

XML / XMI

Peter Braun

Institut für Informatik

Lehrstuhl für Software und
Systemsengineering

Inhalt

- Einführung in XML
- Geschichte
- Ziele/Defizite von XML
- Anwendungsgebiete
- XMI – Warum?
 - Geschichte, Hintergründe
- XMI – Wie funktioniert's?
- XMI und UML
- Quellen, Tools, ...

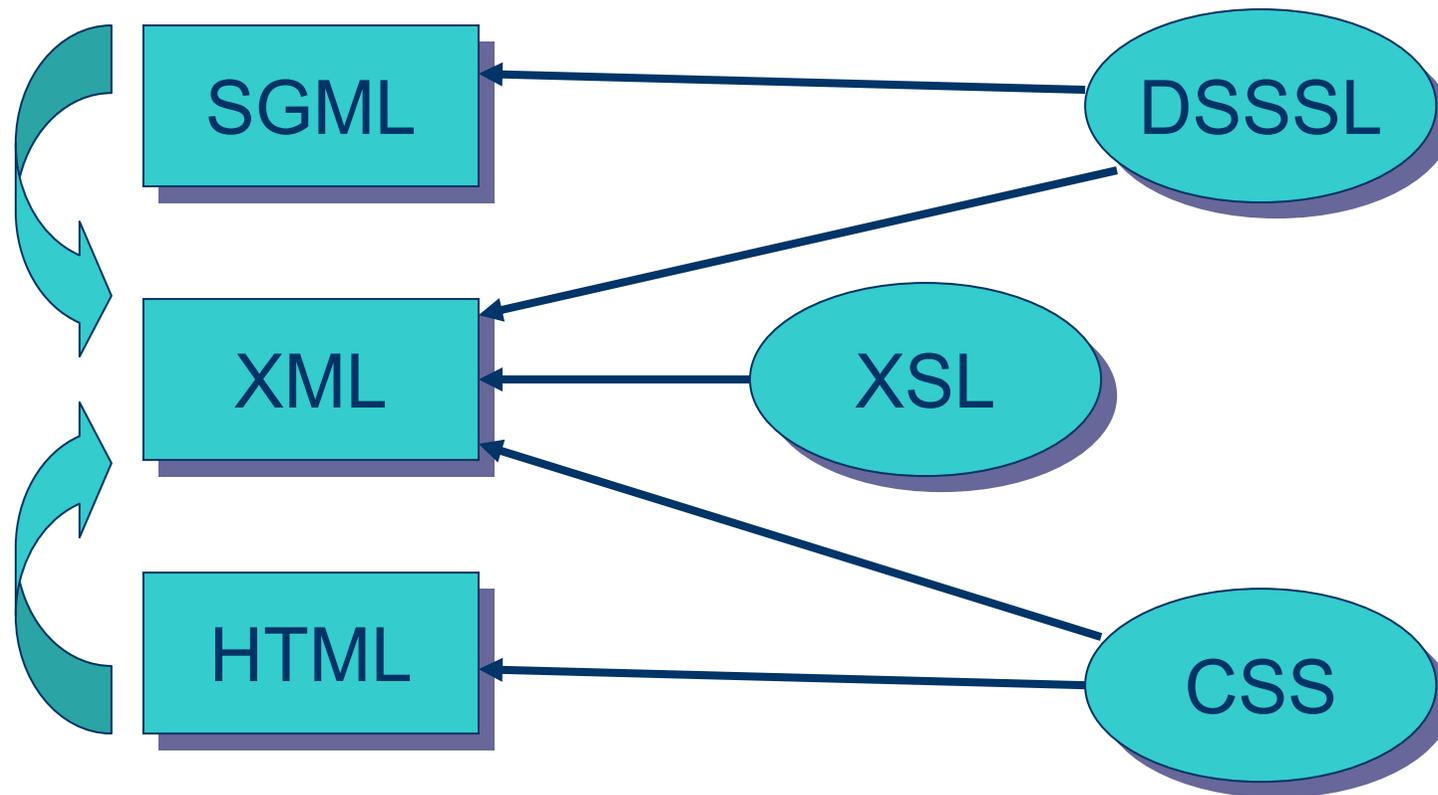
Einführung in XML (1)

- XML: eXtensible Markup Language
- XML ist eine Metasprache, d.h.

XML ist eine Sprache, mit deren Hilfe andere Sprachen definiert werden können.

„XML“ bietet eine Fülle von Technologien

Geschichte von XML



Ein Beispiel (1)

- XML – Dokument:

```
<?xml version="1.0">
```

```
<anschrift>
```

```
  <name>Karl Mustermann</name>
```

```
  <strasse>Bergweg 22</strasse>
```

```
  <ort>München</ort>
```

```
</anschrift>
```

Ein Beispiel (2)

- XML-DTD:

```
<!ELEMENT anschrift (name, strasse,  
    ort) >
```

```
<!ELEMENT name          (#PCDATA) >
```

```
<!ELEMENT strasse      (#PCDATA) >
```

```
<!ELEMENT ort          (#PCDATA) >
```

XML Einführung (2)

- XML geeignet für die Ablage strukturierter Daten in eine Text-Datei
- DTD beschreibt strukturelle Gemeinsamkeiten gleichartiger Dokumente
- Trennung von Layout und Inhalt
- Erweiterbarkeit

Ziele und Defizite

- Einfachheit!
- SGML kompatibel
- Unterstützung für ein breites Spektrum von Anwendungen
- Dokumente von Menschen lesbar
- Knappheit ist von minimaler Bedeutung

Anwendungsgebiete

Bosak: „XML gives Java something to do“

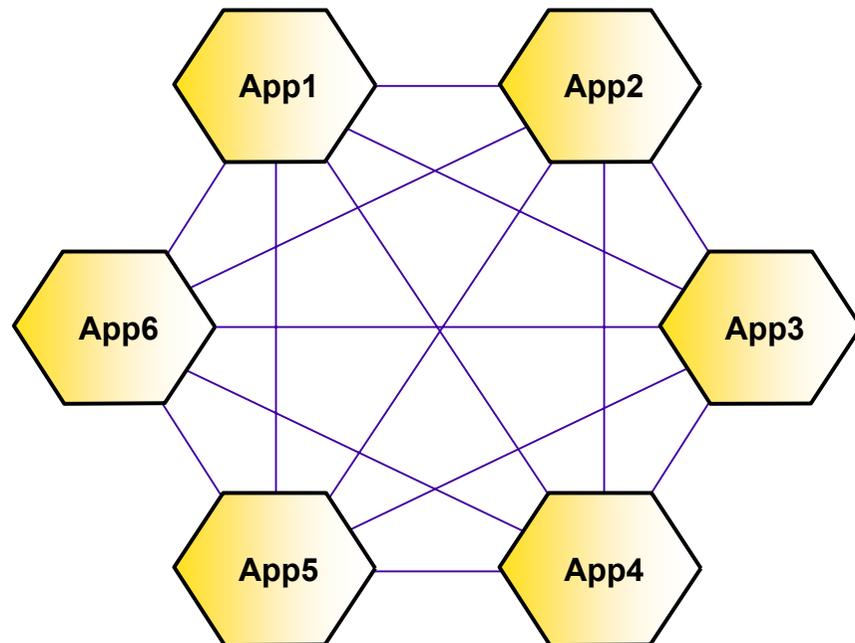
- XML als standardisiertes Datenaustauschformat
- Verteilung von Prozessor- und Netzlast vom Server zum Client
- Variable Darstellung von Informationen
- Agenten und Retrievalmöglichkeiten

Teil 2: XMI

Das XML Metadata Interchange Format

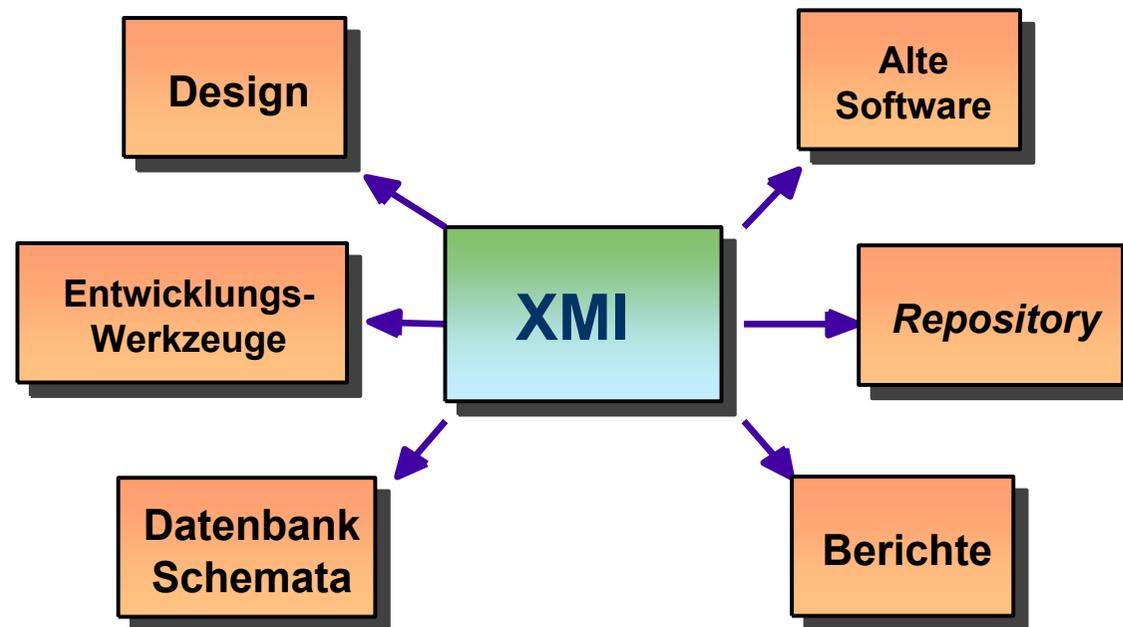
Das Problem

$N*(N-1) = 30$ Übergänge
 $N = 6$ Hersteller
Versionsprobleme!



Die Vision

6 Übergänge
6 Hersteller



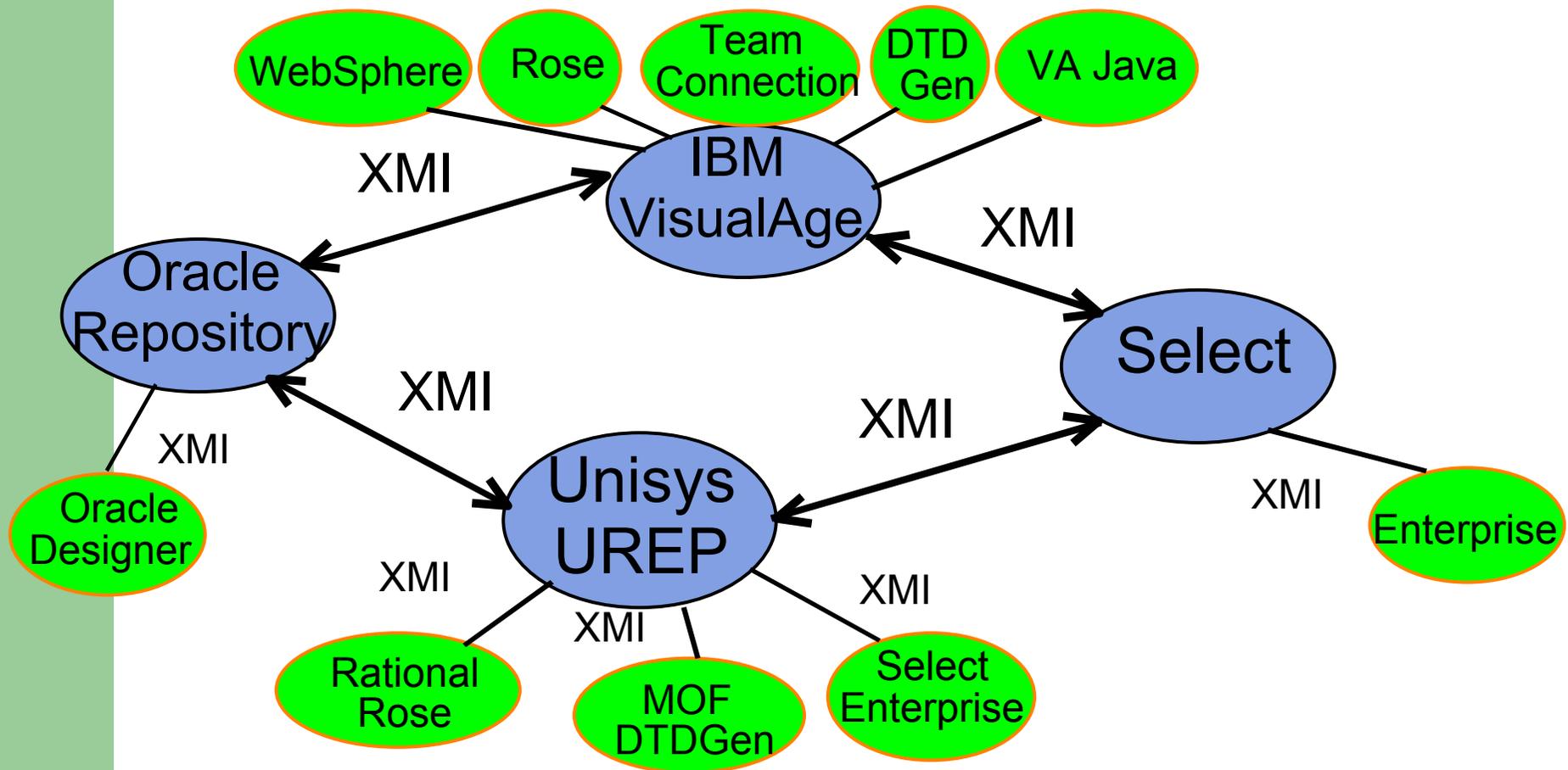
Die Ziele

- Industriestandard für Stream-based Model Interchange Format (SMIF)
- Generisches Format für eine Vielzahl von Modellen
- Demonstration der Nutzbarkeit
 - Mittels OAD - also UML – Modelltransfer
 - Anwendung auf MOF-gemäßen Metamodellen
- Herstellerunabhängige Transferformate

Geschichtliches

- 12/97 RFP für SMIF
- 07/98 Vorschläge (XMI, CDIF, UOL)
- 10/98 Überarbeiteter XMI Vorschlag
- 11/98 „Beweis des Konzepts“
- 01/99 OMG Standardisierung beginnt
- 04/99 XMI 1.0 verabschiedet
- 10/99 XMI 1.1
- 11/01 RFP für XMI 2.0

„Das“ Anwendungsbeispiel



XMI - Überblick

- Basiert auf XML
 - Spezifiziert XML-DTDs für
 - MOF basierte Metamodelle und deren Instanzen
 - UML basierte Modelle
 - Unicode, Internet, DB, ...
- Spezifiziert MOF nach XML Abbildung
 - MOF → XML-DTD
 - MOF → XML-Stream
 - (MOF → CORBA IDL)
- Benutzt UML für Metamodell Design

XML
Syntax und Kodierung

MOF
Metadaten Definitionen
& Management

UML
Metamodell
Analysis & Design

**X
M
I**

UML Modelle
CWM Modelle
MOF Meta Modelle

XML Ströme (Modelle)
(Viele – mit jeweils einer DTD)

UML 1.1 DTD
CWM DTD
MOF 1.1 DTD ...

XML DTD (Meta Modelle)
(1 DTD pro Metamodell zur Verifikation)

↑ Verifizieren

```

<!-- Dokument Prolog, etc. -->
<Model xmi.id="a1"> <name>Auto</name><visibility xmi.value="public"/>
  <ownedElement>
    <Class xmi. id="a7"><name>Motor</name>
      <feature>
        <Attribute><name>Zylinder</name>
          <multiplicity><XML.field>1</ XML.field>
            < XML.field>1</ XML.field></multiplicity>
          <type>< DataType href="|a247"/></type> <!-- Anzahl -->
        </Attribute>
        <Operation><name>anlassen</name>
          <scope xmi.value="instance"/>
        </Operation>
      </feature>
    </Class>
  </ownedElement>
</Model>

```



UML DTD Fragment

```
...
<!ELEMENT Class (name, visibility, isRoot, isLeaf, isAbstract, isActive, XMI.extension*,
    constraint*, requirement*, provision*, stereotype*, elementReference*,
    collaboration*, partition?, template?, view*, presentation*, namespace?,
    behavior*, binding? implementation*, generalization*, specialization*,
    parameter*, structuralFeature*, specification*, associationEnd*, participant*,
    createAction*, instance*, classifierRole*, realization*, classifierInState*,
    taggedValue*, ownedElement*, feature*)?>
<!ATTLIST Class XMI.element.att; XMI.link.att;>
<!ELEMENT name (#PCDATA | XMI.reference)*>
<!ELEMENT feature (Feature| StructuralFeature| Attribute| BehavioralFeature| Operation| Method
    Reception)*>
...
```

Tools

- IBM's (alphaWorks) XMI-Toolkit
 - Rational Rose 98 / XMI-UML → XMI-DTD
 - Rational Rose 98 → XMI-UML
 - Rational Rose 98 ↔ Java
- Leider: XMI-UML ≠ XMI-UML
- XML-Tools: JDK 1.4, Xerces, Xeena, XMLSpy, Emacs, ...

Gegenwart und Zukunft

- Freier Standard unterstützt von vielen Firmen
- Freie Tools
- Hauptanwendung: UML

- XML Schema
- MOF, CDIF und XMI wachsen weiter zusammen