

<seminar>

<begrueessung>Willkommen zum</begrueessung>
<seminarthema>XML-Workshop</seminarthema>

<veranstalter>KnowHow AG</veranstalter>

<referent> christoph runkel</referent>

</seminar>

XML-Workshop

Christoph Runkel

XML-Definition

XML: „Extensible Markup Language“

- **Extensible** „Erweiterbar“, d.h. sie hat keinen starren Befehlsumfang.
- **Markup Language** „Auszeichnungssprache die die in ihr enthaltenen Informationen nicht nur darstellt, sondern qualifiziert.

XML ist unabhängig...

- Vom Hersteller
- Vom Betriebssystem
- Von dem Ausgabemedium
- Von der Anwendung

➔ XML kann als Strategie verstanden werden !

Die Umgebung von XML

- XML-Dokumente
 - Beihaltet lediglich die Informationen
- DTD's(Document Type-Definitions) / X-Schema
Definieren die zulässige Dokumentenstruktur
- CSS (Cascading Style-Sheets)
 - Legen die Gestaltungsregeln fest
- XSL(T)(eXtended Style Language (Transform))
 - Definiert Gestaltungs- / Konvertierungsregeln

Die XML-Umgebung

- Eingabe
 - Durch Standardeditoren (z.B. Notepad)
 - Durch Spezialapplikationen (z.B. XMLSPY)
 - Konvertierung aus bestehenden Anwendungen (z.B. aus Office XP, insbesondere Access!)
- Verarbeitung
 - Verarbeitung/Validierung durch Parser
 - Konvertierung durch XML-Prozessoren)
- Ausgabe / Export
 - Anzeige durch Browser oder andere Applikationen
 - Export in andere Formate z.B. WAP-XML, HTML, PDF, Datenbankformate, RTF ..

Das erste XML-Dokument

```
<?xml version=„1.0“ encoding=„ISO-8859-1“?>
```

Prolog

```
<person kategorie=„schauspieler“ >  
<name>  
  <vorname>Walter</vorname>  
  <nachname>Matthau</nachname>  
</name>  
<geburtstag>31.01.29</geburtstag>  
</person>
```

Inhalt



Element



Wurzelement



Attribut

Wichtige Grundregeln zu XML

- Jedes XML-Dokument sollte einen Prolog aufweisen
- Jedes XML hat genau EIN root element
- Jedes XML-Element muß geschlossen werden `<name> ... </name>` oder kurz `<telefax/>`
- Keine Leerzeichen/Umlaute innerhalb der Elementnamen
- Attribute stehen in Anführungszeichen
- XML unterscheidet Groß-Kleinschreibung
- Kommentare wie in HTML `<!-- Kommentar -->`
- Processing instructions (Verarbeitungsanweisungen) sind eingeschlossen in `<? ?>`
- Die Elemente müssen korrekt verschachtelt sein.

Die Einhaltung dieser Regeln führt zu einem „wohlgeformten“ XML-Dokument

Document Type Definitions

- Beschreiben die Grammatik eines XML-Dokuments
 - Können vorliegen (auch gemischt)
 - Integriert in das XML-Dokument
 - Als externe DTD die auf viele XML-Dokumente angewendet werden kann
 - Als öffentliche DTD auf die allgemein zugegriffen werden kann (Standard-DTD's)
 - Ermöglichen die Prüfung eines XML-Dokumentes entsprechend der Grammatik (Fehlerreduzierung)
- XML-Dokumente die der DTD-Grammatik entsprechen werden als „gültig“ bezeichnet.

XSchema - Nachfolger der DTD?

- Erfüllen den gleichen Zweck wie DTD's
- Sind XML-Konform aufgebaut
- Bieten zusätzliche Möglichkeiten
 - Datentypen
 - Feldlängen
 - Genaue Anzahl an Aufrufen
- Sind wesentlich aufwändiger in der Erstellung.

XSchema - ein Beispiel

....

```
<xsd:element name="PLZ" minOccurs="0"
  maxOccurs="1">
```

```
<xsd:simpleType>
```

```
<xsd:restriction base="xsd:string">
```

```
<xsd:maxLength value="5"/>
```

```
</xsd:restriction>
```

```
</xsd:simpleType>
```

```
</xsd:element>
```

CSS: Cascading Style Sheets

- Eine der Möglichkeiten XML-Dokumente zu gestalten
- Bekannt aus HTML, wird von Browsern unterstützt
- Relativ einfach zu erstellen, bzw. sogar aus bestehenden Styleguides zu übernehmen
- Kann für einzelnes Element, eine XML-Seite oder für einen kompletten Dokumentenbestand definiert werden.

XSL-Extended Style Language

- Ist ebenfalls für die Formatierung von XML-Dokumenten konzipiert
- Ist XML-konform aufgebaut (keine eigene Syntax)
- Besitzt mehr Möglichkeiten als CSS
- Komponenten sind XSL-FO und XSLT

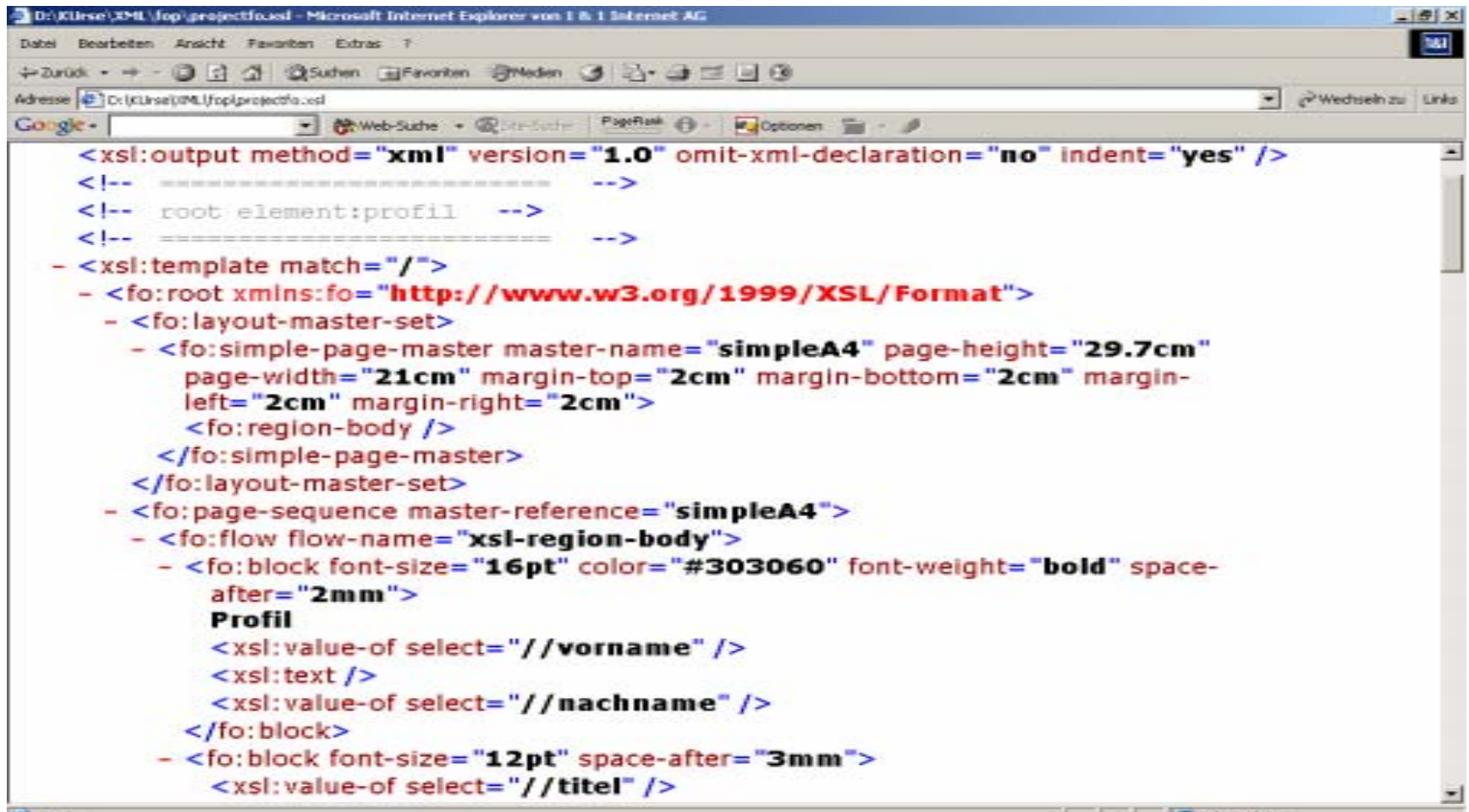
XSL-Formatting Objects

- Formatierungskomponente von XSL
- In der Regel für druckorientierte Formate (PDF, RTF) eingesetzt
- Mächtigste gestalterische Möglichkeiten

XSL-FO: Das XML-Dokument

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
- <profil>
- <persdat>
  <titel>Diplom-Betriebswirt(BA)</titel>
  - <name>
    <vorname>Eberhard</vorname>
    <nachname>Muster</nachname>
  </name>
  <gebdat>03.11.1961</gebdat>
</persdat>
- <anschrift>
  <strasse>Hasenstrasse 100</strasse>
  <plz>70371</plz>
  <ort>Stuttgart</ort>
  <fon>+49 711 61 11 398</fon>
  <fax>+49 711 74 56 551</fax>
  <mail>dialog@musteradresse.de</mail>
  <internet>http://www.musteradresse.de</internet>
</anschrift>
  <bild type="jpg" width="" height="" src="musterfoto.jpg" />
- <katgorie>
  <kname>Position</kname>
- </katgorie>
</eintrag>
</profil>
```

XSL-FO: Das XSL-Stylesheet



The screenshot shows a web browser window displaying an XSL-FO stylesheet. The code is as follows:

```
<xsl:output method="xml" version="1.0" omit-xml-declaration="no" indent="yes" />
<!-- ----- -->
<!-- root element:profil -->
<!-- ----- -->
- <xsl:template match="/">
- <fo:root xmlns:fo="http://www.w3.org/1999/XSL/Format">
- <fo:layout-master-set>
- <fo:simple-page-master master-name="simpleA4" page-height="29.7cm"
  page-width="21cm" margin-top="2cm" margin-bottom="2cm" margin-
  left="2cm" margin-right="2cm">
  <fo:region-body />
</fo:simple-page-master>
</fo:layout-master-set>
- <fo:page-sequence master-reference="simpleA4">
- <fo:flow flow-name="xsl-region-body">
- <fo:block font-size="16pt" color="#303060" font-weight="bold" space-
  after="2mm">
  Profil
  <xsl:value-of select="//vorname" />
  <xsl:text />
  <xsl:value-of select="//nachname" />
</fo:block>
- <fo:block font-size="12pt" space-after="3mm">
  <xsl:value-of select="//titel" />
```


XSL-FO: Das Ergebnis in PDF

Adobe Acrobat - [profil3.pdf]

Datei Bearbeiten Dokument Werkzeuge Anzeige Fenster Hilfe

100%

Profil Eberhard Muster
Diplom-Betriebswirt(BA)

 Hasenstrasse 100
70371 Stuttgart

Telefon: +49 711 61 11 308
Telefax: +49 711 74 56 551
e-mail: dialog@musteradresse.de
http://www.musteradresse.de
Geburtsdatum: 03.11.1961

Position

TrainerIn seit	1995
BeraterIn seit	2001

Ausbildung Studium Betriebswirtschaft an der Berufsakademie Stuttgart

Zusatzausbildung Präsentations- und Moderationstechniken, Web Design, DB-Design, zahlreiche Programmierer-Trainings

Sprachen Englisch: sehr gut in Wort und Schrift

Produktportfolio

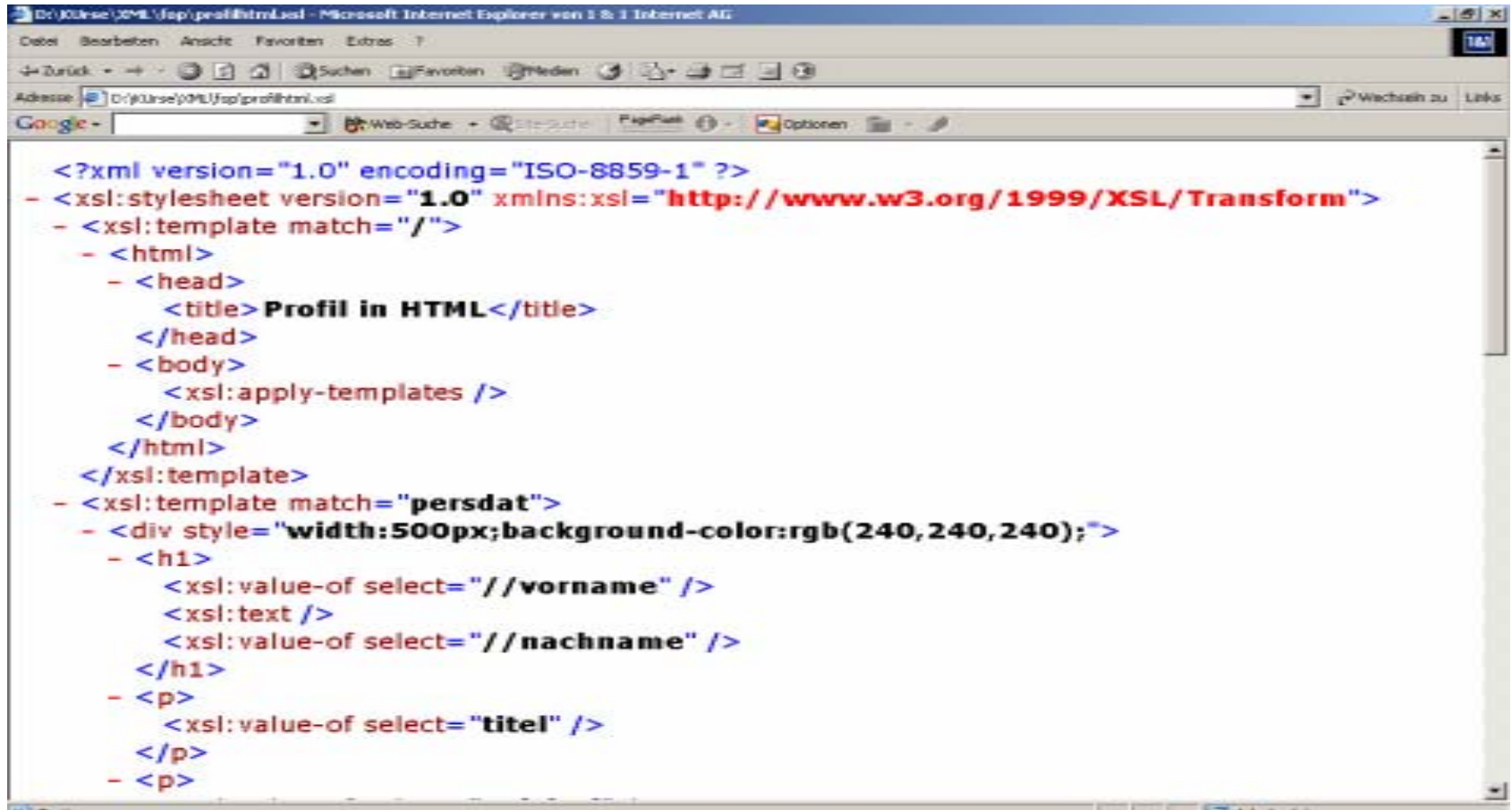
Kernkompetenzen Beratung, Projektleitung, Realisierung zu Webprojekten aller Art, sowie Durchführung von Methodik- und Programmierlehrgängen im gesamten Bereich Internet / Intranet / E-Business
Suchmaschinenoptimierung, Webmarketing

Seminare / Trainings in: Webdesign (HTML, CSS, JavaScript)
Internet-Anwendungsentwicklung(Perl, ASP, ASP.NET/PHP) XML, XSLT Anwendungsschulungen (Photoshop, Dreamweaver, Visual Studio, Acrobat, Frontpage) Methodik-Seminare (Marketing in

XSLT - XML Transformieren

- Bestandteil von XSL
- Konzipiert um XML-Dokumente in andere Formate (HTML, PDF, SQL-Statements) zu überführen
- Besitzt wesentliche Merkmale einer Programmiersprache (Schleifen, Bedingungen)
- Sehr mächtiges Werkzeug

XSLT: Ein Beispiel

A screenshot of a Microsoft Internet Explorer browser window. The title bar reads "Dr:\Kurse\XML\jsp\profil.html.xsl - Microsoft Internet Explorer von 1 & 1 Internet AG". The address bar shows "Dr:\Kurse\XML\jsp\profil.html.xsl". The main content area displays XSLT code with red and blue syntax highlighting. The code includes an XML declaration, an XSL stylesheet reference, a root template, and a specific template for "persdat" that uses XSLT functions like `value-of` and `text` to output XML elements.

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
- <xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
- <xsl:template match="/">
- <html>
- <head>
  <title>Profil in HTML</title>
</head>
- <body>
  <xsl:apply-templates />
</body>
</html>
</xsl:template>
- <xsl:template match="persdat">
- <div style="width:500px;background-color:rgb(240,240,240);">
- <h1>
  <xsl:value-of select="//vorname" />
  <xsl:text />
  <xsl:value-of select="//nachname" />
</h1>
- <p>
  <xsl:value-of select="titel" />
</p>
- <p>
```

XSLT: Ein Beispiel - das Ergebnis

Christoph Runkel
Diplom-Betriebswirt(BA)

Einsteinstr.
70374
Stuttgart.
+49 711 6
+49 711 6
dialog@ab
http://www
Position

Trainer/in
Berater/in
Ausbildun

```
<html><head>      <META http-equiv="Content-Type" content="text/html; charset=UTF-8"><title>  
</div>  
<div style="width:500px;background-color:rgb(200,200,200);"><p>Einsteinstrasse 100<br>  
</div>  
<div style="width:500px;background-color:rgb(200,200,200);">  
  <table border="0"><thead><tr><th colspan="2">Position</th></tr>  
  </thead><tbody>Position<tr><td>Trainer/in seit</td><td>1995</td>  
  </tr>  
  <tr><td>Berater/in seit</td><td>2001</td>  
  </tr>  
  <tr><td>Ausbildung</td><td>Studium Betriebswirtschaft an der Berufsakademi  
</tr>  
  <tr><td>Zusatzausbildung</td><td>Präsentations- und Moderationstechniken,  
</tr>  
  <tr><td>Sprachen</td><td>Englisch: sehr gut in Wort und Schrift </td>  
</tr>  
</tbody></table>  
</div>  
<div style="width:500px;background-color:rgb(200,200,200);">  
  <table border="0"><thead><tr><th colspan="2">Produktportfolio</th></tr>
```

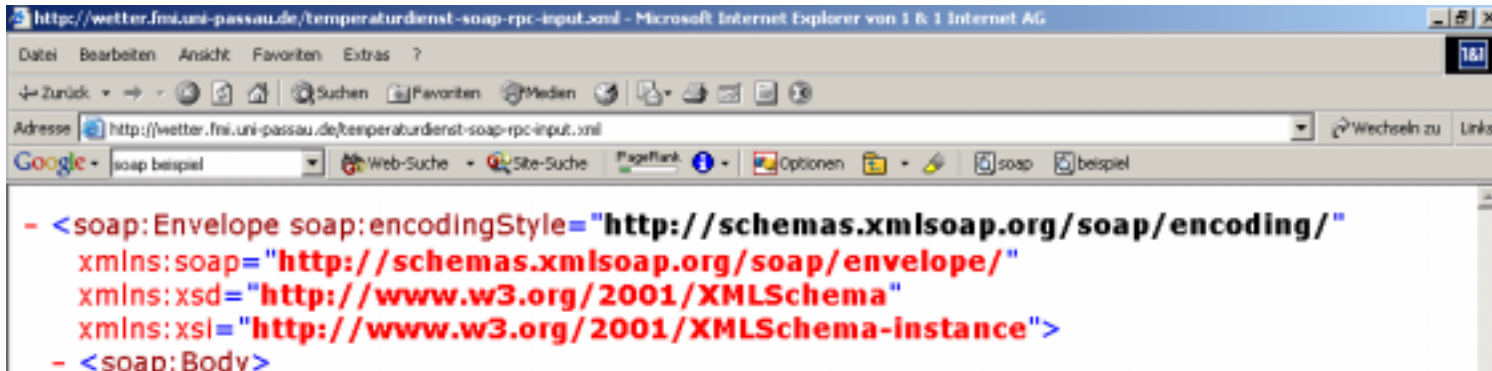
Trends: XML und Datenbanken

- XML und Datenbanken konkurrieren nicht zwangsläufig, sie können sich ergänzen
- Datenbanktypen
 - XML-fähige Datenbanken
Speichern relational, leiten XML-Struktur aus DTD oder XML-Schema her
 - Native XML-Datenbanken
Speichern XML-Strukturen originär ab, geeignet für komplexe Informationsstrukturen (Stücklisten, Dokumentationen etc.)

Trends: XML und Kommunikation

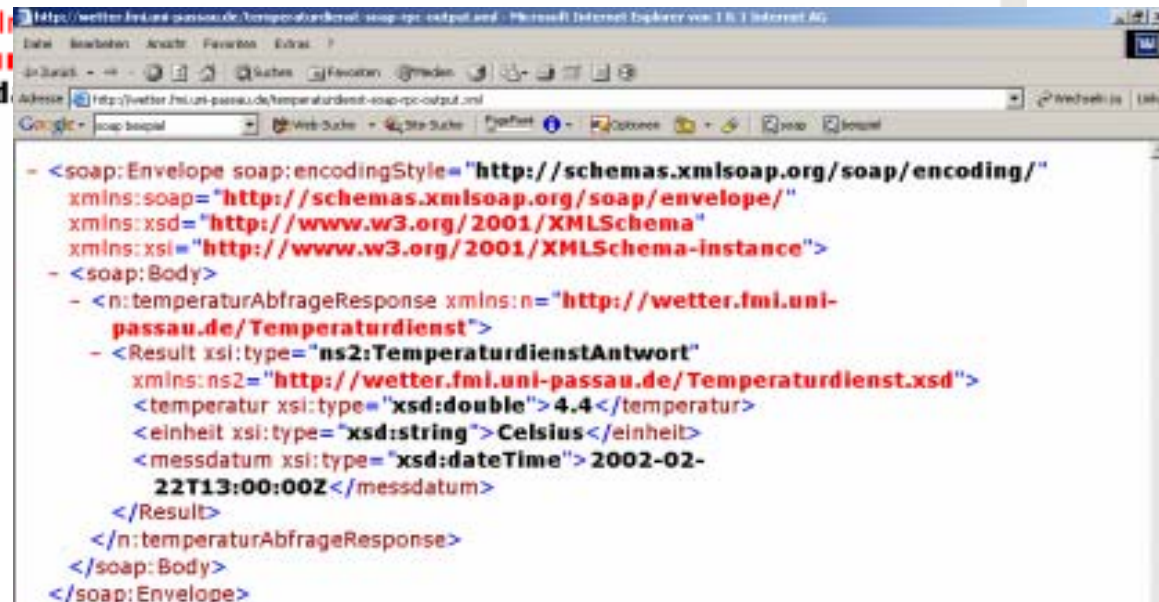
- SOAP: Simple Object Access Protocol
Daten- und Anwendungskommunikation über XML-basiertes Nachrichtenprotokoll.
 - Anwendungs- und Plattformübergreifende Kommunikation
 - Einfacher Datenaustausch über XML - Strukturen
 - Basiert auf TCP/IP / HTTP →Keine Firewallprobleme

SOAP - Ein Beispiel



```
http://wetter.fmi.uni-passau.de/temperaturdienst-soap-rpc-input.xml - Microsoft Internet Explorer von 1 & 1 Internet AG
Datei Bearbeiten Ansicht Favoriten Extras ?
Zurück Suchen Favoriten Medien
Adresse http://wetter.fmi.uni-passau.de/temperaturdienst-soap-rpc-input.xml
Google soap beispiel Web-Suche Site-Suche PageRank Optionen soap beispiel

- <soap:Envelope soap:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
- <soap:Body>
  - <n:temperaturAbfrage xmlns:wetter="http://wetter.fmi.uni-passau.de/Temperaturdienst"
    <datum xsi:type="xsd:date">2002-02-22T13:00:00Z</datum>
  </n:temperaturAbfrage>
</soap:Body>
</soap:Envelope>
```



```
http://wetter.fmi.uni-passau.de/temperaturdienst-soap-rpc-output.xml - Microsoft Internet Explorer von 1 & 1 Internet AG
Datei Bearbeiten Ansicht Favoriten Extras ?
Zurück Suchen Favoriten Medien
Adresse http://wetter.fmi.uni-passau.de/temperaturdienst-soap-rpc-output.xml
Google soap beispiel Web-Suche Site-Suche PageRank Optionen soap beispiel

- <soap:Envelope soap:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
- <soap:Body>
  - <n:temperaturAbfrageResponse xmlns:wetter="http://wetter.fmi.uni-passau.de/Temperaturdienst"
    - <Result xsi:type="ns2:TemperaturdienstAntwort"
      xmlns:ns2="http://wetter.fmi.uni-passau.de/Temperaturdienst.xsd">
      <temperatur xsi:type="xsd:double">4.4</temperatur>
      <einheit xsi:type="xsd:string">Celsius</einheit>
      <messdatum xsi:type="xsd:dateTime">2002-02-22T13:00:00Z</messdatum>
    </Result>
  </n:temperaturAbfrageResponse>
</soap:Body>
</soap:Envelope>
```

Web Services

- Auf HTTP/SOAP basierende und hierüber kommunizierende internettaugliche und über URL adressierbare Web-Anwendungen
- Sie werden zur Integration von bestehenden modularen Anwendungsfunktionalitäten (Börsenkurse, Suchmaschine, Wetterdaten ...) in andere Anwendungen oder Webseiten eingesetzt.
- Zur automatischen Einbindung von Web-Services in die Kundenanwendung dient WSDL (Web Service Description Language).

Web Services - ein Beispiel

The image displays two overlapping Microsoft Internet Explorer windows. The left window shows an XML Schema Definition (XSD) for a web service. The right window shows a search results page for 'Stephan Janssch'.

```
<part name="phrase" type="s:string" />
</message>
- <message name="doSpellingSuggestionResponse">
  <part name="return" type="s:string" />
</message>
- <message name="doGoogleSearch">
  <part name="key" type="s:string" />
  <part name="q" type="s:string" />
  <part name="start" type="s:int" />
  <part name="maxResults" type="s:int" />
  <part name="filter" type="s:boolean" />
  <part name="restrict" type="s:string" />
  <part name="safeSearch" type="s:boolean" />
  <part name="lr" type="s:string" />
  <part name="ie" type="s:string" />
  <part name="oe" type="s:string" />
</message>
- <message name="doGoogleSearchResponse">
  <part name="return" type="s0:GoogleSearchResult" />
</message>
- <portType name="GoogleSearchPort">
  - <operation name="doGetCachedPage">
    <input message="s0:doGetCachedPage" />
    <output message="s0:doGetCachedPageResponse" />
```

The right window shows a search results page for 'Stephan Janssch'. The search results include:

- Name: Stephan Janssch
- URL: <http://www.sjanssch.org/>
- Größe: 11K
- Name: Google Web Services - Stephan Janssch Homepage
- URL: <http://www.sjanssch.org/projects/google-api/index.php>
- Größe: 14K
- Name: Stephan Janssch Homepage
- URL: <http://www.sjanssch.org/old/projects/google-api/index.php>
- Größe: 12K
- Name: ECLIF Router - Stephan Janssch Homepage
- URL: <http://www.sjanssch.org/projects/steader/index.php>
- Größe: 16K