

TAC-POT #2

XML





Einführung in die Theorie

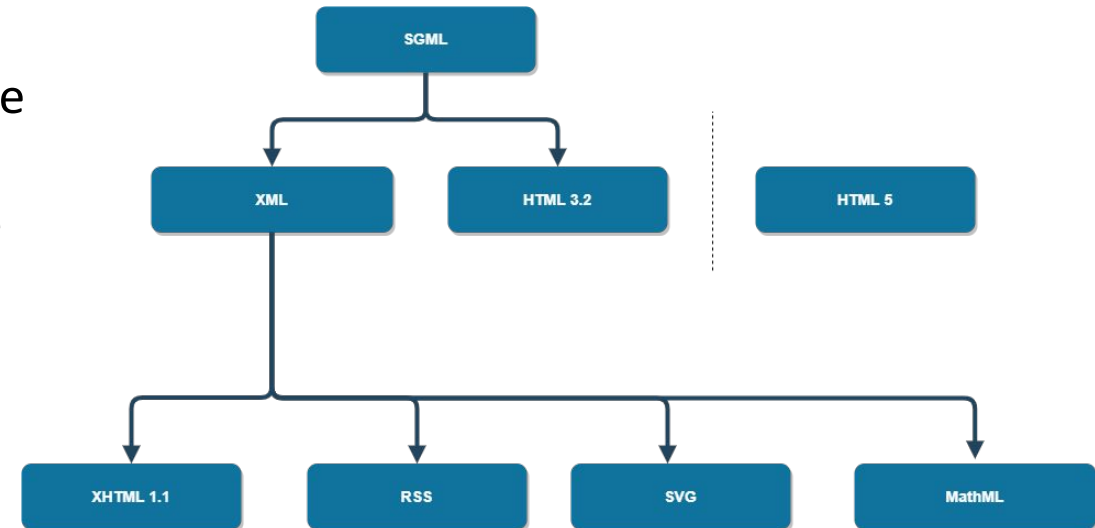
1. Geschichte von XML
2. X-Stack Technologien
3. XML vs. JSON
4. XML und pom

Praxisbeispiel

Blackjack

SGML als Ausgangspunkt

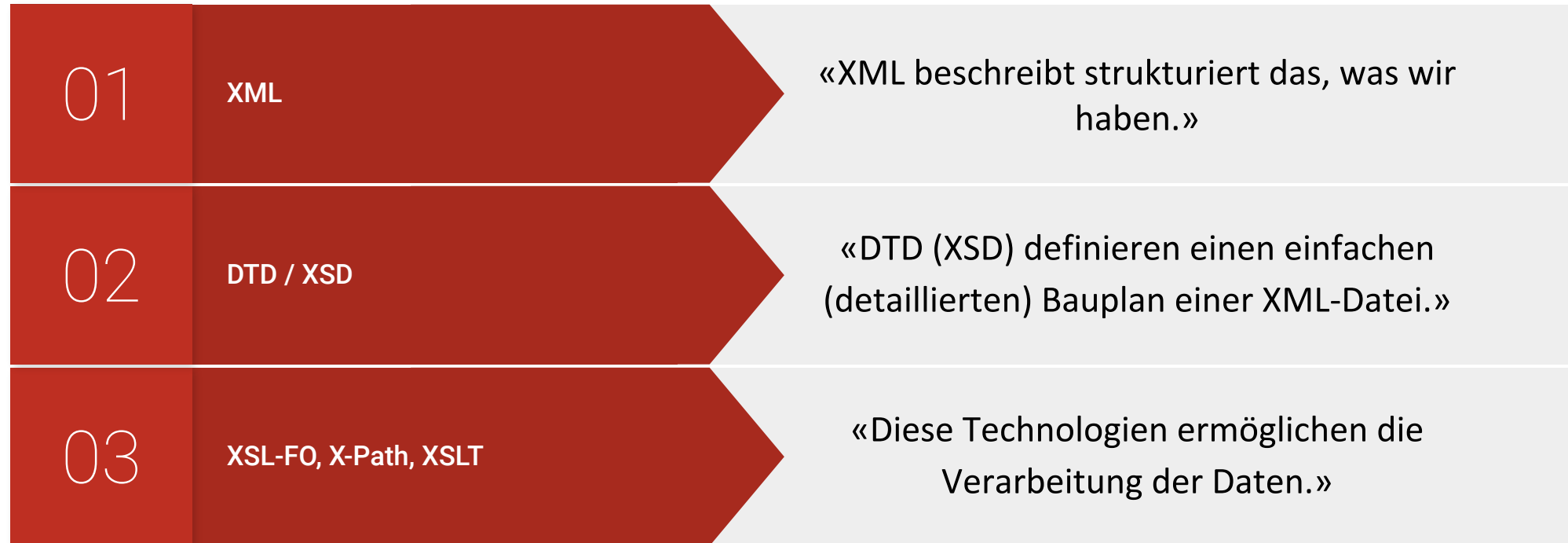
- Standard Generalized Markup Language ist eine Auszeichnungssprache
- SGML enthielt bereits DTD-Dokumente
- 1998: Entwicklung von XML als vereinfachte “Teilmenge” von SGML



Weitere Entwicklung von XML

- Auf Basis von XML wurden beispielsweise SVG oder XHTML entwickelt
- XHTML sollte HTML ablösen, was jedoch nicht gelang
- HTML5 (seit 2014) vereint als “core language” im Web die Standards HTML, XHTML und DOM (Document Object Model)

Was ist der X-Stack?



1. XML - Basics

- eXtensible Markup Language
 - Methode zur Speicherung strukturierter Daten in Textform
 - Menschenlesbar und hierarchisch
 - Standardisiert und austauschbar
 - DTD und XSD zum Überprüfen valider XML Dokumente
-
- Dokument beginnt optional mit `<?xml version=".." encoding="UTF-8" ?>`
 - DTD wird über `!DOCTYPE` definiert
 - Opening- (`<to>`) und ein closing-tags (`</to>`) sind erforderlich

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE note SYSTEM "Note.dtd">
<note>
<to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend!</body>
</note>
```

2. XML - Namespaces

- Definiert XML Schema
- `xmlns:prefix="URI"` (Uniform Resource Identifier)
- Erlaubt gleiche Namen mit unterschiedlichem Prefix
- Kann in jedem tag definiert werden
- Selbst definierter Namespace: XSD oder DTD

```
<root>

<h:table xmlns:h="http://www.w3.org/TR/html4/">
  <h:tr>
    <h:td>Apples</h:td>
    <h:td>Bananas</h:td>
  </h:tr>
</h:table>

<f:table xmlns:f="https://www.w3schools.com/furniture">
  <f:name>African Coffee Table</f:name>
  <f:width>80</f:width>
  <f:length>120</f:length>
</f:table>

</root>
```

2. XML - DTD und XSD

DTD

```
<!ELEMENT note (to, from, heading, body)>  
<!ELEMENT to (#PCDATA)>  
<!ELEMENT from (#PCDATA)>  
<!ELEMENT heading (#PCDATA)>  
<!ELEMENT body (#PCDATA)>
```

XSD

```
<?xml version="1.0"?>  
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"  
  targetNamespace="https://www.w3schools.com"  
  xmlns="https://www.w3schools.com"  
  elementFormDefault="qualified">  
  
  <xs:element name="note">  
    <xs:complexType>  
      <xs:sequence>  
        <xs:element name="to" type="xs:string"/>  
        <xs:element name="from" type="xs:string"/>  
        <xs:element name="heading" type="xs:string"/>  
        <xs:element name="body" type="xs:string"/>  
      </xs:sequence>  
    </xs:complexType>  
  </xs:element>  
  
</xs:schema>
```

3. DTD

- Document Type Definition
- Überprüfen valider XML Dokumente
- Definieren von Elementen, Attributen und ihre Datentypen
- Separates Dokument oder innerhalb des XML-Dokuments

- !DOCTYPE: definiert Wurzelelement
- !ELEMENT: definiert Element
- Element besteht (Elementen || Datentyp)
- #PCDATA: parsable character data

- Hands-On: Rezeptbuch

```
<!ELEMENT note (to, from, heading, body)>  
<!ELEMENT to (#PCDATA)>  
<!ELEMENT from (#PCDATA)>  
<!ELEMENT heading (#PCDATA)>  
<!ELEMENT body (#PCDATA)>
```

```
<!DOCTYPE note  
[  
<!ELEMENT note (to,from,heading,body)>  
<!ELEMENT to (#PCDATA)>  
<!ELEMENT from (#PCDATA)>  
<!ELEMENT heading (#PCDATA)>  
<!ELEMENT body (#PCDATA)>  
>
```


4. XSL-FO

- Mit *XSL Formatting Objects* lassen sich für den Druck optimierte Dateien (PDF) erzeugen.
- Festlegung aller Output-Parameter, z.B. Aufteilung und Definition der Größe einer Seite.
- Aus einer XML-Datei und einem XSL-Stylesheet wird mit Hilfe eines XSLT-Prozessors eine XSL-FO-Datei generiert (vgl. HTML). Visualisierung wie auch bei HTML mittels Browser.
- CSS ist für Bildschirme und XSL-FO für den Druck entwickelt worden.

```
<fo:layout-master-set>
  <fo:simple-page-master master-name="Postkartenformat" page-width="105mm" page-height="148.5mm">
    ...
  </fo:simple-page-master>
</fo:layout-master-set>
```

5. XPath

- Adressierung von Elementen in XML-Dokumenten, durch Navigation durch die hierarchische Struktur (Mini-Query-Sprache)
- Filterung einzelner Elemente
- Nutzung in JS, Java, XML Schema, Python, etc.
- Bsp.:
 - `blackjack/games/game[@id = 2]`
 - `players/player[@type='user']`
 - `players/player[position() = 1]`
 - `./ancestor::node()/child::hands/child::hand[@id=$playerID]/total`

6. XSLT

- *XSL Transformation* definiert Regeln, mit denen sich ein XML-Dokument in etwas anderes transformieren lässt. Zum Beispiel in HTML.
- In folgendem Beispiel wird die Anzahl von Spielern gezählt, welche in “Spiel 1” registriert sind. Hierbei kommt auch die X-Path Technologie zur Anwendung (vgl. Slide 10):

```
<xsl:variable name="checkForFreePlayers" select="count(blackjack/games/game[@id=1]/players/player)"/>
```

- Try XSLT und HTML:
<https://www.w3schools.com/xml/tryxslt.asp?xmlfile=catalog&xsltfile=catalog>

XML

- Unterstützt REST-Idee (X-Stack)
- Als Datenbank nutzbar
- Namespaces können Elemente und Attribute weltweit eindeutig definieren

XML

```
<employees>
  <employee>
    <firstName>John</firstName> <lastName>Doe</lastName>
  </employee>
  <employee>
    <firstName>Anna</firstName> <lastName>Smith</lastName>
  </employee>
  <employee>
    <firstName>Peter</firstName> <lastName>Jones</lastName>
  </employee>
</employees>
```

JSON

- Leichter lesbar
- Kürzer
- Serialisierung von Objekten
- Format zur Datenübertragung zwischen Server und Client

JSON

```
{"employees":[
  { "firstName":"John", "lastName":"Doe" },
  { "firstName":"Anna", "lastName":"Smith" },
  { "firstName":"Peter", "lastName":"Jones" }
]}
```

Was ist pom?

- Das "Project Object Model" ist eine XML-Repräsentation eines Maven Projektes zur Vereinfachung des Build-Managements
- Standardisierung von Software, welche häufig dieselben Lebenszyklus-Phasen durchläuft
- Erweiterung von Java Projekten
- Eine *pom.xml*-Datei enthält strikt definierte Elemente, u.a.:
 - Konfigurationsdateien,
 - URL des Projekts,
 - Dependencies, aber auch
 - Plugins (Bsp.: JAR).
- Vergleichbar mit *build.xml* von ANT



Wie ist eine *pom.xml*-Datei aufgebaut?

Minimaler Inhalt einer pom.xml:

- project - root des Projekts
- modelVersion - Standardwert ist: 4.0.0
- groupId - ID der Projektgruppe
- artifactId - ID des artifacts (Projekts)
- version - Version des artifacts

```
1. <project>
2.   <modelVersion>4.0.0</modelVersion>
3.   <groupId>com.mycompany.app</groupId>
4.   <artifactId>my-app</artifactId>
5.   <version>1</version>
6. </project>
```

Beispiel: “Blackjack”

- <https://web-development.github.io/xml/was-ist-xml/>
- <https://de.wikipedia.org/wiki/HTML5>
- <https://www.data2type.de/xml-xslt-xslfo/xsl-fo/in-drei-schritten-zum-druck/>
- [https://de.wikibooks.org/wiki/Websiteentwicklung: XML: Geschichte](https://de.wikibooks.org/wiki/Websiteentwicklung:_XML:_Geschichte)
- <https://www.digicomp.ch/blog/2017/07/04/xml-dtd-xsd-xsl-xslt-xsl-fo-und-xpath-im-uberblick>
- <https://maven.apache.org/pom.html>
- https://de.wikipedia.org/wiki/Apache_Maven
- <https://www.w3schools.com/xml/default.asp>
- <https://www.predic8.de/xml-json-yaml.htm>
- Ausführliche Zusammenfassung:
<http://www.lgis.informatik.uni-kl.de/archiv/wwwdvs.informatik.uni-kl.de/courses/seminar/WS0203/ausarbeitung1.pdf>