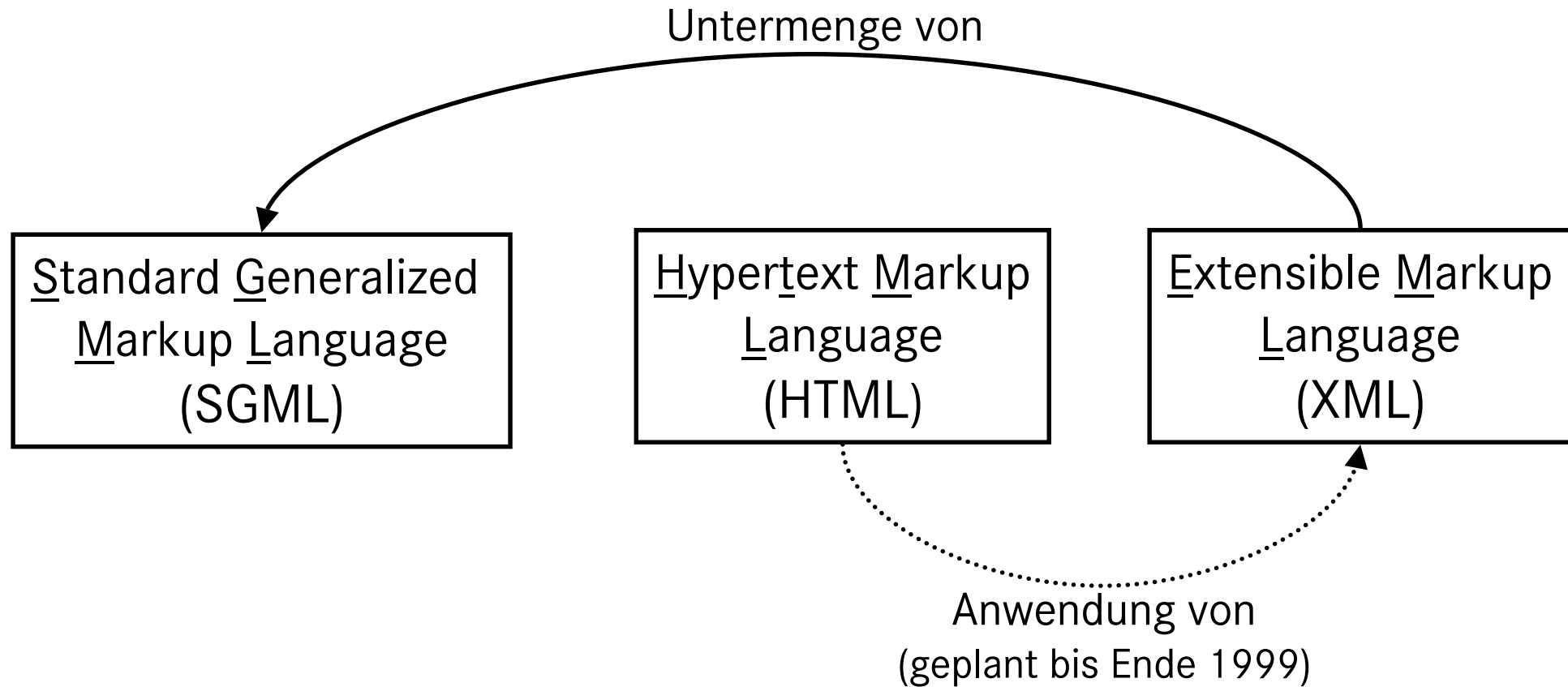
- Nutzung von XML
- DTD generation rules
- Beispiel DTDs für UML und MOF
- Repository Architektur



# Entstehung der Extensible Markup Language (XML)



# Entstehung der Extensible Markup Language (XML)



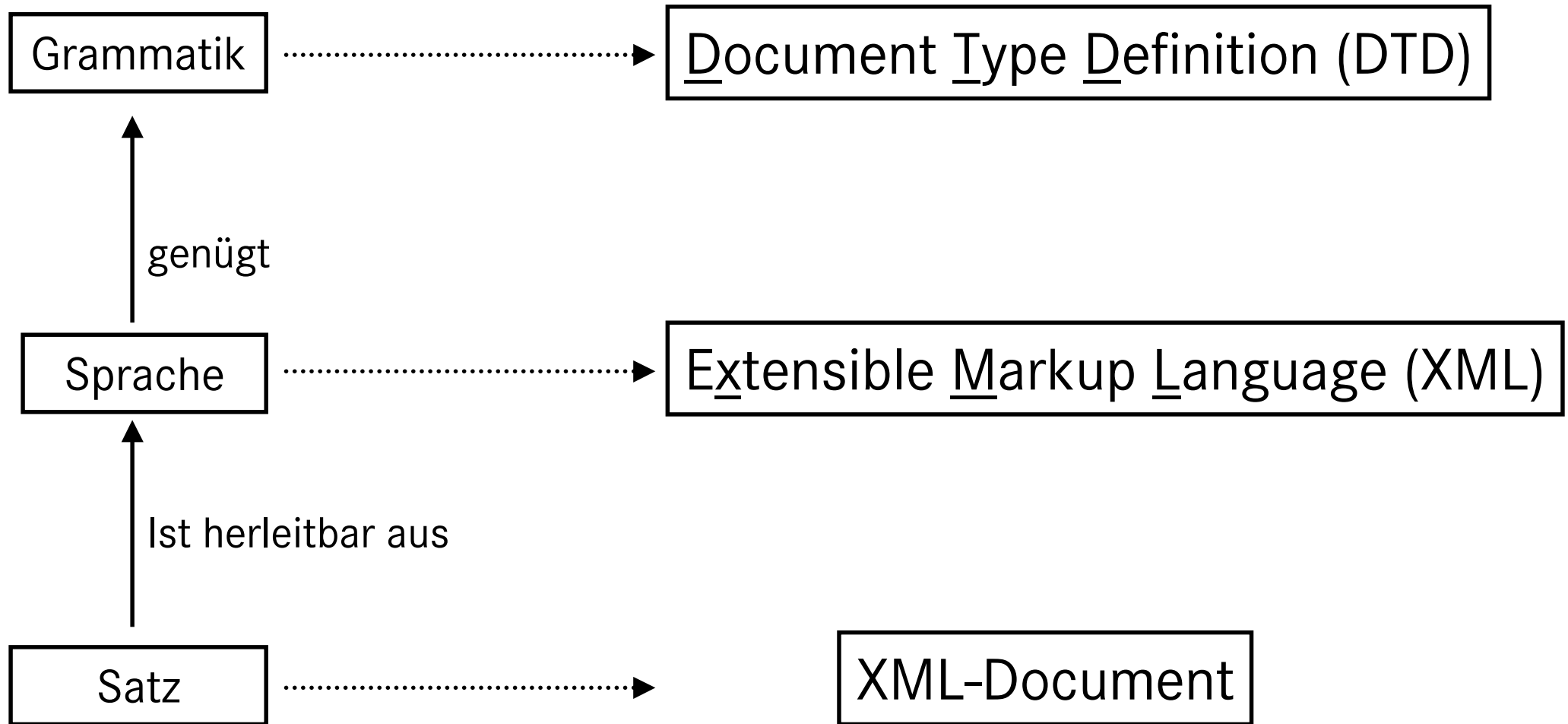
1986

1989

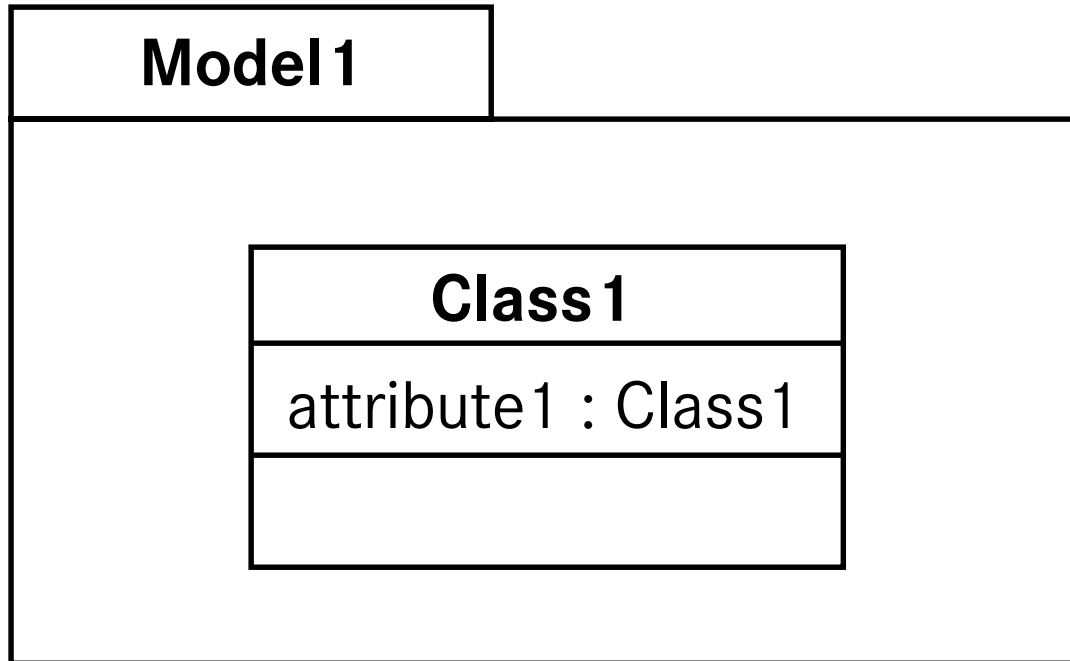
1997



# Extensible Markup Language (XML) – im Überblick



# XML im Beispiel – Grammatik in EBNF-Darstellung



## EBNF-Produktionen:

Model  $\rightarrow$  *Name* Class\*

Class  $\rightarrow$  *Name* Attribute\*

Attribute  $\rightarrow$  *Name* [Cardinality] [*Type*]

Cardinality  $\rightarrow$  *Min Max*

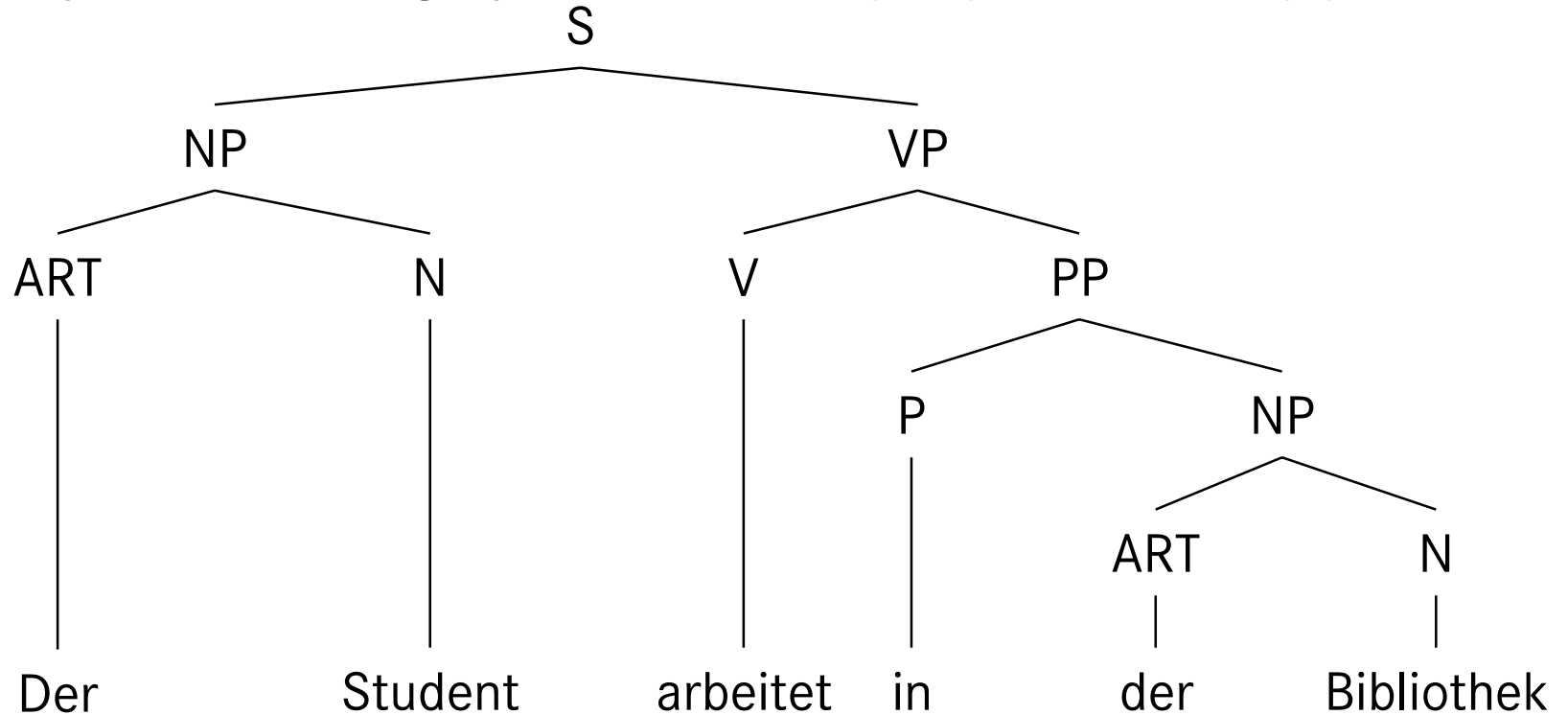
S = Model

NT = {Model, Class, Attribute, Cardinality}

T = {Name, Type, Min, Max}

# Konstituenten-Analyse (Immediate constituent analysis)

**Prinzip:** Der Satz (S) wird in die Kategorie Nominalphrase (NP) und Verbalphrase (VP) geteilt. Die Nominalphrase wird aufgespalten in Artikel (ART) und Nomen (N).

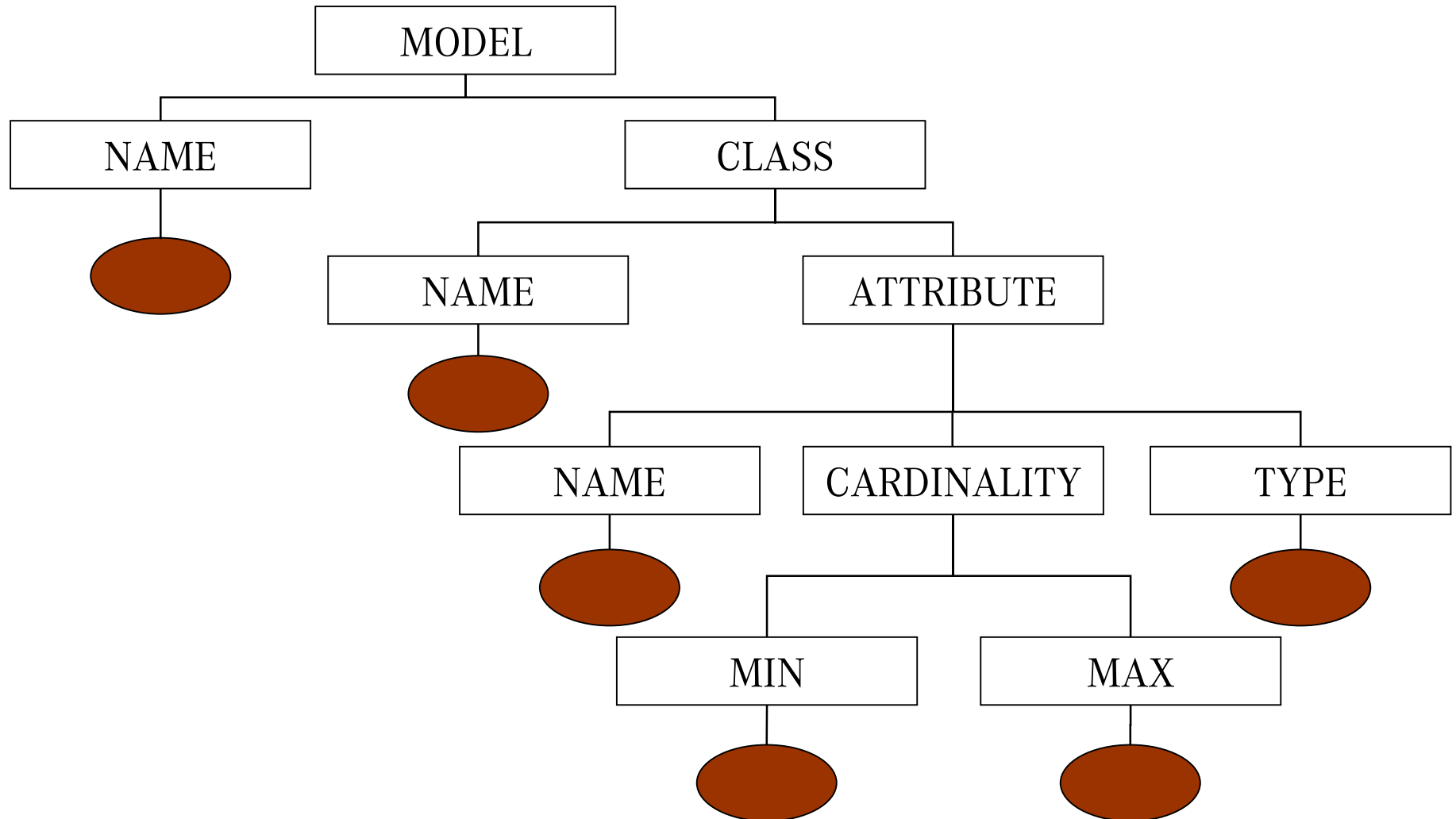


## Äquivalente Klammerdarstellung:

[S[NP[ART Der]ART[N Student]N]NP[VP[V arbeitet]V[PP[P in]P  
 [NP[ART der]ART [N Bibliothek]N]NP]PP]VP]S

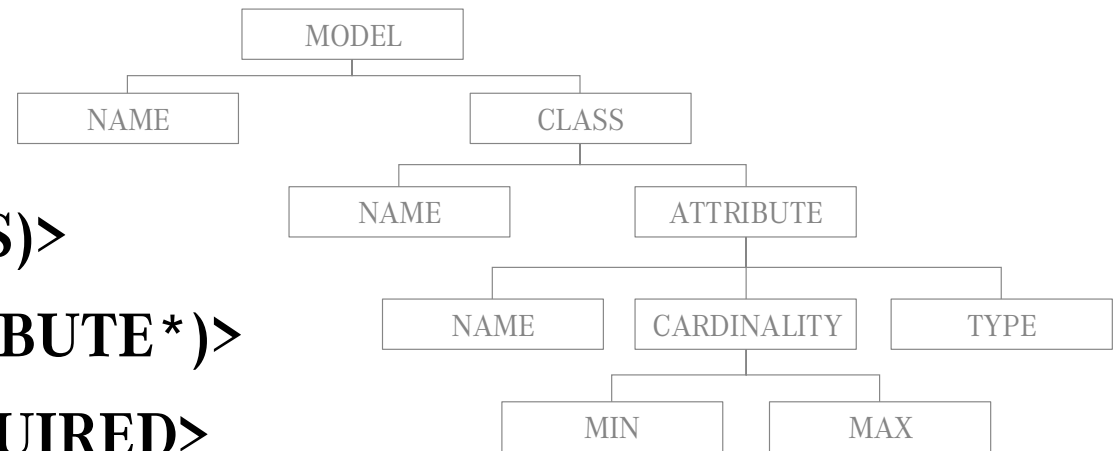
Beispiel nach:  
 Brockhaus98, 20. Aufl., Bd. 12, S. 317

## XML im Beispiel – Baumstruktur



Baumartige Struktur des gesamten Dokuments mit Nutzdaten an den Blattknoten

# XML im Beispiel – die Document Type Definition



**<!ELEMENT MODEL (Name,CLASS)>**

**<!ELEMENT CLASS (Name, ATTRIBUTE\*)>**

**<!ELEMENT Name #PCDATA REQUIRED>**

**<!ELEMENT ATTRIBUTE (Name, CARDINALITY?, Type?)>**

**<!ELEMENT CARDINALITY (Min, Max)>**

**<!ELEMENT Min (#PCDATA) REQUIRED>**

**<!ELEMENT Max (#PCDATA)>**

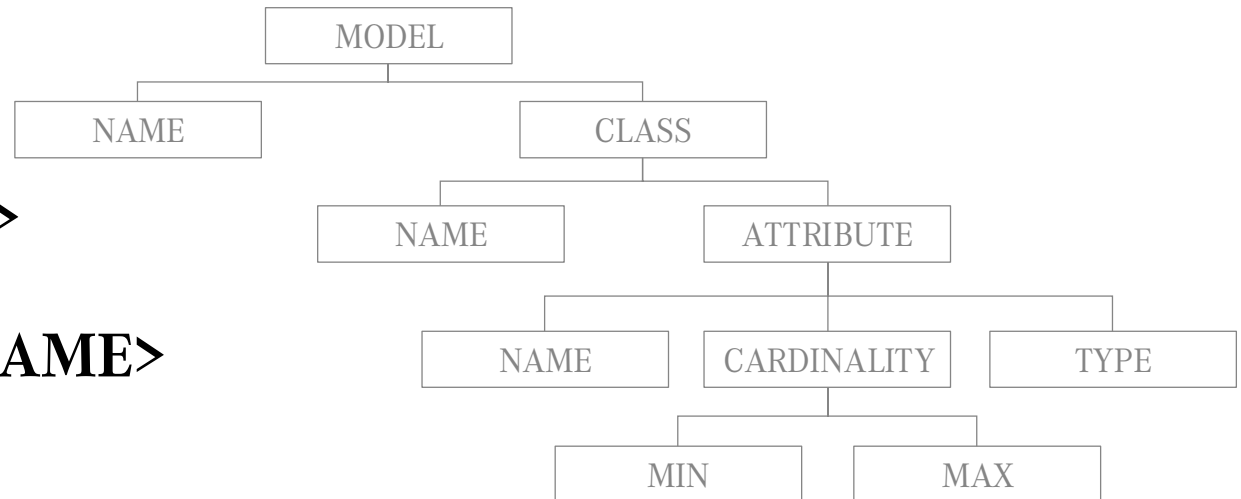
**<!ELEMENT Type (#PCDATA)>**

## XML im Beispiel – Dokumentinstanz

```

<?xml version= "1.0"?>
<!DOCTYPE oomodel SYSTEM "oom.dtd">
<MODEL>
  <NAME>"Model1"</NAME>
  <CLASS>
    <NAME>"Class1"</NAME>
    <ATTRIBUTE>
      <NAME>"attribute1"</NAME>
      <CARDINALITY>
        <MIN>"1"</MIN>
        <MAX>"1"</MAX>
      </CARDINALITY>
      <TYPE>"Class1"</TYPE>
    </ATTRIBUTE>
  </CLASS>
</MODEL>

```



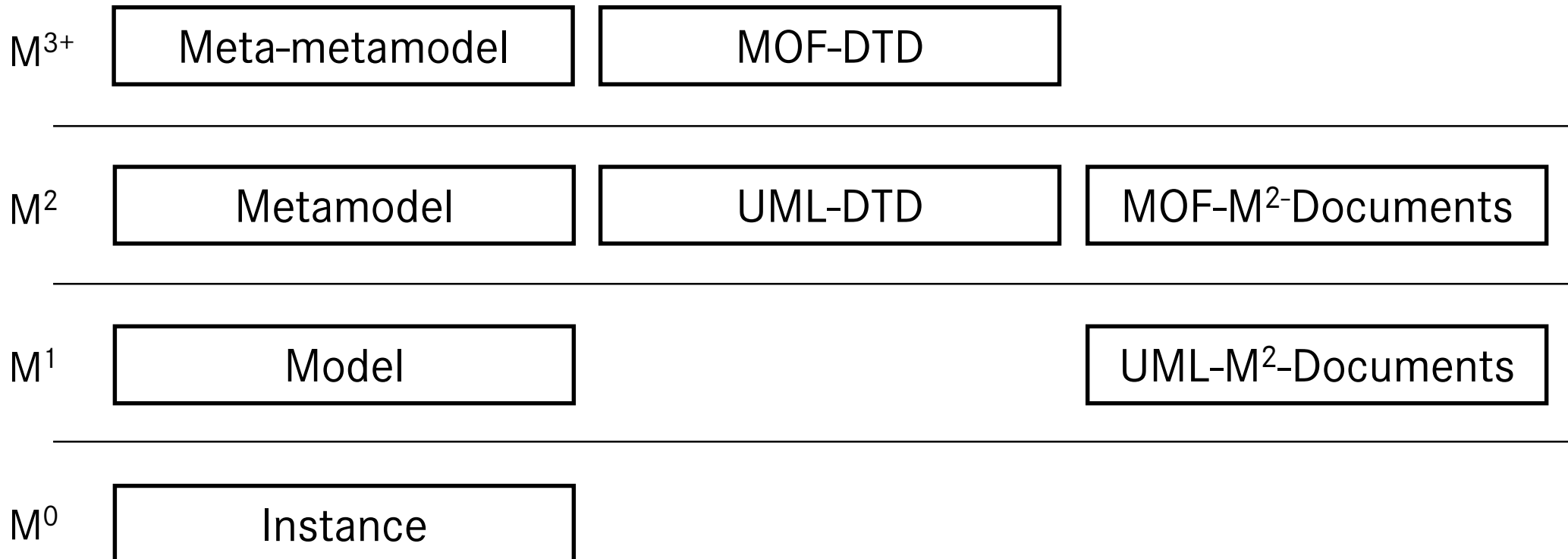


# XMI als M<sup>3</sup>-Sprache

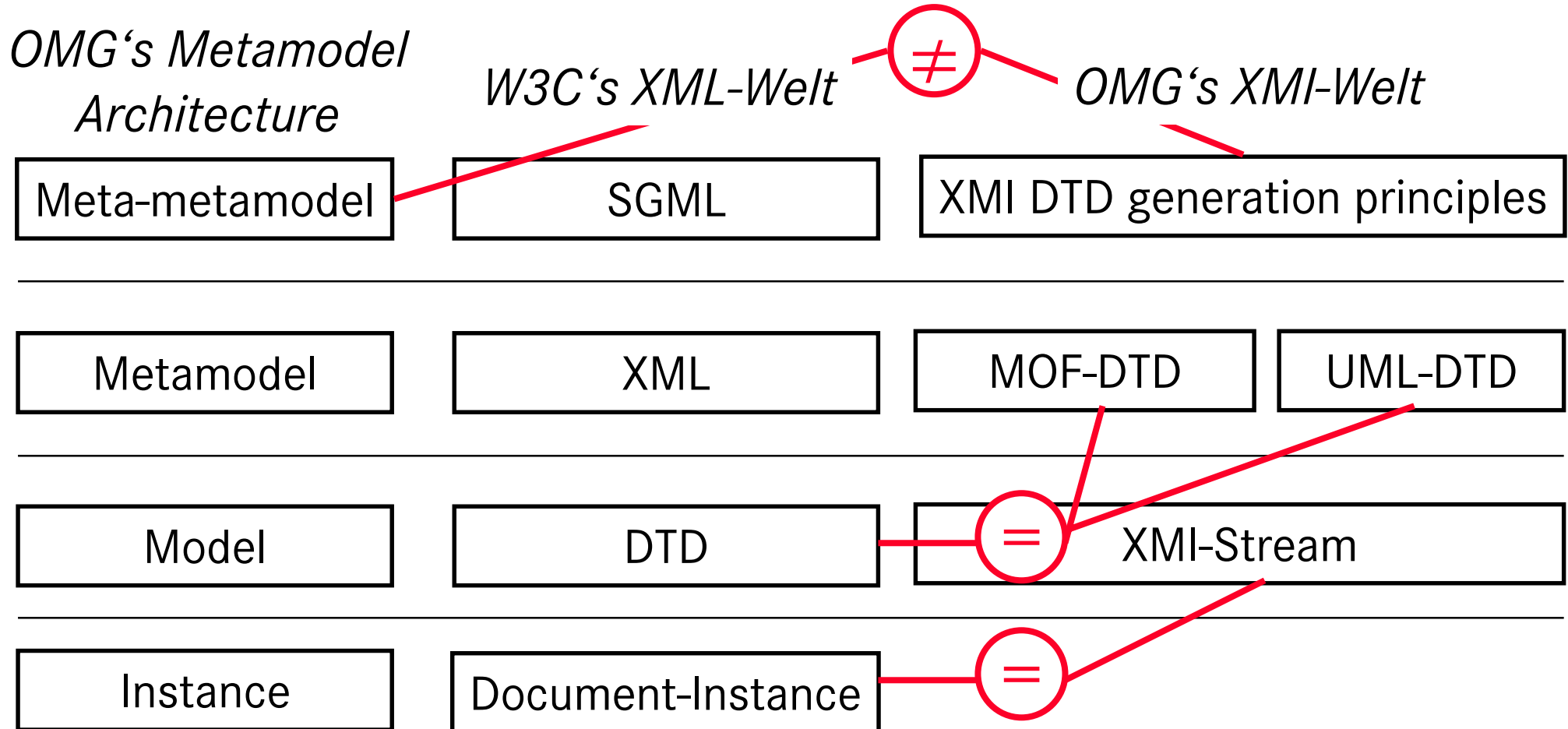
*OMG's Metamodel  
Architecture*

*XMI-DTDs*

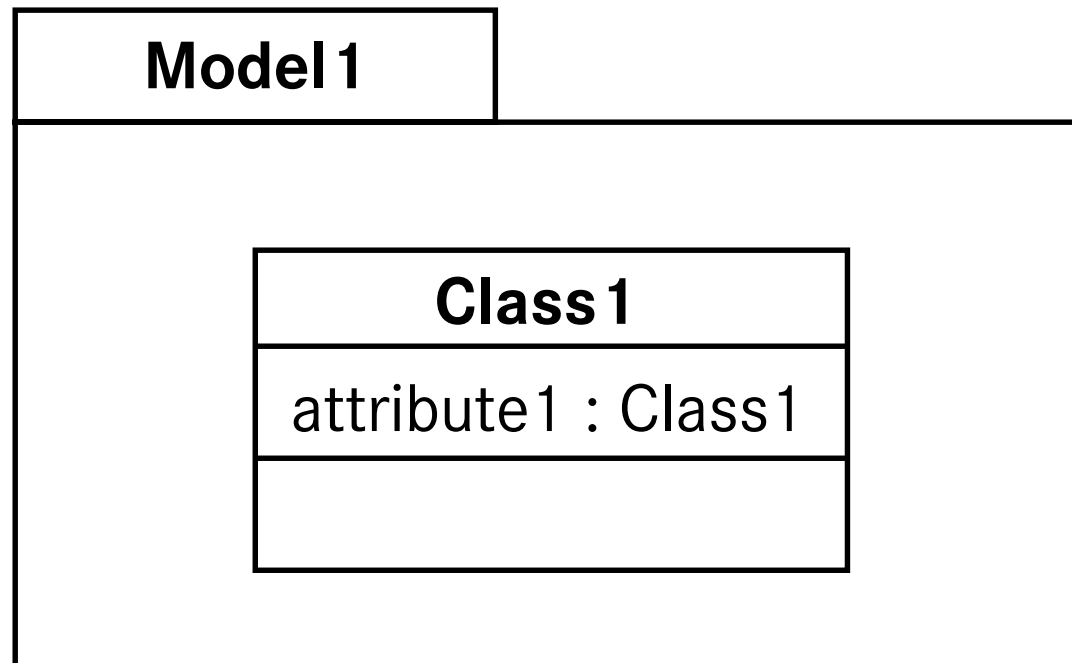
*XMI-Documents*



# XMI als M<sup>3</sup>-Sprache



# XML basierter Metadatenaustausch (XMI)



# XML basierter Metadatenaustausch (XMI)

```

<?xml version = "1.0"?>
<!DOCTYPE XMI SYSTEM "uml.dtd">
<XMI xmi.version="1.0">
  <XMI.header>
    <XMI.metamodel xmi.name="uml" xmi.version="1.1"/>
  </XMI.header>
  <XMI.content>
    <Model_Management.Model xmi.id="i00000001">
      <Foundation.Core.ModelElement.name>
        Modell
      </Foundation.Core.ModelElement.name>
      <Foundation.Core.ModelElement.visibility xmi.value="public"/>
      <Foundation.Core.GeneralizableElement.isRoot xmi.value="true"/>
      <Foundation.Core.GeneralizableElement.isLeaf xmi.value="true"/>
      <Foundation.Core.GeneralizableElement.isAbstract xmi.value="false"/>
      <Foundation.Core.Namespace.ownedElement>
        <Foundation.Core.Class xmi.id="i00000002">
          <Foundation.Core.ModelElement.name>
            class1
          </Foundation.Core.ModelElement.name>
          <Foundation.Core.ModelElement.visibility xmi.value="public"/>
          <Foundation.Core.GeneralizableElement.isRoot xmi.value="true"/>
          <Foundation.Core.GeneralizableElement.isLeaf xmi.value="true"/>
          <Foundation.Core.GeneralizableElement.isAbstract xmi.value="false"/>
          <Foundation.Core.Class.isActive xmi.value="true"/>
          <Foundation.Core.Classifier.feature>
            <Foundation.Core.Attribute xmi.id="i00000003">
              <Foundation.Core.ModelElement.name>
                attribute1
              </Foundation.Core.ModelElement.name>
              <Foundation.Core.ModelElement.visibility xmi.value="public"/>
              <Foundation.Core.Feature.ownerScope xmi.value="instance"/>
              <Foundation.Core.StructuralFeature.multiplicity>
                1
              </Foundation.Core.StructuralFeature.multiplicity>
              <Foundation.Core.StructuralFeature.changeable xmi.value="none"/>
              <Foundation.Core.StructuralFeature.targetScope xmi.value="instance"/>
              <Foundation.Core.Attribute.initialValue>
                <Foundation.Data_Types.Expression>
                  <Foundation.Data_Types.Expression.language>
                    </Foundation.Data_Types.Expression.language>
                  <Foundation.Data_Types.Expression.body>
                    0
                  </Foundation.Data_Types.Expression.body>
                </Foundation.Data_Types.Expression>
              </Foundation.Core.Attribute.initialValue>
              <Foundation.Core.StructuralFeature.type>
                <Foundation.Core.DataType xmi.idref = "i00000004"/> <!-- integer -->
              </Foundation.Core.StructuralFeature.type>
            </Foundation.Core.Attribute>
          </Foundation.Core.Classifier.feature>
        </Foundation.Core.Class>
      </Foundation.Core.Namespace.ownedElement>
    </Model_Management.Model>
  </XMI.content>
</XMI>

```

## Literatur und Referenzen

### Initial Submissions:

- CDIF Partners: *Stream-based Model Interchange Format*, OMG-Dokument ad/98-07-09
- UOL Partners: *Universal Object Language*, OMG-Dokument ad/98-07-07
- XMI Partners: *XML Metadata Interchange*, OMG-Dokument ad/98-07-01
  
- Bruce McLean: *Evaluation Report OMG OA&DTF RFP3 Stream-based Model Interchange Format (SMIF)*, OMG A&D-TF

### Revised Joint-Submission:

- XMI Partners: *XML Metadata Intechange*, OMG-Dokument ad/98-10-05

### WWW:

- <http://www.software.ibm.com/ad/features/xmi.html>
- <http://www.ontogenics.com/repository/Default.htm>
- <http://www.krumbach.de/home/jeckle>