

Entwurf von XML-Sprachen mit XML-Schema

Mario Jeckle *

DaimlerChrysler Forschungszentrum Ulm
mario.jeckle@daimlerchrysler.com
mario@jeckle.de
www.jeckle.de

Abstract

Das weltumspannende Netz bietet ein enormes Potential hinsichtlich schnellem und ungehindertem Datenaustausch. Staatliche Grenzen als limitierender Faktor globalen Informations- und Warenaustauschs treten ebenso wie das klassische Produzenten-Erzeugerschema in den Hintergrund.

Die sich abzeichnende Informationsgesellschaft, flankiert durch neue ökonomische Paradigmen, ermöglicht die aktive Teilhabe jedes Netzteilnehmers.

Die durch Tim Berners-Lee definierte Hypertextsprache *HTML* läutete vor eineinhalb Jahrzehnten den Siegeszug des World-Wide-Web ein. Waren zu Anfang der 1990er Jahre Computernetzwerke noch überwiegend technisch versierten *Geeks* bekannt und vorbehalten, so hat das WWW weite Teile der westlichen Industriegesellschaft nachhaltig verändert. Die Informationsrecherche und das Anbieten eigener Information auf dem entstandenen globalen Marktplatz hat Einzug in den beruflichen und privaten Alltag vieler Menschen gehalten.

In der Konsequenz entstanden Internet-Kataloge und *Suchmaschinen*, Anwendungen zur gezielten Informationsrecherche. Mit der Erschließung des entstandenen – offenkundig schrankenlosen – Marktes treten jedoch die Grenzen des rein präsentationsorientierten Informationsangebots zu Tage. So katalogisieren selbst die „besten“ (im Sinne von größten) Verzeichnisse nur einen Bruchteil der existierenden Internetseiten. Vielmehr noch, ist nicht einmal die tatsächliche Anzahl der verfügbaren Seiten bekannt. (Ein Zustand, der übertragen auf das nachgerade klassisch anmutende Kommunikationsmedium Telephon, unhaltbar erscheint – im Netz jedoch als gegeben und unveränderbar hingenommen wird).

Die rasch entstandenen Volltextindizes, welche mit dem Anspruch der Erfassung jeder erreichbaren Textpassage aufwarteten, leiden noch viel mehr als die redaktionell betreuten Verzeichnisse an der Informationsflut. So liefern einzelne Suchanfragen durchaus jenseits von einer Million Treffern, welche unkategorisiert und unbewertet der Bearbeitung durch den Suchenden harren.

Die aus der *SGML for the Web* entstandene Metasprache *eXtensible Markup Language* (XML) schickt sich nun an, diesen Mißstand beheben zu helfen. Als eine Sprache zur Sprachdefinition soll XML die verschiedenen Informationsanbieter in der einheitlichen Strukturierung ihrer Daten unterstützen. Hierzu definiert XML eine überschaubare Menge von einfachen Basiskonzepten, mittels dieser sich beliebige Informationsstrukturen formalisieren lassen.

Daraus resultiert auch das Potential von XML in Bezug auf Web-Verarbeitung. Insbesondere in Webindices können Suchalgorithmen zusätzlich zu der bestehenden Inhaltssuche hinsichtlich klassifizierender Information, ausgedrückt durch XML-Tags, angereichert werden.

Die derzeit größte zu bewältigende Herausforderung im praktischen Einsatz der *eXtensible Markup Language* ist die Entwicklung problemadäquater XML-Sprachen. Zusätzlich werfen Qualitätsaspekte wie Wiederverwendbarkeit und flexible Adaptierbarkeit an sich verändernde Gegebenheiten weitere Fragestellungen in diesem Kontext auf.

1 Schemadialekte – Konkurrenz und Konfluenz

Bereits kurz nach der Verabschiedung der XML-Recommendation wurde durch Microsoft im März 1998 unter dem Titel *XML-Data* ein erster Vorschlag für eine XML-Schemasprache eingereicht. Bereits dieser erste Wurf enthielt in Grundzügen die technischen Eckpunkte der aktuellen Diskussion. So ist XML-Data selbst eine XML-Sprache; mithin die Schemasprache selbstbeschreibend. Ebenso sind in diesem Papier bereits Ansätze für ein Typsystem enthalten. Allerdings stieß der Vorschlag zur Definition (noch) auf keine breite Zustimmung innerhalb der W3C-Arbeitsgruppen. Zunächst sollte die weitere Entwicklung und Akzeptanz seitens des Marktes der Metasprache selbst beobachtet werden.

Parallel zu den ersten XML-Sprachdefinitionen offenbarten sich die für datenintensive Applikationen immensen Einschränkungen der von SGML übernommenen Document Type Definitions. Waren ihre Ausdrucksmöglichkeiten durchaus hinreichend für die dokumentenorientierten Strukturen der (SGML-)Vergangenheit, so stellt die de facto Abwesenheit programmiersprachennaher Datentypen und mangelnde Unterstützung komplexerer Strukturierungsprimitiven eine deutliche Einschränkung dar.

Inzwischen sind eine breite Palette verschiedenster zueinander in Konkurrenz stehender Schemasprachen entwickelt worden, und per Web frei verfügbar. Gemäß den beschrifteten Ansätzen lassen sie sich in vier Kategorien einteilen:

- Orientierung am bestehenden DTD-Mechanismus
Erweiterungen des bestehenden Mechanismus um zusätzliche Sprachelemente.
- Orientierung an Wissensdarstellungen
Interpretation des Schemas einer XML-Sprache als Wissen über die Sprache.

- Orientierung an der programmiersprachlichen Umsetzung
Versuch XML und ein Ausführungsmodell möglichst eng zu koppeln.
- XML-Sprachen zur Inhaltsbeschreibung
Per definitionem kann XML zur Beschreibung beliebigster Informationen im Allgemeinen herangezogen werden, daher ist XML auch im Speziellen für XML-Strukturen als Beschreibungsmechanismus denkbar.

Die bei der Erweiterung des existierenden DTD-Mechanismus entstehenden beiden Freiheitsgrade – Erweiterung in Richtung SGML-Mächtigkeit, und Erweiterung um neue, d.h. nicht in SGML enthaltene Elemente – sind beide technisch möglich und gleichermaßen problembehaftet. Während die erste Option XML sukzessive der Mächtigkeit von SGML annähert, ohne letztlich über diese hinauszugehen, favorisiert die letztere Alternative die Aufgabe des XML-Grundprinzips der SGML-Kompatibilität. Mithin wären die so entstehenden XML-Sprachen nicht mehr als SGML-Dokumente interpretierbar.

Ausgehend von den Wissensdarstellungen wurde bereits durch das W3C das *Resource Description Framework* (RDF) verabschiedet, um eine einheitliche Sprache zur XML-basierten Formalisierung beliebiger Wissenstrukturen und Ontologien zu schaffen. So vielversprechend solche Ansätze sind, negieren sie jedoch die *datenorientierte* Auslegung von XML, und vergrößern in der Konsequenz den *impedance mismatch* bei der Integration von XML-Sprache und Anwendungsprogrammierung.

Daher liegt das Extremum der möglichst engen Kopplung von Programmiersprache und Dokumentenstruktur nahe. Jedoch kranken auch diese Ansätze an den verschiedenen Abstraktionsebenen von Programm und Dokument. Wird das Dokumentschema aus dem Programmiermodell abgeleitet, so verbieten sich naturgemäß Dokumentstrukturen, für die kein programmiersprachliches Äquivalent existiert. In der Folge würde die XML-Entwicklung deutlich ihrer Dynamik beraubt, und eng an die verfügbaren Programmiersprachen gekoppelt.

Die größte Bedeutung nehmen – schon allein – quantitativ, und auch gemessen an der Resonanz der Anwender und Entwickler, die XML-basierten Schemasprachen ein. Über ein Dutzend Varianten dieser Sprachklasse stehen zur Verfügung. Bei näherer Betrachtung offenbart sich diese, zunächst Augenscheinlich als Konkurrenz anmutende, Situation als breite Übereinkunft hinsichtlich der eingesetzten Mittel und unterstützten Grundkonzepte. Durch nähere Analysen der Sprachen läßt sich zeigen, daß sie in großen Bereichen äquivalent sind, wodurch sogar eine (nahezu) verlustlose Transformation ineinander möglich wird.

Auch die *XML Schema Description Language* (XSD), die derzeit durch das W3C erarbeitet wird, gehört zu diesem Sprachtyp. Zu ihrer Entwicklung tragen die wichtigsten und anerkanntesten Schemadialekte bei. Es ist damit zu rechnen, daß sich die XSD bereits im kommenden Jahr auf breiter Marktfrent gegenüber den DTDs durchsetzen wird; und diese langfristig im Umfeld datenorientierter Dokumentstrukturen ersetzen wird.

2 Verschiedene Schema-Codierungen des Beispiels

2.1 Document Type Definiton (DTD)

```
<!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 >
```

2.2 BizTalk

```

<?xml version = "1.0"?>
<Schema name = ""
  xmlns = "urn:schemas-microsoft-com:xml-data"
  xmlns:dt = "urn:schemas-microsoft-com:datatypes">
<ElementType name = "ProjektAbwicklung" content = "eltOnly" order = "seq">
<element type = "person" minOccurs = "1" maxOccurs = "*" />
<element type = "projekt" minOccurs = "0" maxOccurs = "*" />
</ElementType>

<ElementType name = "person" content = "eltOnly" order = "seq">
<AttributeType name = "persID" dt:type = "id" required = "yes" />
<AttributeType name = "gehaltsGrp" dt:type = "enumeration"
dt:values = "1 1a 2" default = "1" />
<attribute type = "persID" />
<attribute type = "gehaltsGrp" />
<element type = "vorname" minOccurs = "1" maxOccurs = "*" />
<element type = "name" />
<element type = "projektLeiter" minOccurs = "0" maxOccurs = "1" />
<element type = "projektMitarb" minOccurs = "0" maxOccurs = "*" />
</ElementType>

<ElementType name = "vorname" content = "textOnly" />
<ElementType name = "name" content = "textOnly" />
<ElementType name = "projektLeiter" content = "empty">
<AttributeType name = "prjRef" dt:type = "idref" required = "yes" />
<attribute type = "prjRef" />
</ElementType>

<ElementType name = "projektMitarb" content = "empty">
<AttributeType name = "prjRef" dt:type = "idref" required = "yes" />
<attribute type = "prjRef" />
</ElementType>

<ElementType name = "projekt" content = "empty">
<AttributeType name = "prjID" dt:type = "id" required = "yes" />
<AttributeType name = "start" dt:type = "string" />
<AttributeType name = "budget" dt:type = "string" required = "yes" />
<attribute type = "prjID" />
<attribute type = "start" />
<attribute type = "budget" />
</ElementType>
</Schema>

```

2.3 Document Content Description (DCD)

```

<?xml version = "1.0"?>
<DCD xmlns:RDF = "http://www.w3.org/1999/02/22-rdf-syntax-ns#">
<ElementDef Type = "ProjektAbwicklung" Content = "Closed"
  Model = "Elements">
  <Group RDF:Order = "Seq">
  <Group Occurs = "OneOrMore">
  <Element>person</Element>
  </Group>
  <Group Occurs = "ZeroOrMore">
  <Element>projekt</Element>
  </Group>
  </Group>
</ElementDef>

<ElementDef Type = "person" Content = "Closed" Model = "Elements">
<AttributeDef Name = "persID" Datatype = "id"
  ID-ROLE = "ID" Occurs = "Required"/>
<AttributeDef Name = "gehaltsGrp" Datatype = "enumeration"
  Default = "1">
<Values>1 1a 2</Values>
</AttributeDef>
<Group RDF:Order = "Seq">
<Group Occurs = "OneOrMore">
<Element>vorname</Element>
</Group>
<Element>name</Element>
<Group Occurs = "Optional">
<Element>projektLeiter</Element>
</Group>
<Group Occurs = "ZeroOrMore">
<Element>projektMitarb</Element>
</Group>
</Group>
</ElementDef>

<ElementDef Type = "vorname" Content = "Closed" Model = "Data"/>
<ElementDef Type = "name" Content = "Closed" Model = "Data"/>
<ElementDef Type = "projektLeiter" Content = "Closed" Model = "Empty">
<AttributeDef Name = "prjRef" Datatype = "idref"
  ID-ROLE = "IDREF" Occurs = "Required"/>
</ElementDef>

```

```
<ElementDef Type = "projektMitarb" Content = "Closed" Model = "Empty">
  <AttributeDef Name = "prjRef" Datatype = "idref"
    ID-ROLE = "IDREF" Occurs = "Required"/>
</ElementDef>

<ElementDef Type = "projekt" Content = "Closed" Model = "Empty">
  <AttributeDef Name = "prjID" Datatype = "id"
    ID-ROLE = "ID" Occurs = "Required"/>
  <AttributeDef Name = "start"/>
  <AttributeDef Name = "budget" Occurs = "Required"/>
</ElementDef>

</DCD>
```

2.4 Document Markup Language (DML)

```

<?xml version = "1.0"?>
<!DOCTYPE DocumentDef [
]>
<DocumentDef name = ""
  xmlns = "http://www.purl.org/NET/ddml/v1"
  xmlns:DDML = "http://www.purl.org/NET/ddml/v1" Version = "1.0">
<ElementDecl Name = "ProjektAbwicklung">
<Model>
<Seq>
<Ref Element = "person" Frequency = "OneOrMore"/>
<Ref Element = "projekt" Frequency = "ZeroOrMore"/>
</Seq>
</Model>

</ElementDecl>
<ElementDecl Name = "person">
<Model>
<Seq>
<Ref Element = "vorname" Frequency = "OneOrMore"/>
<Ref Element = "name"/>
<Ref Element = "projektLeiter" Frequency = "Optional"/>
<Ref Element = "projektMitarb" Frequency = "ZeroOrMore"/>
</Seq>
</Model>
<AttGroup>
<AttDef Name = "persID" Type = "ID" Required = "Yes"/>
<AttDef Name = "gehaltsGrp" Type = "Enumeration" AttValue = "1">
<Enumeration>
<EnumerationValue Value = "1"/>
<EnumerationValue Value = "1a"/>
<EnumerationValue Value = "2"/>
</Enumeration>
</AttDef>
</AttGroup>

<AttDef Name = "persID" Type = "ID" Required = "Yes"/>
<AttDef Name = "gehaltsGrp" Type = "Enumeration" AttValue = "1">
<Enumeration>
<EnumerationValue Value = "1"/>
<EnumerationValue Value = "1a"/>
<EnumerationValue Value = "2"/>

```

```
</Enumeration>
</AttDef>

</ElementDecl>
<ElementDecl Name = "vorname">
  <Model>
  <PCData/>
  </Model>

</ElementDecl>
<ElementDecl Name = "name">
  <Model>
  <PCData/>
  </Model>

</ElementDecl>
<ElementDecl Name = "projektLeiter">
  <Model>
  <Empty/>
  </Model>
  <AttGroup>
  <AttDef Name = "prjRef" Type = "IDRef" Required = "Yes"/>
  </AttGroup>

  <AttDef Name = "prjRef" Type = "IDRef" Required = "Yes"/>
  </ElementDecl>
  <ElementDecl Name = "projektMitarb">
  <Model>
  <Empty/>
  </Model>
  <AttGroup>
  <AttDef Name = "prjRef" Type = "IDRef" Required = "Yes"/>
  </AttGroup>

  <AttDef Name = "prjRef" Type = "IDRef" Required = "Yes"/>
  </ElementDecl>
  <ElementDecl Name = "projekt">
  <Model>
  <Empty/>
  </Model>
  <AttGroup>
  <AttDef Name = "prjID" Type = "ID" Required = "Yes"/>
  <AttDef Name = "start"/>
```

```
<AttDef Name = "budget" Required = "Yes"/>  
</AttGroup>
```

```
<AttDef Name = "prjID" Type = "ID" Required = "Yes"/>  
<AttDef Name = "start"/>  
<AttDef Name = "budget" Required = "Yes"/>  
</ElementDecl>  
</DocumentDef>
```

2.5 Microsofts XML-Data

```
<?xml version = "1.0"?>
<Schema name = ""
  xmlns = "urn:schemas-microsoft-com:xml-data"
  xmlns:dt = "urn:schemas-microsoft-com:datatypes">
<ElementType name = "ProjektAbwicklung" content = "eltOnly" order = "seq">
<element type = "person" minOccurs = "1" maxOccurs = "*" />
<element type = "projekt" minOccurs = "0" maxOccurs = "*" />
</ElementType>

<ElementType name = "person" content = "eltOnly" order = "seq">
<AttributeType name = "persID" dt:type = "id" required = "yes" />
<AttributeType name = "gehaltsGrp" dt:type = "enumeration"
dt:values = "1 1a 2" default = "1" />
<attribute type = "persID" />
<attribute type = "gehaltsGrp" />
<element type = "vorname" minOccurs = "1" maxOccurs = "*" />
<element type = "name" />
<element type = "projektLeiter" minOccurs = "0" maxOccurs = "1" />
<element type = "projektMitarb" minOccurs = "0" maxOccurs = "*" />
</ElementType>

<ElementType name = "vorname" content = "textOnly" />
<ElementType name = "name" content = "textOnly" />
<ElementType name = "projektLeiter" content = "empty">
<AttributeType name = "prjRef" dt:type = "idref" required = "yes" />
<attribute type = "prjRef" />
</ElementType>

<ElementType name = "projektMitarb" content = "empty">
<AttributeType name = "prjRef" dt:type = "idref" required = "yes" />
<attribute type = "prjRef" />
</ElementType>

<ElementType name = "projekt" content = "empty">
<AttributeType name = "prjID" dt:type = "id" required = "yes" />
<AttributeType name = "start" dt:type = "string" />
<AttributeType name = "budget" dt:type = "string" required = "yes" />
<attribute type = "prjID" />
<attribute type = "start" />
<attribute type = "budget" />
</ElementType>

</Schema>
```

2.6 RELAX

```

<?xml version = "1.0"?>
<module moduleVersion = "1.2"
  relaxCore = "1.0"
  targetNamespace = ""
  xmlns = "http://www.xml.gr.jp/xmlns/relaxCore">
<interface>
<export label = "ProjektAbwicklung"/>
</interface>
<elementRule role = "ProjektAbwicklung">
<sequence>
<ref label = "person" occurs = "+"/>
<ref label = "projekt" occurs = "*"/>
</sequence>
</elementRule>
<tag name = "ProjektAbwicklung"/>
<elementRule role = "person">
<sequence>
<ref label = "vorname" occurs = "+"/>
<ref label = "name"/>
<ref label = "projektLeiter" occurs = "?"/>
<ref label = "projektMitarb" occurs = "*"/>
</sequence>
</elementRule>
<tag name = "person">
<attribute name = "persID" required = "true" type = "ID"/>
<attribute name = "gehaltsGrp" type = "NMTOKEN">
<enumeration value = "1"/>
<enumeration value = "1a"/>
<enumeration value = "2"/>
</attribute>
</tag>

<elementRule role = "vorname" type = "string"/>
<tag name = "vorname"/>
<elementRule role = "name" type = "string"/>
<tag name = "name"/>
<elementRule role = "projektLeiter">
<empty/>
</elementRule>
<tag name = "projektLeiter">
<attribute name = "prjRef" required = "true" type = "IDREF"/>

```

```
</tag>
```

```
<elementRule role = "projektMitarb">
```

```
<empty/>
```

```
</elementRule>
```

```
<tag name = "projektMitarb">
```

```
<attribute name = "prjRef" required = "true" type = "IDREF"/>
```

```
</tag>
```

```
<elementRule role = "projekt">
```

```
<empty/>
```

```
</elementRule>
```

```
<tag name = "projekt">
```

```
<attribute name = "prjID" required = "true" type = "ID"/>
```

```
<attribute name = "start" type = "string"/>
```

```
<attribute name = "budget" required = "true" type = "string"/>
```

```
</tag>
```

```
</module>
```

2.7 Schema for object oriented XML Version 2(SOX2)

```
<?xml version = "1.0"?>
<!DOCTYPE schema SYSTEM "urn:x-commerceone:document:com:commerceone:xdk:xml:schema.dt
<schema uri = "" soxlang-version = "V0.2.2">
<elementtype name = "ProjektAbwicklung">
<model>
<sequence>
<element type = "person" occurs = "+"/>
<element type = "projekt" occurs = "*"/>
</sequence>
</model>
</elementtype>

<elementtype name = "person">
<model>
<sequence>
<element type = "vorname" occurs = "+"/>
<element type = "name"/>
<element type = "projektLeiter" occurs = "?"/>
<element type = "projektMitarb" occurs = "*"/>
</sequence>
</model>
<attdef name = "persID" datatype = "ID">
<required/>
</attdef>
<attdef name = "gehaltsGrp">
<enumeration datatype = "ENUMERATION">
<option>1</option>
<option>1a</option>
<option>2</option>
</enumeration>
<default>1</default>
</attdef>
</elementtype>

<elementtype name = "vorname">
<model>
<string/>
</model>
</elementtype>

<elementtype name = "name">
```

```
<model>
<string/>
</model>
</elementtype>

<elementtype name = "projektLeiter">
<empty/>
<attdef name = "prjRef" datatype = "IDREF">
<required/>
</attdef>
</elementtype>

<elementtype name = "projektMitarb">
<empty/>
<attdef name = "prjRef" datatype = "IDREF">
<required/>
</attdef>
</elementtype>

<elementtype name = "projekt">
<empty/>
<attdef name = "prjID" datatype = "ID">
<required/>
</attdef>
<attdef name = "start">
<implied/>
</attdef>
<attdef name = "budget">
<required/>
</attdef>
</elementtype>

</schema>
```

2.8 Microsofts XML-Data Reduced

```

<?xml version = "1.0"?>
<Schema name = ""
  xmlns = "urn:schemas-microsoft-com:xml-data"
  xmlns:dt = "urn:schemas-microsoft-com:datatypes">
<ElementType name = "ProjektAbwicklung" content = "eltOnly" order = "seq">
<element type = "person" minOccurs = "1" maxOccurs = "*" />
<element type = "projekt" minOccurs = "0" maxOccurs = "*" />
</ElementType>

<ElementType name = "person" content = "eltOnly" order = "seq">
<AttributeType name = "persID" dt:type = "id" required = "yes" />
<AttributeType name = "gehaltsGrp" dt:type = "enumeration"
  dt:values = "1 1a 2" default = "1" />
<attribute type = "persID" />
<attribute type = "gehaltsGrp" />
<element type = "vorname" minOccurs = "1" maxOccurs = "*" />
<element type = "name" />
<element type = "projektLeiter" minOccurs = "0" maxOccurs = "1" />
<element type = "projektMitarb" minOccurs = "0" maxOccurs = "*" />
</ElementType>

<ElementType name = "vorname" content = "textOnly" />
<ElementType name = "name" content = "textOnly" />
<ElementType name = "projektLeiter" content = "empty">
<AttributeType name = "prjRef" dt:type = "idref" required = "yes" />
<attribute type = "prjRef" />
</ElementType>

<ElementType name = "projektMitarb" content = "empty">
<AttributeType name = "prjRef" dt:type = "idref" required = "yes" />
<attribute type = "prjRef" />
</ElementType>

<ElementType name = "projekt" content = "empty">
<AttributeType name = "prjID" dt:type = "id" required = "yes" />
<AttributeType name = "start" dt:type = "string" />
<AttributeType name = "budget" dt:type = "string" required = "yes" />
<attribute type = "prjID" />
<attribute type = "start" />
<attribute type = "budget" />
</ElementType>

</Schema>

```

2.9 W3Cs XML Schema Description (XSD)

```

<?xml version = "1.0"?>
<schema xmlns = "http://www.w3.org/1999/XMLSchema">
<element name = "ProjektAbwicklung">
<complexType content = "elementOnly">
<sequence>
<element ref = "person" minOccurs = "1" maxOccurs = "unbounded"/>
<element ref = "projekt" minOccurs = "0" maxOccurs = "unbounded"/>
</sequence>
</complexType>
</element>

<element name = "person">
<complexType content = "elementOnly">
<sequence>
<element ref = "vorname" minOccurs = "1" maxOccurs = "unbounded"/>
<element ref = "name"/>
<element ref = "projektLeiter" minOccurs = "0" maxOccurs = "1"/>
<element ref = "projektMitarb" minOccurs = "0" maxOccurs = "unbounded"/>
</sequence>
<attribute name = "persID" use = "required" type = "ID"/>
<attribute name = "gehaltsGrp" use = "default" value = "1"/>
<simpleType base = "ENUMERATION">
<enumeration value = "1"/>
<enumeration value = "1a"/>
<enumeration value = "2"/>
</simpleType>
</attribute>
</complexType>
</element>

<element name = "vorname" type = "string"/>
<element name = "name" type = "string"/>
<element name = "projektLeiter">
<complexType content = "empty">
<attribute name = "prjRef" use = "required" type = "IDREF"/>
</complexType>
</element>

<element name = "projektMitarb">
<complexType content = "empty">
<attribute name = "prjRef" use = "required" type = "IDREF"/>

```

```
</complexType>
</element>

<element name = "projekt">
  <complexType content = "empty">
    <attribute name = "prjID" use = "required" type = "ID"/>
    <attribute name = "start" type = "string"/>
    <attribute name = "budget" use = "required" type = "string"/>
  </complexType>
</element>

</schema>
```

2.10 Software AGs Tamino Schema

```

<?xml version = "1.0"?>
<Schema name = "example.tam"
  xmlns = "urn:schemas-microsoft-com:xml-data"
  xmlns:dt = "urn:schemas-microsoft-com:datatypes"
  xmlns:tam = "http://namespaces.softwareag.com/tamino/schema">
<ElementType name = "ProjektAbwicklung" content = "eltOnly" order = "seq">
<element type = "person" minOccurs = "1" maxOccurs = "*" />
<element type = "projekt" minOccurs = "0" maxOccurs = "*" />
</ElementType>

<ElementType name = "person" content = "eltOnly" order = "seq">
<AttributeType name = "persID" dt:type = "id" required = "yes" />
<AttributeType name = "gehaltsGrp" dt:type = "enumeration" dt:values = "1 1a 2" default = "1" />
<attribute type = "persID" />
<attribute type = "gehaltsGrp" />
<element type = "vorname" minOccurs = "1" maxOccurs = "*" />
<element type = "name" />
<element type = "projektLeiter" minOccurs = "0" maxOccurs = "1" />
<element type = "projektMitarb" minOccurs = "0" maxOccurs = "*" />
</ElementType>

<ElementType name = "vorname" content = "textOnly" />
<ElementType name = "name" content = "textOnly" />
<ElementType name = "projektLeiter" content = "empty">
<AttributeType name = "prjRef" dt:type = "idref" required = "yes" />
<attribute type = "prjRef" />
</ElementType>

<ElementType name = "projektMitarb" content = "empty">
<AttributeType name = "prjRef" dt:type = "idref" required = "yes" />
<attribute type = "prjRef" />
</ElementType>

<ElementType name = "projekt" content = "empty">
<AttributeType name = "prjID" dt:type = "id" required = "yes" />
<AttributeType name = "start" dt:type = "string" />
<AttributeType name = "budget" dt:type = "string" required = "yes" />
<attribute type = "prjID" />
<attribute type = "start" />
<attribute type = "budget" />
</ElementType>
</Schema>

```

3 References

3.1 Schema-Dialekte

- W3Cs XML-Schema
(Primer) <http://www.w3.org/TR/xmlschema-0>
(Structure) <http://www.w3.org/TR/xmlschema-1>
(DataTypes) <http://www.w3.org/TR/xmlschema-2>
- DataTypes 4 DTDs
<http://www.w3.org/TR/dt4dtd>
- Document Content Definition (DCD)
<http://www.w3.org/TR/NOTE-dcd>
- Document Description Markup Language (DDML)
<http://www.w3.org/TR/NOTE-ddml>
- Document Structure Description (DSD)
<http://www.brics.dk/DSD/>
- RELAX (REgular LAnguage description for XML)
<http://www.xml.gr.jp/relax/>
- Schema for object oriented XML (SOX)
<http://www.w3.org/TR/NOTE-SOX/>
- Schematron
<http://www.ascc.net/xml/resource/schematron/schematron.html>
- Microsofts XML-Data
<http://www.w3.org/TR/1998/NOTE-XML-data>

3.2 Sekundärquellen

- (IBM) The basics of using XML Schema to define elements
<http://www-4.ibm.com/software/developer/library/xml-schema/?open&l=136,t=gr,p=XMLschem>
- (A. Layman) XML Schema NG Guide
<http://www.lindamann.com/xml/XML%20Schemas%20NG%20Guide%20HTML.htm>
- (S. St. Laurent) DTDs and Schema
<http://www.xml.com/pub/1999/12/dtd/index.html>
- (L. Dodds) Spotlight on Schemas
http://xml.com/pub/2000/02/23/xmldeviant/index.html?wwrrrr_20000223.txt
- (L. Dodds) XML Schema revisited
http://www.xml.com/pub/2000/07/12/deviant.html?wwrrrr_20000712.txt

- Anwenderumfrage zum Schemadraft
<http://metalab.unc.edu/xql/tally.html>
- (R. E. Costello) Schematutorial
<http://www.xfront.com/xml-schema.html>
- (T. Bray) Adding Strong Data Typing to SGML and XML
<http://www.textuality.com/xml/typing.html>
- Mario Jeckle Online
<http://www.jeckle.de>

3.3 Verfügbare Softwareunterstützung

- Übersicht verfügbarer Schemaparser beim W3C
<http://www.w3.org/XML/Schema.html#Tools>
- XML Authority
<http://www.extensibility.com>
- Xerces-Parser
<http://xml.apache.org>