

# 5. XMI - XML Metadata Interchange

---

- 5.1 Meta Object Facility (MOF)
- 5.2 Extensible Markup Language (XML)
- 5.3 XML Metadata Interchange (XMI)

## 5.1 Meta Object Facility (MOF)

---

- **Meta Object Facility (MOF)** ist **OMG-Standard** für **Spezifikation von Metadaten**
- Zugrundeliegende Idee: Spezifikation von Metadaten soll **objektorientiert** geschehen

⇒ Auswahl einer Untermenge von UML zur Modellierung von Metadaten (**MOF Core**)

## 5.2 Extensible Markup Language (XML)

---

- Wurzeln in der elektronischen Textverarbeitung
- **Grundidee:** Trennung von Inhalt, Struktur und Informationen zur Darstellung
  - **Inhalt:** Text, Bilder, ...
  - **Struktur:** Kapitel, Abschnitt, Zitat, ...
  - **Informationen zur Darstellung:** Formatierung, Layout, Schriftart, ...

# XML - Chronologie der Markup Languages

---

## Kurze Chronologie der Markup Languages:

- 1978 Standard General Markup Language (SGML)
- 1989 Hypertext Markup Language (HTML)
- ab 1994 zunehmende **Abweichung** von HTML-Standard durch Microsoft und Netscape
- 1997 „Wiederentdeckung“ der SGML, Definition der Extensible Markup Language (XML) als Untermenge von SGML  $\Rightarrow$  Vereinfachung

# XML - Unterschied XML, HTML

---

## Unterschied zwischen XML und HTML:

- HTML besitzt nur eine begrenzte, fest vorgegebene Menge von Strukturelementen, z.B.: <P>, <H1>
- XML erlaubt die Spezifikation beliebiger Dokumenttypen
  - Für jeden Dokumenttyp können beliebig viele Strukturelemente festgelegt werden
- HTML ist ein spezieller Dokumenttyp, der in SGML definiert wurde

# XML - Bestandteile

---

XML beinhaltet:

- **Dokument Typ Definitionen (DTD) (Struktur)**
  - Festlegung der Strukturelemente („Tags“) und deren Aufbau und Attribute
- **XML-Dokumente (Inhalt)**
  - Text, Grafik, etc., welche gemäss einer DTD aufgebaut sind
- **Style Sheets (Informationen zur Darstellung)**
  - Konkrete Angaben, wie die Strukturelemente darzustellen sind

# XML - Beispiel DTD

---

## Beispiel (DTD für Emails):

```
<!ELEMENT email (empfaenger, absender, thema,  
                nachricht)>  
  
<!ELEMENT empfaenger (#PCDATA)>  
<!ELEMENT absender (#PCDATA)>  
<!ELEMENT thema (#PCDATA)>  
<!ELEMENT nachricht (#PCDATA)>
```

# XML - Beispiel Dokument

---

## Beispiel (XML Dokument):

```
<?xml version="1.0"?>
<!DOCTYPE email SYSTEM "email1.dtd">
<email>
  <empfaenger>Dino Ahr</empfaenger>
  <absender>Dino Ahr</absender >
  <thema>Test</thema>
  <nachricht>
    Dies ist ein Test!
  </nachricht>
</email>
```



# XML - Vorteile

---

## Vorteile von XML:

- Systemunabhängiger Austausch von Metadaten und Daten über das Internet
- Konzept bereits erprobt mit HTML
- Einfacher zu verstehen und zu benutzen als andere Metadaten-Technologien
- grosse Akzeptanz und Unterstützung von Seiten der Industrie

## 5.3 XML Metadata Interchange (XMI)

---

### Ausgangspunkt:

- 1998: **OMG-Ausschreibung** für ein **textbasiertes Austauschformat für Metadaten und Daten** (MOF Core besitzt nur die graphischen Elemente von UML!)
- 3 Vorschläge, von denen letztendlich XMI akzeptiert wurde (u.a. IBM, Oracle, Unisys)
- **Anfang 1999**: erste XMI Spezifikation
- **Feb. 2000**: Verabschiedung XMI V1.1

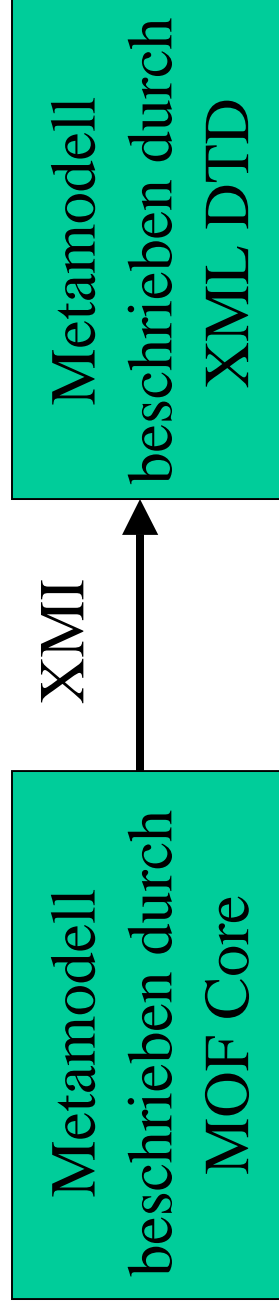
# XMI - Realisierung

---

## Realisierung:

- Definition einer Abbildung von MOF Core Elementen auf XML DTD Elemente

⇒ Automatische Generierung von DTDs möglich!



# XMI - Nutzen für UML

---

## Nutzen/Entwicklung für UML:

- UML ist in seiner Spezifikation vollständig mit MOF Core Elementen beschrieben
- ⇒ Mittels XMI lässt sich eine **UML DTD** erstellen
  - August 1999: offizielle DTD für UML 1.3
- ⇒ **UML Modelle** lassen sich in ein **XML Dokument transformieren** und somit leicht austauschen
- September 2000: XMI-Erweiterung für Rational Rose

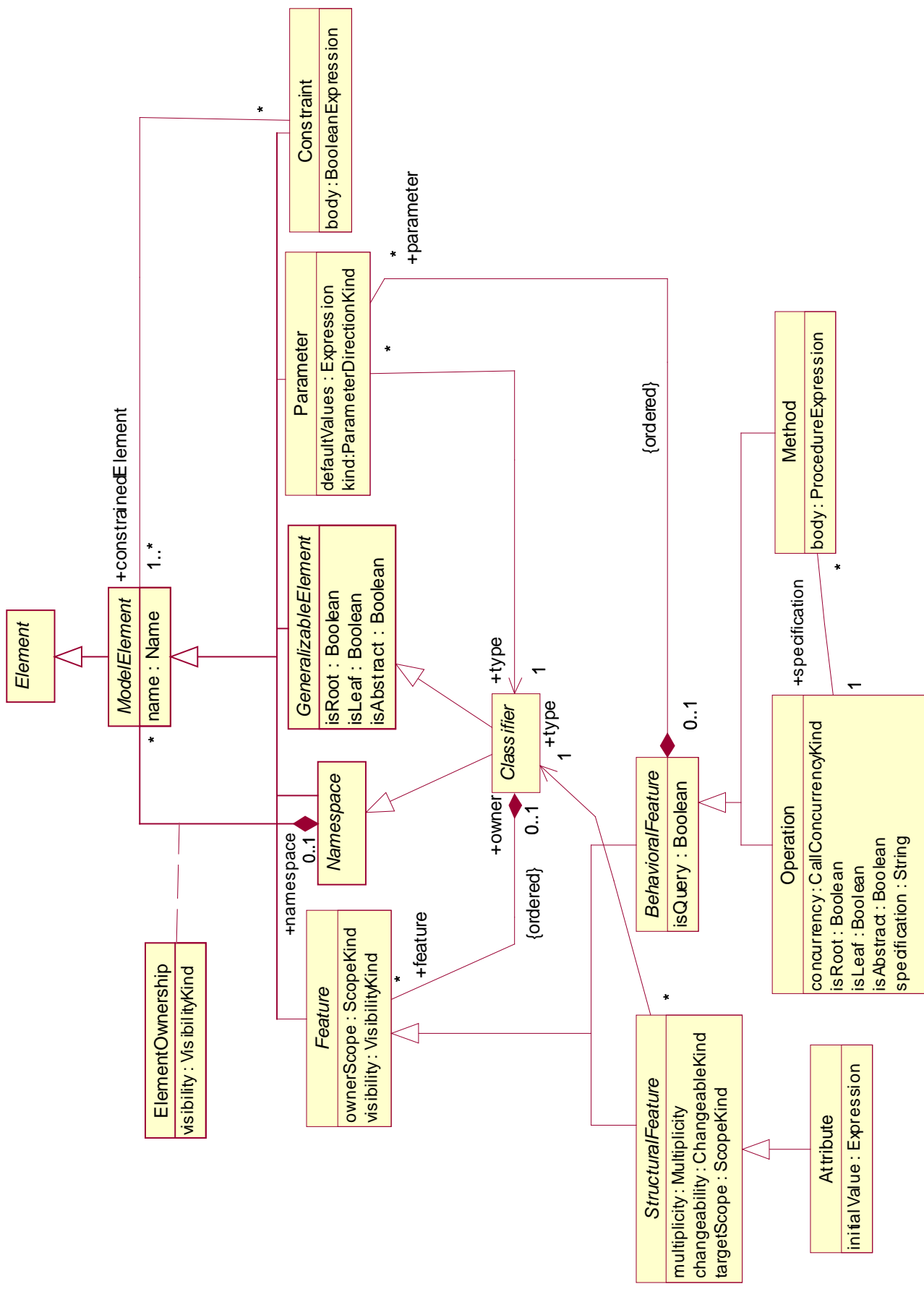
# XMI - Auszug UML1.3 DTD

---

## Auszug aus der UML1.3 DTD:

```
<?xml version="1.0" encoding="UTF-8" ?>
...
<!--
<!--
<!-- XMI is the top-level XML element for XMI transfer text
<!--
<!ELEMENT XMI (XMI.header, XMI.content?, XMI.difference*,
XMI.extensions*) >
...
```

# Zur Erinnerung: Klassendiagramm Foundation.Core - Backbone



# XMI - Auszug UML1.3 DTD (Forts.)

---

```
<!ELEMENT Foundation.Core.Class (Foundation.Core.ModelElement.name?,  
    ..  
    Foundation.Core.GeneralizableElement.isRoot?,  
    Foundation.Core.GeneralizableElement.isLeaf?,  
    Foundation.Core.GeneralizableElement.isAbstract?,  
    ..  
    Foundation.Core.Classifier.feature*)? >  
  
...  
  
<!ELEMENT Foundation.Core.Classifier.feature (Foundation.Core.Feature |  
    Foundation.Core.BehavioralFeature |  
    Behavioral_Elements.Common_Behavior.Reception |  
    Foundation.Core.Method |  
    Foundation.Core.Operation |  
    Foundation.Core.StructuralFeature |  
    Foundation.Core.Attribute)* >
```

# XMI - Auszug UML1.3 DTD (Forts.)

---

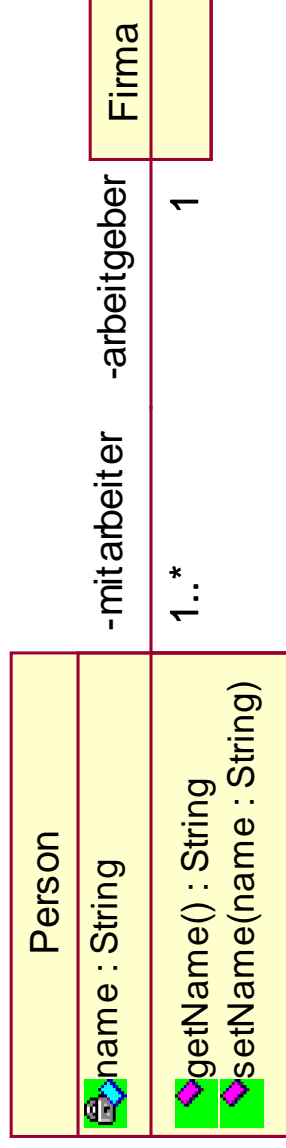
```
<!ELEMENT Foundation.Core.Attribute (Foundation.Core.ModelElement.name?,  
Foundation.Core.ModelElement.visibility?,  
...  
Foundation.Core.StructuralFeature.multiplicity?,  
... >
```



# XMI - UML-Modell als XML- Dokument

---

## Beispiel (UML-Modell):



# XMI - UML-Modell als XML-Dokument (Forts.)

---

## Beispiel (dazugehöriges XML-Dokument):

```
<?xml version = '1.0' encoding = 'ISO-8859-1' ?>
<!DOCTYPE XMI SYSTEM 'uml.dtd' />
...
<Foundation.Core.Class xmi.id = 'S.10001'>
<Foundation.Core.ModelElement.name>Person</Foundation.Core.ModelElement.name>
...
<Foundation.Core.GeneralizableElement.isRoot xmi.value = 'true' />
<Foundation.Core.GeneralizableElement.isLeaf xmi.value = 'true' />
<Foundation.Core.GeneralizableElement.isAbstract xmi.value = 'false' />
...
```

# XMI - UML-Modell als XML- Dokument (Forts.)

---

```
<Foundation.Core.Classifier.feature>
  <Foundation.Core.Attribute xmi.id = 'S.10002'>
    <Foundation.Core.ModelElement.name>name</Foundation.Core.ModelElement.name>
    <Foundation.Core.ModelElement.visibility xmi.value = 'private' />
    ...
  <Foundation.Core.StructuralFeature.multiplicity>1..1
    </Foundation.Core.StructuralFeature.multiplicity>
  ...
</Foundation.Core.Attribute>
...
</Foundation.Core.Classifier.feature>
...
</Foundation.Core.Class>
```

# XMI - Zusammenfassung

---

- XMI = Abbildung der MOF Core Elemente auf XML DTD Elemente
- **Textbasiertes Format** für Metadaten und Daten
- Für UML: Generierung einer UML DTD aus der UML Spezifikation  
⇒ **Generisches Austauschformat für UML Modelle**