

DAIMLERCHRYSLER

XML Schemata

Notwendigkeit, Ansätze, Standards

Mario Jeckle

DaimlerChrysler Forschungszentrum Ulm
Abt. Prozeßkette Produktentwicklung (FT3/EK)

mario.jeckle@daimlerchrysler.com

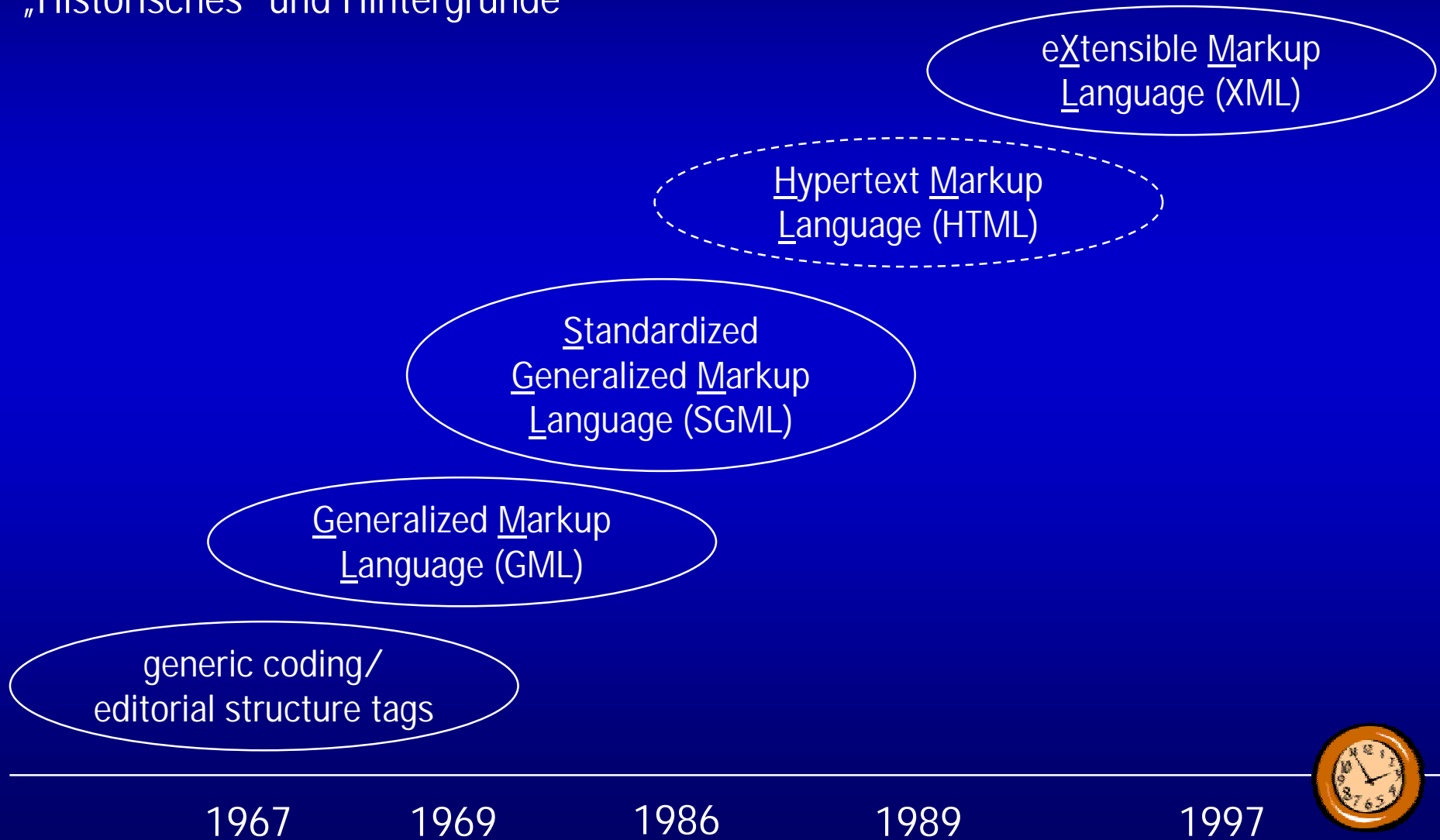
www.jeckle.de

Inhaltsübersicht

- Grundlegendes zur eXtensible Markup Language (XML)
 - Hintergrund: *Auszeichnungssprachen*
 - Dokument und *Document Type Definition* (DTD)
 - *well-formedness, valid, schema valid*
- Der DTD-Mechanismus in XML v1.0 und seine Grenzen
- Anforderungen an eine adäquate Schemaunterstützung
- Metasprachen als Ansatz
- Existierende Ansätze
- Standards; Einordnung und Bewertung -- Zukunftsaussichten
- Beispiel

Die *eXtensible Markup Language* (XML)

„Historisches“ und Hintergründe



Strukturelle Primitive eines XML-Dokuments

```
<person name="mario">  
  <xml-vortrag/>  
</person
```

Strukturelle Primitive eines XML-Dokuments

start tag

`<person name="mario">`

`<xml-vortrag />`

`</person`

Strukturelle Primitive eines XML-Dokuments

start tag

`<person name="mario">`

`<xml-vortrag />`

`</person`

end tag

Strukturelle Primitive eines XML-Dokuments

```
<person name="mario">  
  <xml-vortrag/>  
</person>
```

tag content

Strukturelle Primitive eines XML-Dokuments

```
<person name="mario">  
  <xml-vortrag/>  
</person
```

empty tag

Strukturelle Primitive eines XML-Dokuments

tag name

```
<person name="mario">  
  <xml-vortrag/>  
</person
```

Strukturelle Primitive eines XML-Dokuments

attribute name

<person name="mario">

<xml-vortrag/>

</person

Strukturelle Primitive eines XML-Dokuments

attribute

<person name="mario">

<xml-vortrag/>

</person

Strukturelle Primitive eines XML-Dokuments

attribute content

```
<person name="mario">  
  <xml-vortrag/>  
</person
```

Anwendung der eXtensible Markup Language (XML)

Ein XML-Dokument

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE ProjektAbwicklung SYSTEM "projekt.dtd">

<ProjektAbwicklung>
  <person persID='P100' gehaltsGrp="2">
    <vorname>Hans</vorname>
    <vorname>Georg</vorname>
    <name>Meier</name>
    <projektLeiter prjRef='F300' />
    <projektMitarb prjRef='F310' />
  </person>
  <projekt prjID='F300' start='1.1.2000' budget='1000' />
  <projekt prjID='F310' budget='10000' />
  <projekt prjID='F320' budget='25000' />
</ProjektAbwicklung>
```

```
<?xml version="1.0" encoding="UTF-8" ?>
```

```
<!ELEMENT      ProjektAbwicklung (person+, projekt*)>
<!ELEMENT      person (vorname+, name, projektLeiter?, projektMitarb*)>
<!ATTLIST      person
                persID          ID          #REQUIRED
                gehaltsGrp      (1|1a|2) "1">
<!ELEMENT      vorname          (#PCDATA)>
<!ELEMENT      name              (#PCDATA)>
<!ELEMENT      projektLeiter     EMPTY>
<!ATTLIST      projektLeiter
                prjRef          IDREF      #REQUIRED>
<!--
<!ELEMENT      projektMitarb     EMPTY>
<!ATTLIST      projektMitarb
                prjRef          IDREF      #REQUIRED>
-->
<!ELEMENT      projekt           EMPTY>
<!ATTLIST      projekt
                prjID          ID          #REQUIRED
                start          CDATA      #IMPLIED
                budget         CDATA      #REQUIRED>
```

Der DTD-Mechanismus in XML v1.0

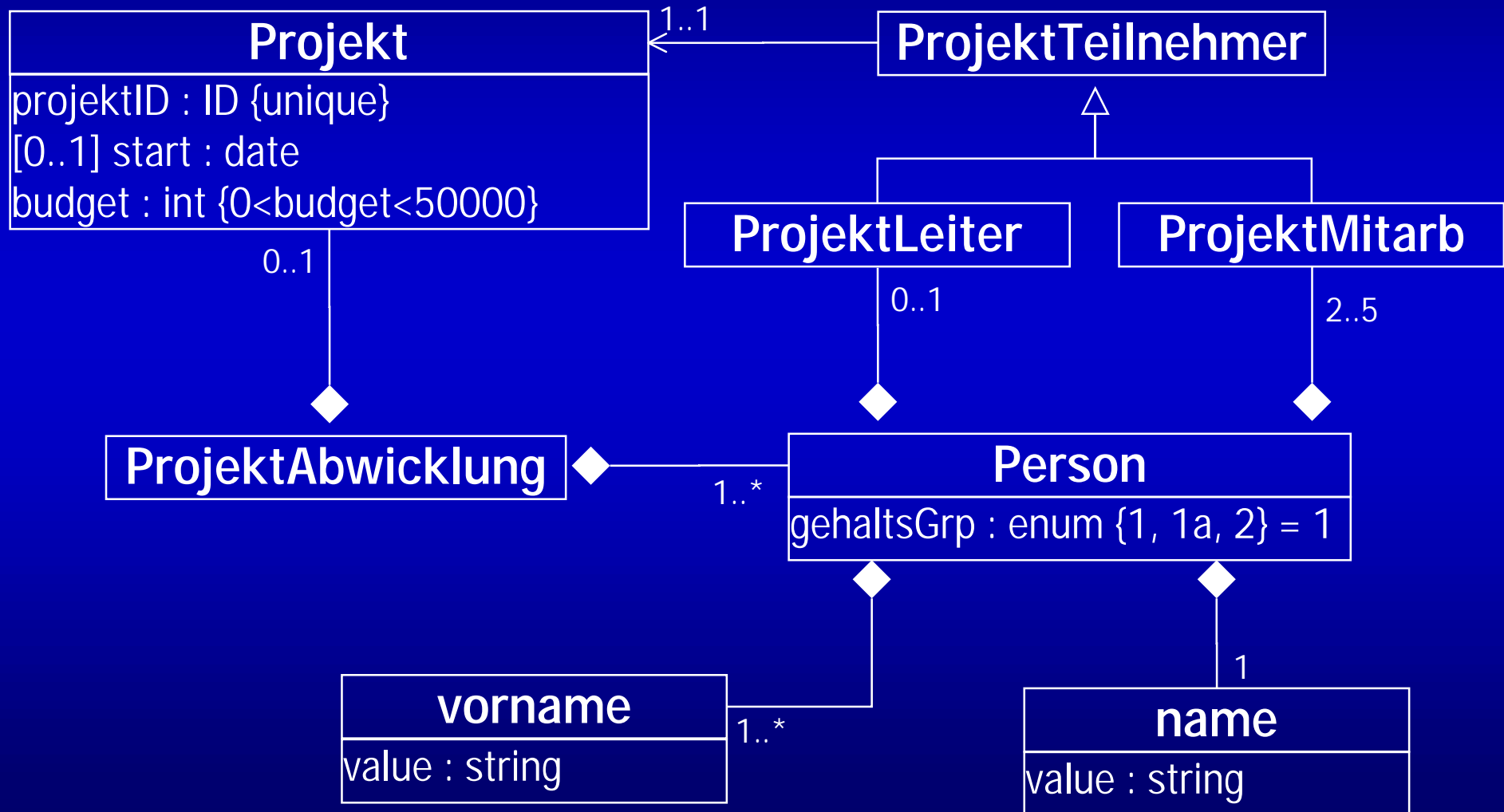
- streng hierarchisch
- *ELEMENTs* als innere Knoten
- *ATTLISTs* zur Attributierung der Knoten
- Keine Datentypen (abgesehen von CHAR-Data)
- Rudimentärer Referenzierungsmechanismus (ID, IDREF)
- Selektionstyp
- Vorgabewerte
- DTD ist nicht XML

Proprietäre Realisierung weiterer Validierungsaufgaben im Parser

Vereinfachte Grammatik eines XML-Dokuments

Document	->	rootElement
rootElement	->	(Element)*
Element	->	'<' Name (Attribute)* '>' Content '</Name '>' '<' Name (Attribute)* '/>'
Content	->	(0-9 a-z A-Z '_' ':' '.' '-')*
Attribute	->	Name '=' '"' Content '"' Name '(' Name (' ' Name)* ')' ('"' Name '"')?
Name	->	NameStart NameChar
NameStart	->	(a-z A-Z) '_' ':'
NameChar	->	NameStart (0-9) '.' '-'

Beispiel



XML-Schema Anforderungen I (Strukturell)

1. Beschreibung der Dokumentstruktur
(namespaces, elements, attributes) und des Dokument-Inhalts
(datentypen, entities, notations)
2. Vererbung zwischen
elements, attributes und Datentypen
3. URI Referenzierung
4. Dokumentationsmöglichkeit im Schema
5. Applikationsspezifische Einschränkungen und Beschreibungen
6. Schemaevolution
7. Integration von Schemata mit Primitivtypen

XML-Schema Anforderungen II (Datentypen)

1. Primitivtypen

byte, date, integer, sequence

SQL- und Java-Primitivtypen

Binärdaten

2. Typsystem adäquat für Im-/Export von DB-Inhalten

(OLAP-Systeme)

3. Unterscheidung zwischen Einschränkungen an lexikalische Repräsentation und Constraints der Informationsstruktur

4. User-defined Types

durch Vererbung und Einschränkung

Lösungsansätze

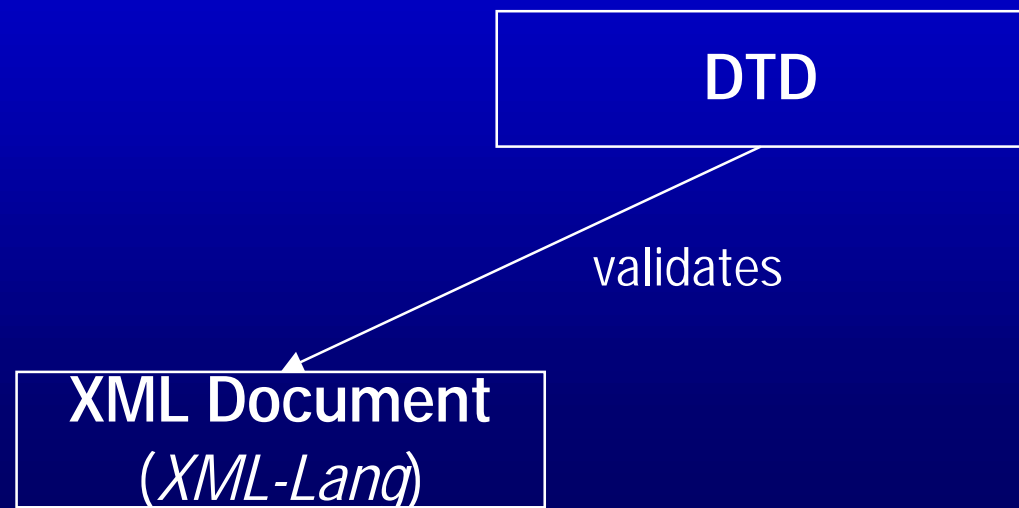
Erweiterung des bestehenden DTD-Mechanismus:

- Data Types for DTDs
- Document Content Description for XML
RDF basierte Semantikbeschreibung
setzt auf *XML-Data (Reduced)* auf
- Document Definition Markup Language/XSchema
- Schema for Object-Oriented XML

	<i>Document Type Definition</i>	<i>XML-Schema Anforderungen</i>
<i>Problem</i>		
Datentypen	<ul style="list-style-type: none"> ■ CDATA; Rest: Applikationslogik 	<ul style="list-style-type: none"> ■ Deklaration im Schema ■ Validierung mit generischem Standardparser
Strukturen	<ul style="list-style-type: none"> ■ Streng hierarchisch 	<ul style="list-style-type: none"> ■ Netzartig ■ Vererbung
Wiederverwendung	<ul style="list-style-type: none"> ■ externe Entities 	<ul style="list-style-type: none"> ■ Importmechanismus ■ Referenzierung
Reverse-Engineering	<ul style="list-style-type: none"> ■ proprietäre Zugriffsstrukturen ■ keine (std.) Validierung 	<ul style="list-style-type: none"> ■ XML-Dokument <ul style="list-style-type: none"> ■ Standardparser ■ Syntaxgesteuertes Editing

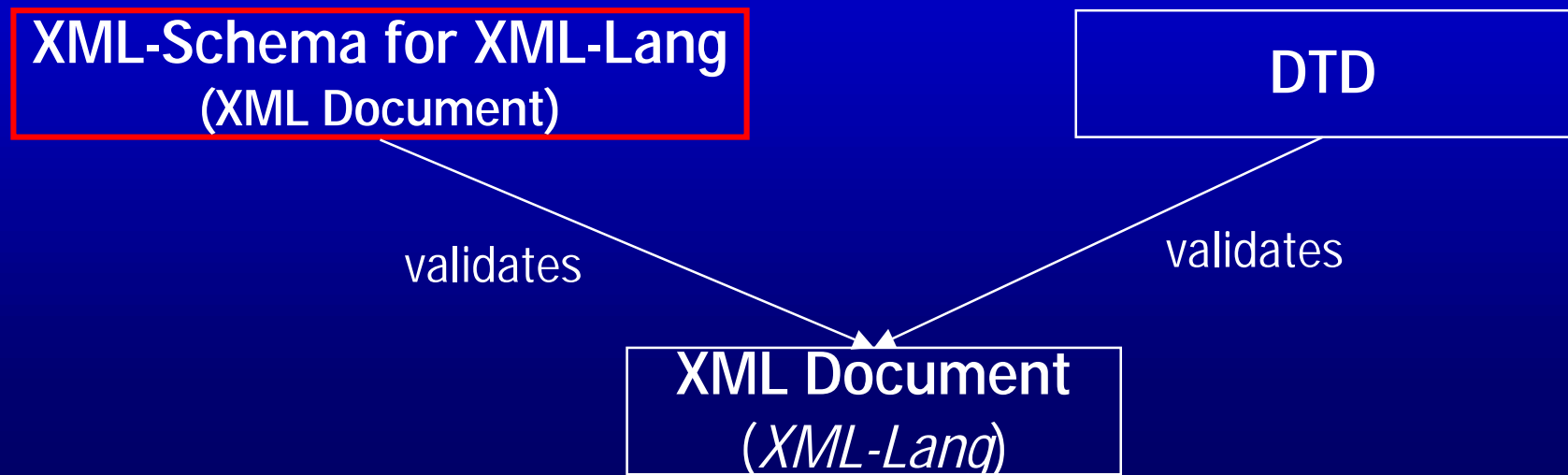
Schemasprachen -- Ansätze und Vorschläge

Metamodellierung



Schemasprachen -- Ansätze und Vorschläge

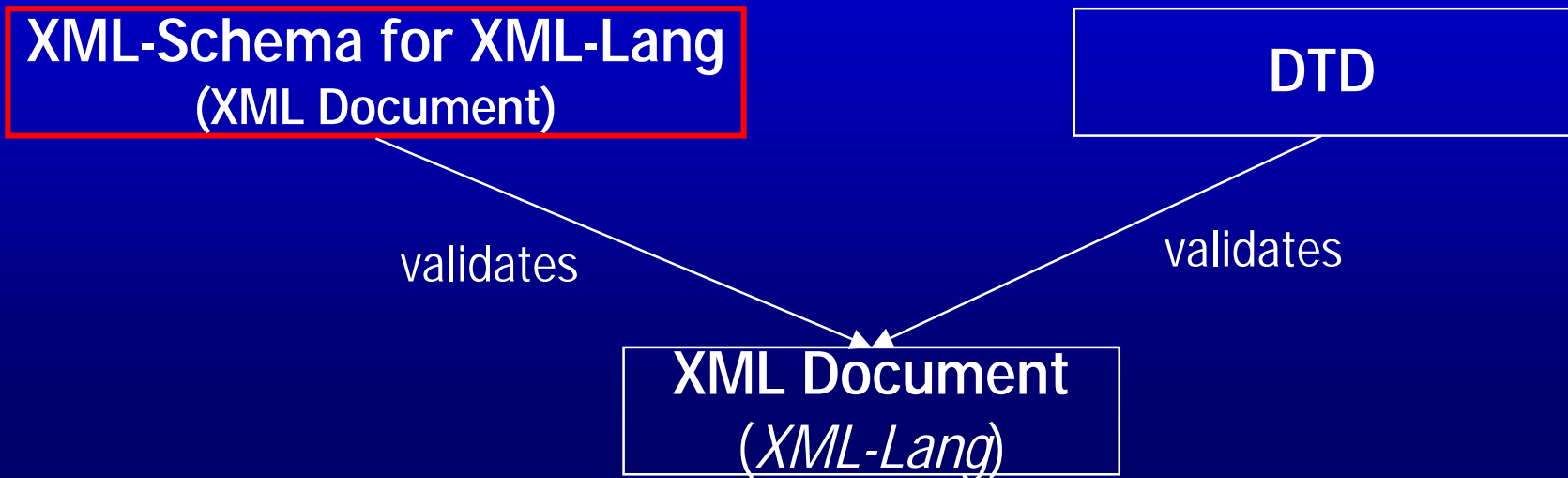
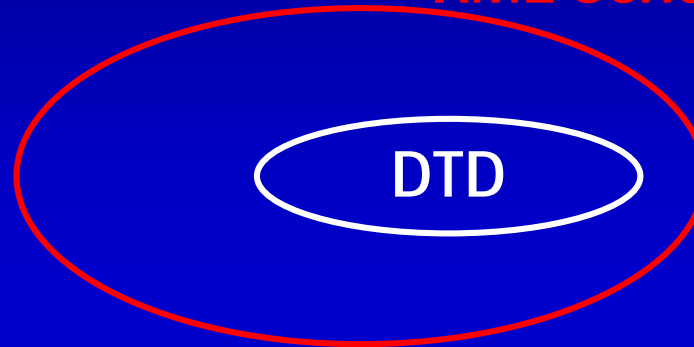
Metamodellierung



Schemasprachen -- Ansätze und Vorschläge

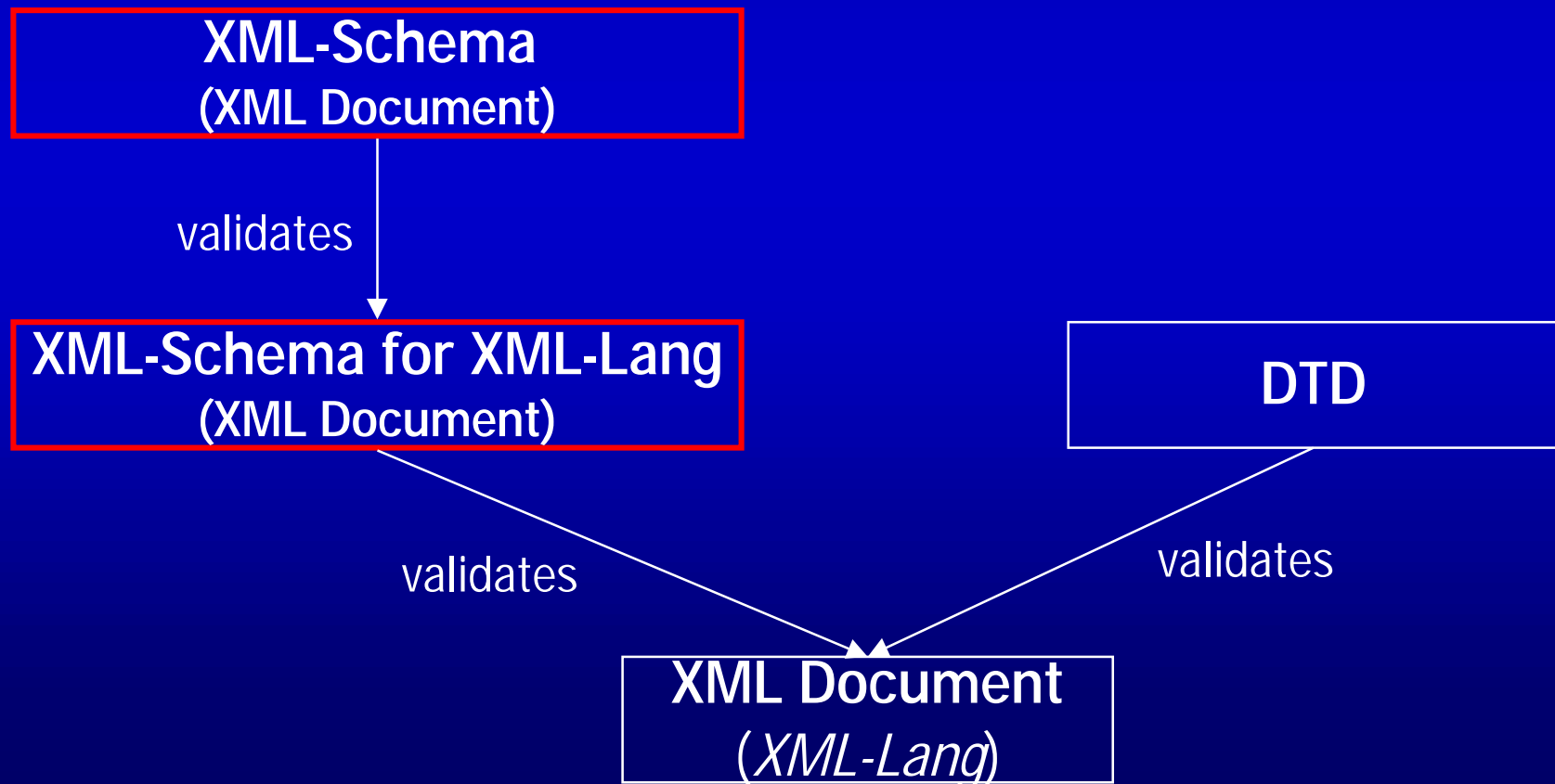
Metamodellierung

XML Schema



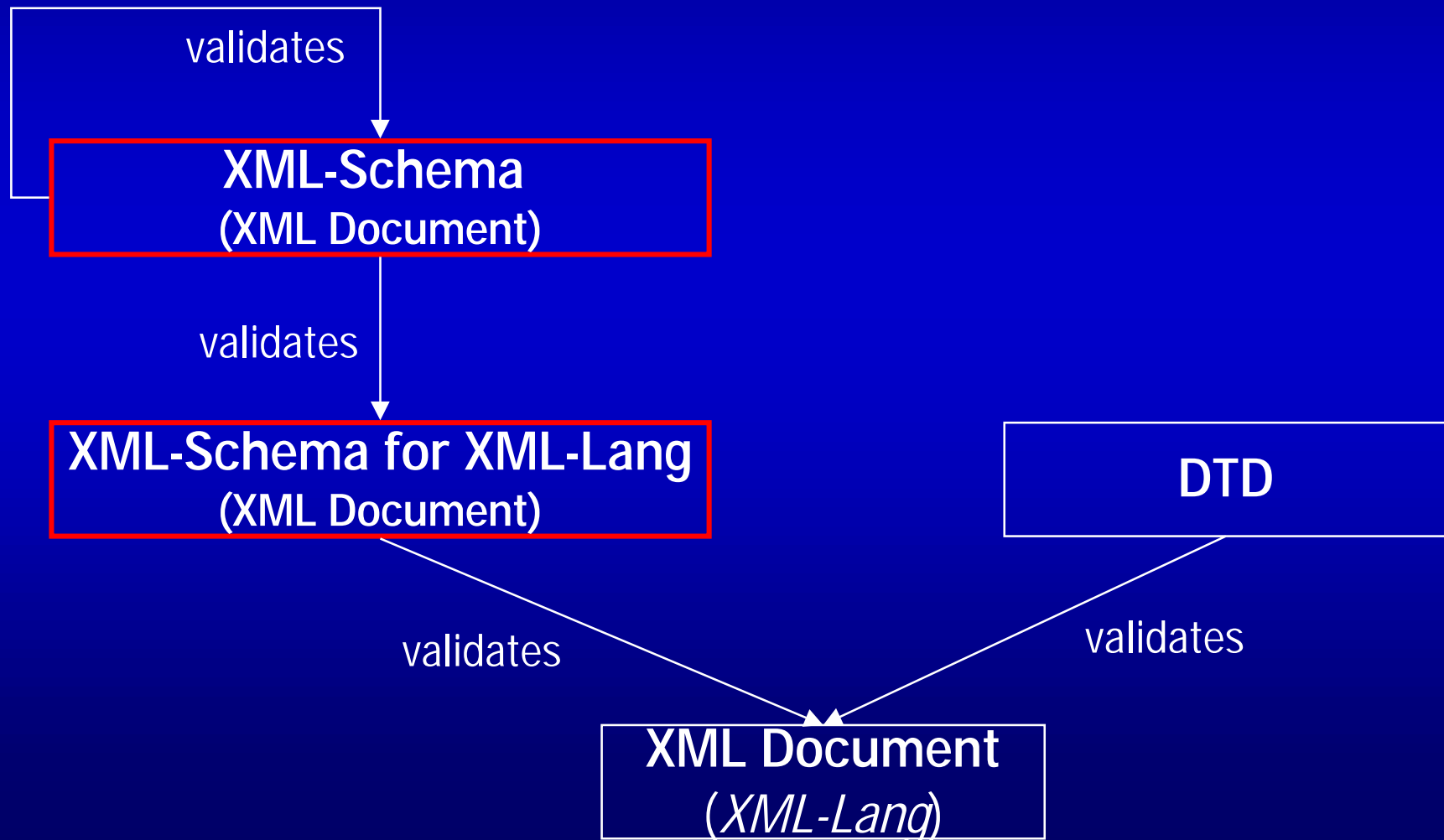
Schemasprachen -- Ansätze und Vorschläge

Metamodellierung



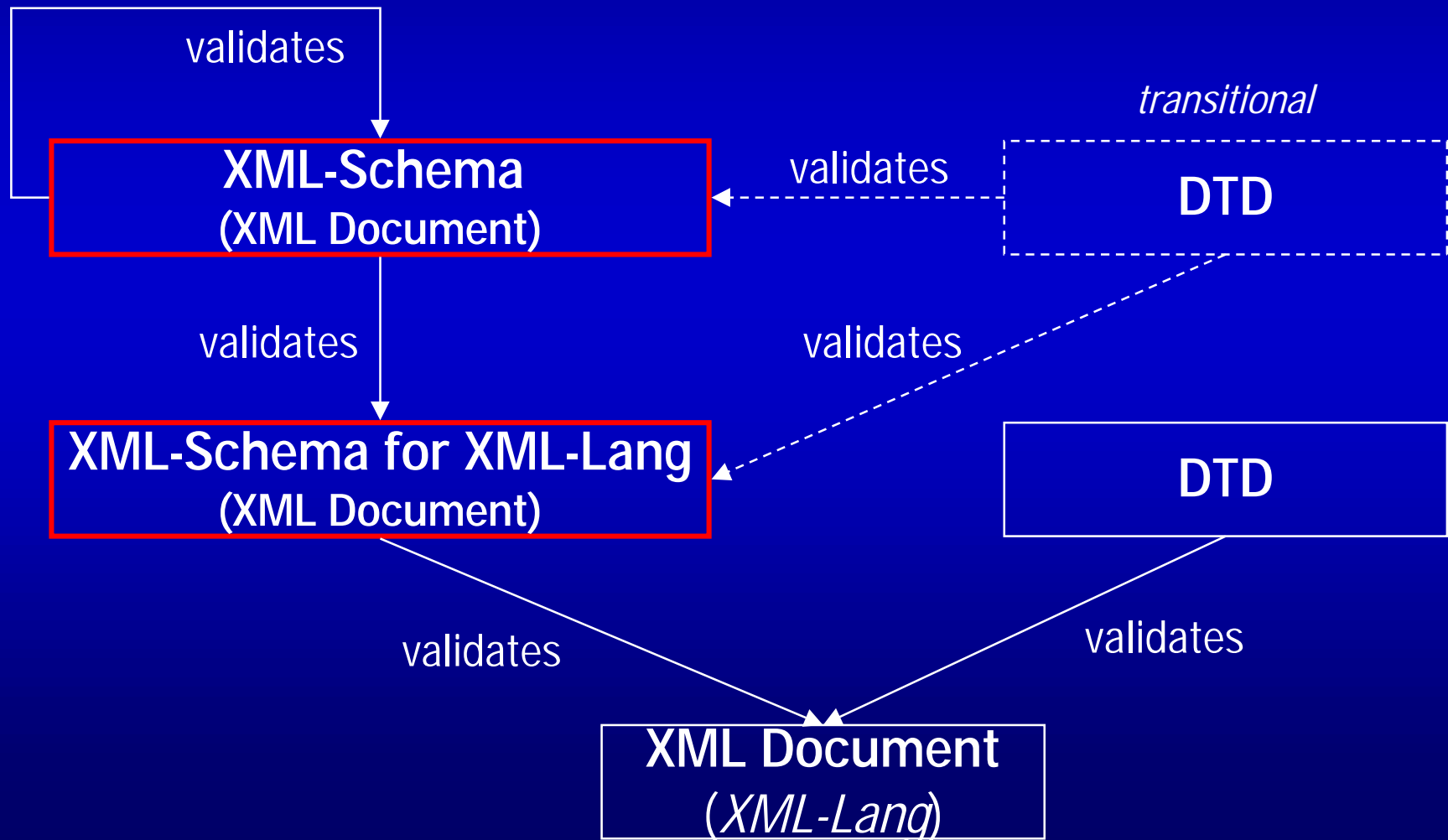
Schemasprachen -- Ansätze und Vorschläge

Metamodellierung



Schemasprachen -- Ansätze und Vorschläge

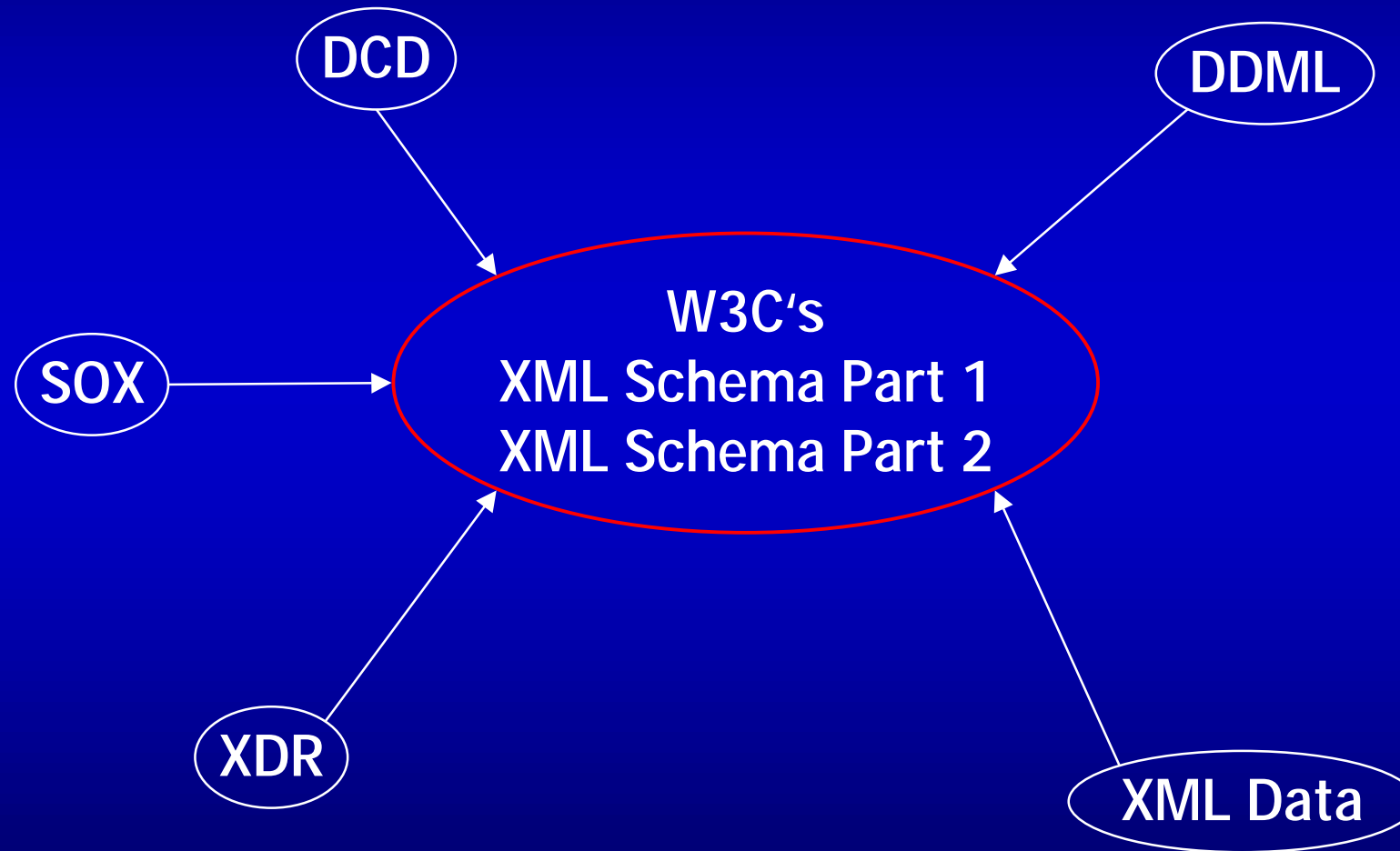
Metamodellierung



Übersicht existierender Ansätze

Document Content Description	Note	Textuality, Microsoft, IBM
Document Definition Markup Language	Note	TU Darmstadt, GMD, xml-dev
Schema for Object-Oriented XML	Note	diverse
XML Data	Note	Microsoft, ArborText, U Edinburgh, DataChannel, Inso Corp.
Document Structure Description	./.	AT&T, U Aarhus (Dmk)
Schematron	./.	R. Jelliffe

Existierende Ansätze



W3C's XML Schema Part 1 & Part 2 (working draft)

Gegenüber DTD ...

- stark erweiterte build-in Typen
- Erweiterbares Typsystem
 - Typextension
 - Typrestriktion
 - lexikalische Definition
- Typbeziehungen (äquivalente Typen)
- „echte“ XML-Sprache
- Explizite Beziehungskardinalität
- Kontextabhängige Eindeutigkeit
- Elemente gleichen Namens mit verschiedenen Inhalten

Umsetzung des Beispiels mit W3C's XML Schema (Typedefinition)

Projekt
projektID : ID {unique}
[0..1] start : date
budget : int {0 < budget < 50000}

```
<type name="projektType" content="empty">  
  <attribute name="prjID" type="ID" minOccurs="1"/>  
  <attribute name="start" type="date"  
    minOccurs="0" maxOccurs="1"/>  
  <attribute name="budget" type="budget" minOccurs="1"/>  
</type>
```

Umsetzung des Beispiels mit W3C's XML Schema (Datentypdefinition)

Projekt
projektID : ID {unique}
[0..1] start : date
budget : int {0 < budget < 50000}

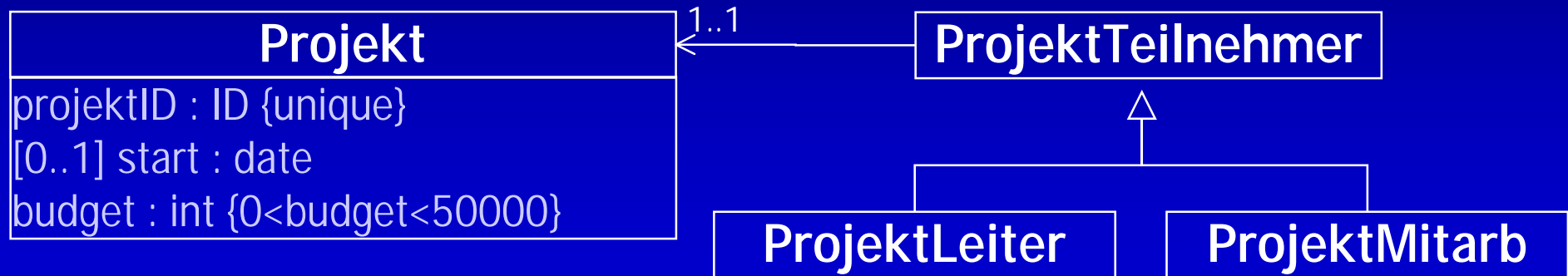
```
<datatype name="budgetType" source="integer">  
  <minExclusive value="0"/>  
  <maxExclusive value="50000"/>  
</datatype>  
...  
<attribute name="budget" type="budget" minOccurs="1"/>  
...
```


Umsetzung des Beispiels mit W3C's XML Schema (Enumeration)

Person
gehaltsGrp : enum {1, 1a, 2} = 1

```
<datatype name="gehaltsGrpType" source="string">
  <enumeration value="1"/>
  <enumeration value="1a"/>
  <enumeration value="2"/>
</datatype>
<type name="personType">
...
  <attribute name="gehaltsGrp"
    type="gehaltsGrpType" default="1"/>
</type>
```

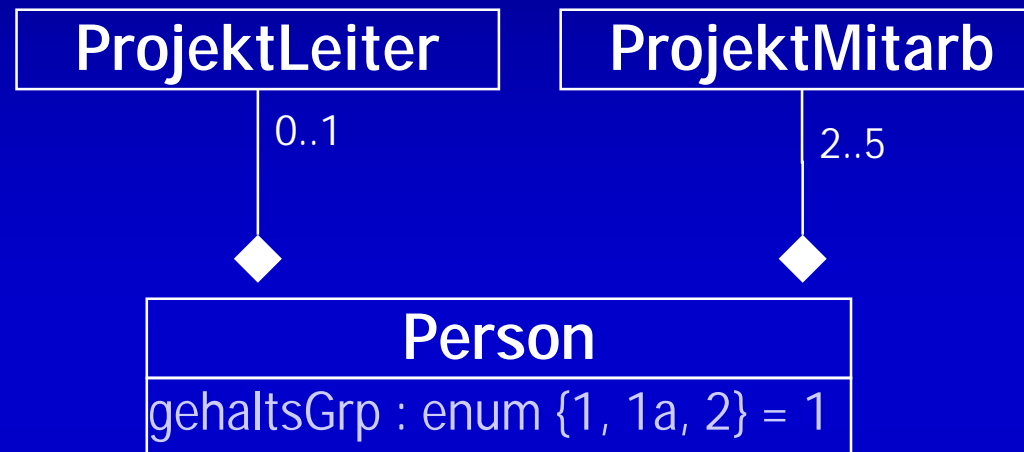
Umsetzung des Beispiels mit W3C's XML Schema (Typvererbung)



```

<type name="projektTeilnehmerType" abstract="true"
  content="empty">
  <attribute name="prjRef" type="IDREF" minOccurs="1"/>
</type>
<type name="projektLeiterType" content="empty"
  source="projektTeilnehmerType" derivedBy="extension"/>
<type name="projektMitarbType" content="empty"
  source="projektTeilnehmerType" derivedBy="extension"/>
    
```

Umsetzung des Beispiels mit W3C's XML Schema (Beziehungskardinalitäten)



```
<type name="personType">
```

...

```

<element name="projektLeiter" minOccurs="0" maxOccurs="1"/>
<element name="projektMitarb" minOccurs="2" maxOccurs="5"/>
<attribute name="persID" type="ID" minOccurs="1"/>
<attribute name="gehaltsGrp" type="gehaltsGrpType" default="1"/>

```

```
</type>
```

Referenzen

XML Schema Requirements: www.w3.org/TR/NOTE-xml-schema-req

XML-Schema Part 1: www.w3.org/TR/xmlschema-1

XML-Schema Part 2: www.w3.org/TR/xmlschema-2

Document Content Description: www.w3.org/TR/NOTE-dcd

Document Definition Markup Language: www.w3.org/TR/NOTE-ddml

XSchema: www.simonstl.com/xschema/spec/xscspecv4.htm

Schema for Object-oriented XML: www.w3.org/TR/NOTE-SOX

XML-Data: www.w3.org/TR/1998/NOTE-XML-data

Document Structure Description: www.brics.dk/DSD

Schematron:

www.ascc.net/xml/resource/schematron/schematron.html