

# Cocoon 2



## Inhalt

- XML based Webpublishing
- Konfiguration mit Sitemap
- Dynamische Seiten
- Request, Sessions & Parameter
- Logicsheets
- Beispiel Anwendungsentwicklung
- Das Pipelinemodell im Detail
- Datenbank Zugriff
- Multichannel Applications
- PDF Erzeugung
- Erzeugung von Grafiken (SVG)
- Cocoon Internas

## XML based Webpublishing



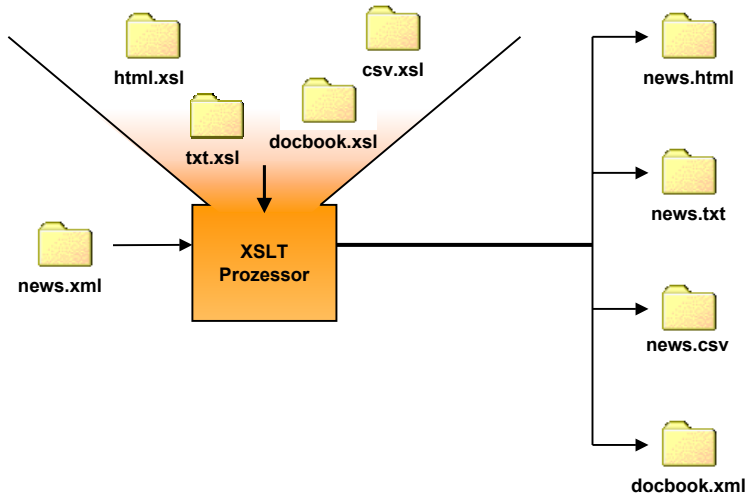
- Seiten werden aus XML Quellen erzeugt
- Daten sind strukturiert
- Inhalt wird von der Präsentation getrennt
- Crossbrowser fähig
- Basiert auf offenen XML Technologien
  - XSL, XSLT, XPath, XLink, SVG,...

## Serverside XML



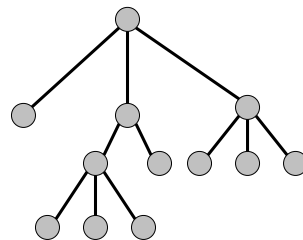
- Server wandelt XML in HTML, WML,... Um
- Aufbereitung für verschiedene Clients ist möglich
  - IE, Netscape, WAP-Client, ...
- Keine speziellen Anforderungen an den Browser
  - Java, Neue Version, ...

# XSLT Prozessor



# Die Quelle als XML

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<cdliste>
...
<cd id="65" jahr="2000">
  <titel>Music</titel>
  <interpret>Madonna</interpret>
  <hersteller>BMG</hersteller>
  <preis/>
  <track id="01">
    <titel>Music</titel>
    <laenge>4:35</laenge>
  </track>
  <track id="02">
    <titel>Impressive Instant</titel>
    <laenge>5:10</laenge>
  </track>
</cd>
...
</cdliste>
```



# Transformation mit XSLT

```
...  
<cd>  
  <titel>Music</titel>  
  <interpret>Madonna</interpret>  
</cd>  
...
```

XML-Quelle

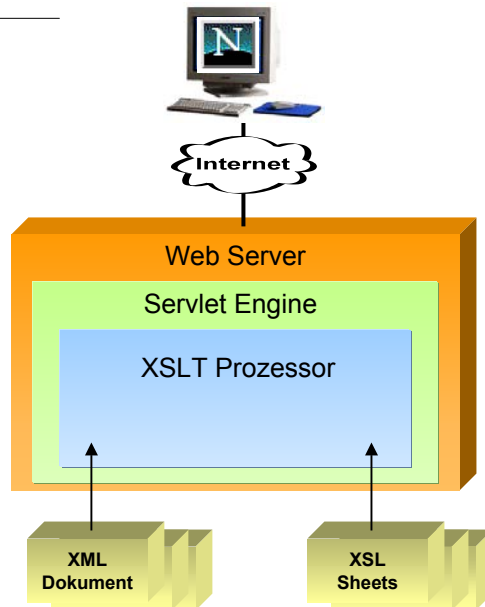
```
...  
<xsl:template match="cd">  
  <b><xsl:value-of select="titel"/></b>,  
  Interpret: <xsl:value-of select="interpret"/><br/>  
</xsl:template>  
...
```

XSL-Stylesheet

```
<b>Music</b>, Interpret: Madonna<br/>
```

Output

# Idee: XSLT im Servlet



- Apache Cocoon is a 100% pure Java publishing framework that relies on new W3C technologies (such as XML, XSL, SVG, etc..) to provide web content.

**Quelle:** Cocoon 2 README

### A small revolution

Stefano Mazzocchi (Autor und Architekt von Cocoon)



- Unter der Apache Software License
- Voraussetzungen
  - Java 1.2 VM (besser 1.3), 2.2 Servlet Engine
- <http://xml.apache.com/cocoon/>

### C1

- Basiert auf DOM
- Reaktorpattern
- Stabile Plattform

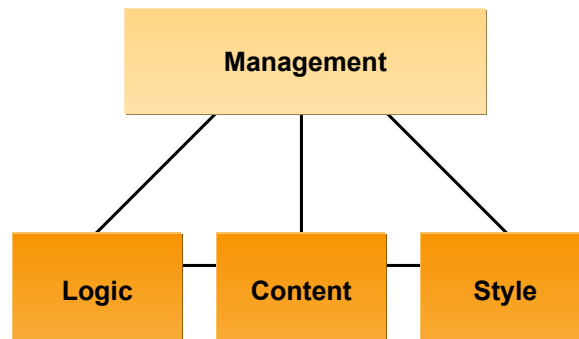
### C2

- Basiert auf SAX
  - mehr Performance
  - weniger Speicherverbrauch
- Sitemap
- Redesigned
- Höhere Modularität

## Auszug fremder Komponenten

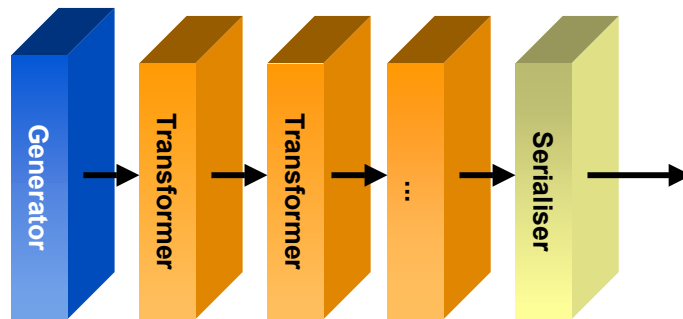
- **avalon-framework** - Komponentenframework
- **axis** - SOAP-Implementierung von Apache
- **batik** - Generierung von Grafiken mittels SVG
- **fop** - Generierung von PDF's mittels FO (Formatting Objects)
- **hsqldb** - Kleine Hypersonic Datenbank
- **logkit** - jakarta-avalon-logkit
- **maybeupload** - Zur Unterstützung zum Upload von Dateien
- **xmldb** - XML-Datenbank
- **xerces** - XML-Parser
- **xalan** - XSLT-Prozessor
- ...

## Pyramid Model of Web Contracts



Quelle: Cocoon Dokumentation

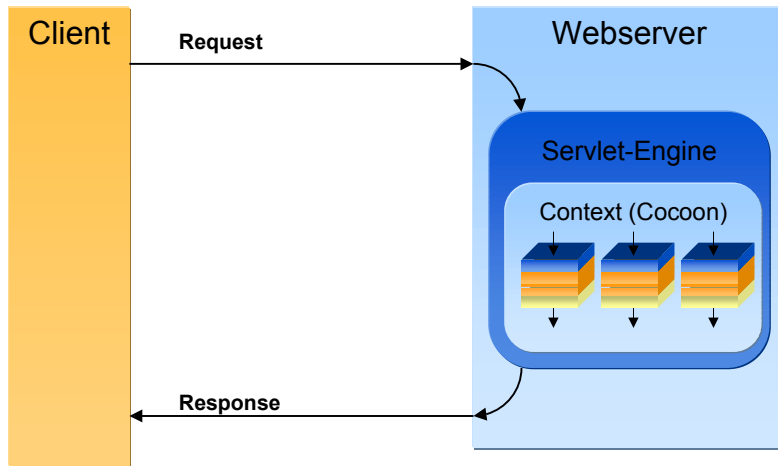
## Pipeline Model



## Inhalt

- XML based Webpublishing
- **Konfiguration mit Sitemap**
- Dynamische Seiten
- Request, Sessions & Parameter
- Logicsheets
- Beispiel Anwendungsentwicklung
- Das Pipelinemodell im Detail
- Datenbank Zugriff
- Multichannel Applications
- PDF Erzeugung
- Erzeugung von Grafiken (SVG)
- Cocoon Internas

## Komponenten



## HelloWorld in 3 Schritten - 1

Die XML-Quelle (hello.xml)

```
<?xml version="1.0"?>
<page>
  <title>Erstes Beispiel - OIO Seminar</title>
  <greeting>Hello World</greeting>
</page>
```



## HelloWorld in 3 Schritten - 2

### Das XSLT-Stylesheet

```
<?xml version="1.0"?>
<xsl:stylesheet
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  version="1.0">

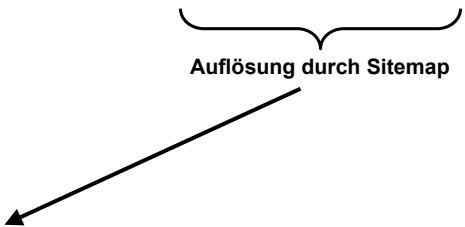
  <xsl:template match="page">
    <html>
      <body>
        <h1><xsl:value-of select="title"/></h1>
        <p><xsl:value-of select="greeting"/></p>
      </body>
    </html>
  </xsl:template>
</xsl:stylesheet>
```

## HelloWorld in 3 Schritten - 3

Definiert die Pipeline, die zu einem Seitenaufruf gehört

<http://localhost:8080/cocoon/hello/hello.html>

Auflösung durch Sitemap



```
<map:match pattern="hello/hello.html">
  <map:generate type="file" src="hello/hello.xml"/>
  <map:transform src="hello/hello.xslt"/>
  <map:serialize type="html"/>
</map:match>
```

## Aufgaben der Sitemap.xmap

- Konfiguration des Frameworks
- Definition von Komponenten
- Einstellungen der Reaktion auf Request
- Definiert Pipelines, Views, Selectors,....

## Sitemap - Definition von festen Pipes

Definiert die Pipeline, die zu einem Seitenaufruf gehört

`http://www.oio.de/cocoon/seminar/cdliste.html`



```
<map:match pattern="seminar/cdliste.html">
  <map:generate type="file" src="xml/cdliste.xml"/>
  <map:transform src="xslt/cdliste2html.xsl"/>
  <map:serialize type="html"/>
</map:match>
```

## Sitemap - Definition von festen Pipes 2

Definiert die Pipeline, die zu einem Seitenaufruf gehört

`http://www.oio.de/cocoon/seminar/cdliste.xml`



```
<map:match pattern="seminar/cdliste.xml">
  <map:generate type="file" src="xml/cdliste.xml"/>
  <map:transform src="xslt/cdlisteFilter.xsl"/>
  <map:serialize type="xml"/>
</map:match>
```

## Sitemap - Variable Pipes

Definiert die Pipeline, die zu einem Seitenaufruf gehört

`http://www.oio.de/cocoon/seminar/cdliste.html`



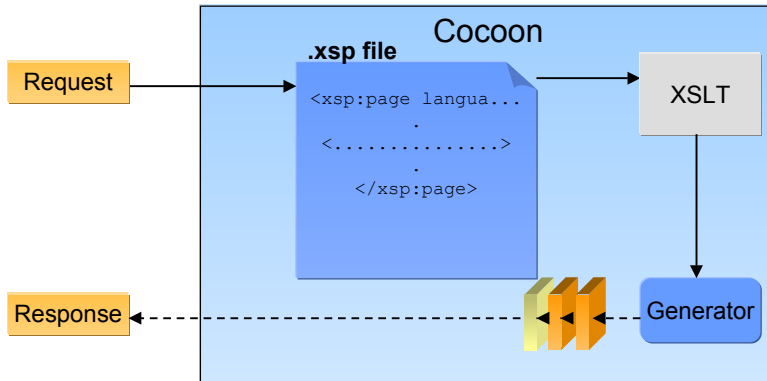
```
<map:match pattern="seminar/**/*.html">
  <map:generate type="file" src="xml/cdliste.xml"/>
  <map:transform src="xslt/{1}.xsl"/>
  <map:serialize type="html"/>
</map:match>
```

## Inhalt

- XML based Webpublishing
- Konfiguration mit Sitemap
- **Dynamische Seiten**
- Request, Sessions & Parameter
- Logicsheets
- Beispiel Anwendungsentwicklung
- Das Pipelinemodell im Detail
- Datenbank Zugriff
- Multichannel Applications
- PDF Erzeugung
- Erzeugung von Grafiken (SVG)
- Cocoon Internas

## eXtensible Server Pages

- Sind XML Dokumente
- Erzeugen XML Dokumente
- Werden mit XSLT in Source Code umgewandelt
- Der Source Code wird kompiliert
- Der neu erzeugte Code wird geladen und ausgeführt



## XSP Aufbau

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xsp:page language="java" xmlns:xsp="http://apache.org/xsp">
...
<page>
...
</page>
</xsp:page>
```

Rootelement von XSP

Rootelement der Ausgabe

## XSP Aufbau

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xsp:page language="java" xmlns:xsp="http://apache.org/xsp">

  <xsp:structure>
    <xsp:include>java.math.*</xsp:include>
  </xsp:structure>

  <page>
    ...
  </page>

</xsp:page>
```

Inkludieren von Paketen

## XSP Aufbau

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xsp:page language="java" xmlns:xsp="http://apache.org/xsp">
  <xsp:structure>
    <xsp:include>java.math.*</xsp:include>
  </xsp:structure>

  <xsp:logic>
    public BigInteger getBig() {
      return new BigInteger("10000000000000000000");
    }
  </xsp:logic>

  <page>
    ...
  </page>
</xsp:page>
```

Definition von Funktionen

## XSP Aufbau

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xsp:page language="java" xmlns:xsp="http://apache.org/xsp">
  <xsp:structure>
    <xsp:include>java.math.*</xsp:include>
  </xsp:structure>
  <xsp:logic>

  public BigInteger getBig() {
    return new BigInteger("10000000000000000000");
  }

</xsp:logic>
<page>
  <xsp:expr>getBig()</xsp:expr>
</page>
</xsp:page>
```

Aufruf der Funktion

## Generierte Java Klasse

```
import java.math.*;

public class aufbau_xsp extends XSPGenerator {
  ...
  public BigInteger getBig() {
    return new BigInteger("10000000000000000000");
  }

  public void generateContent() throws SAXException {
    AttributesImpl xspAttr = new AttributesImpl();
    this.contentHandler.startElement("", "page", "page", xspAttr);
    xspAttr.clear();
    this.characters("\n      ");
    XSPObjektHelper.xspExpr(contentHandler, getBig());
    this.characters("\n    ");
    this.contentHandler.endElement("", "page", "page");
  }

  public void generate() throws SAXException {
    this.contentHandler.startDocument();
    generateContent();
    this.contentHandler.endDocument();
  }
}
```

**vereinfacht!**

## <xsp:logic>

### <xsp:logic>

```
<xsp:logic>
    String name = person.getName();
    String vorname = person.getVorname();
    person.setVorname("Jonny");
    ...
</xsp:logic>
```

**Es erfolgt keine Ausgabe**

## <xsp:expr>

### <xsp:expr>

```
<counter><xsp:expr>i</xsp:expr></counter>
<name>
    <xsp:expr>person.getName()</xsp:expr>,
    <xsp:expr>person.getVorname()</xsp:expr>
</name>
```

**Ausgabe**

↓

```
<counter>5</counter>
<name>Erdnuss, Jonny</name>
```



## <xsp:content>

### <xsp:content> - Ausgabe in Logic-Blöcken

```
<xsp:logic>
  if (y==x) {
    <xsp:content>
      <input type="hidden" name="type" value="work"/>
    </xsp:content>
  }
</xsp:logic>
```

Wenn x=y

#### Ausgabe

```
...
<input type="hidden" name="type" value="work"/>
...
```

## <xsp:content> - Verschachtelte Ausgabe

### <xsp:content>

```
<xsp:logic>
  if (y==x) {
    <xsp:content>
      <b>Gehalt: <xsp:expr>getSalary()</xsp:expr></b>
    </xsp:content>
  }
</xsp:logic>
```

Wenn x=y

#### Ausgabe

```
...
<b>Gehalt: 1300</b>
...
```

## <xsp:attribute>

```
<xsp:attribute name="xyz">
```

```
<input type="text" name="name">  
  <xsp:attribute name="value">  
    <xsp:expr>getName()</xsp:expr>  
  </xsp:attribute>  
</input>
```

### Ausgabe

...

```
<input type="text" name="name" value="Jonny" />
```

...

## XSP Tags

<xsp:page>	XSP Root Element
<xsp:structure>	beinhaltet Deklarationen wie <xsp:include>
<xsp:logic>	Enthält Code und „User Markup“
<xsp:content>	Ausgabe innerhalb <xsp:logic>
<xsp:expr>	Enthält Ausdruck der Text liefert
<xsp:element>	Erzeugt Element
<xsp:attribute>	Fügt Attribut hinzu
<xsp:comment>	Erzeugt XML Kommentar
<xsp:pi>	Erzeugt Processing Instruction

- XML based Webpublishing
- Konfiguration mit Sitemap
- Dynamische Seiten
- **Request, Sessions & Parameter**
- Logicsheets
- Beispiel Anwendungsentwicklung
- Das Pipelinemodell im Detail
- Datenbank Zugriff
- Multichannel Applications
- PDF Erzeugung
- Erzeugung von Grafiken (SVG)
- Cocoon Internas

## Interface ServletResponse

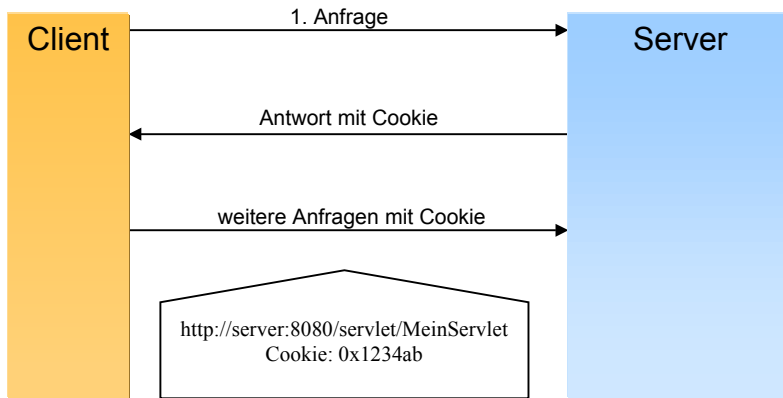
### Zur Verarbeitung der Ausgabe

(org.apache.cocoon.environment)

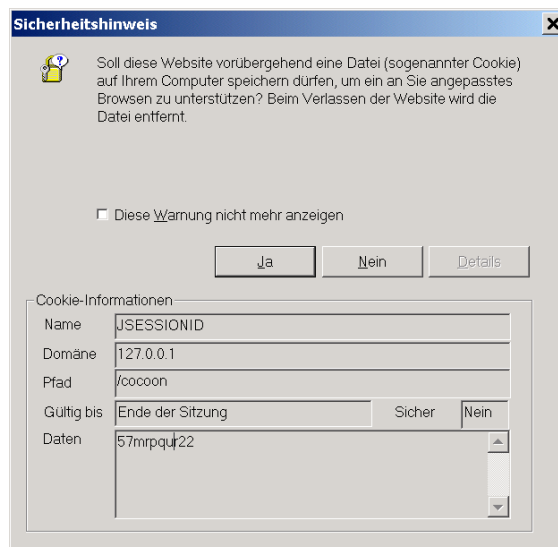
- **void sendRedirect(String url)**
- **void sendError(int number)**
- PrintWriter getWriter()
- addCookie(Cookie c)
- addHeader(String name, String value)

...

# Servlets und Cookies



# Cookies



## Session - Übergabe in der URL

The screenshot shows two browser windows. The left window displays a menu with 'Personen' highlighted. An arrow points from 'Personen' to the right window. The right window shows the URL `http://127.0.0.1:8080/cocoon/seminar/personedit.jsp?PER_ID=2` with the `PER_ID=2` part circled in red. Below the URL bar is a form with fields for NAME (Rauh), VORNAME (Manfred), STRASSE (Westendstr. 34), ORT (60315 Frankfurt), TEL ((069) 186468), and MAIL (rauh@freemailer.com), and an 'abschicken' button.

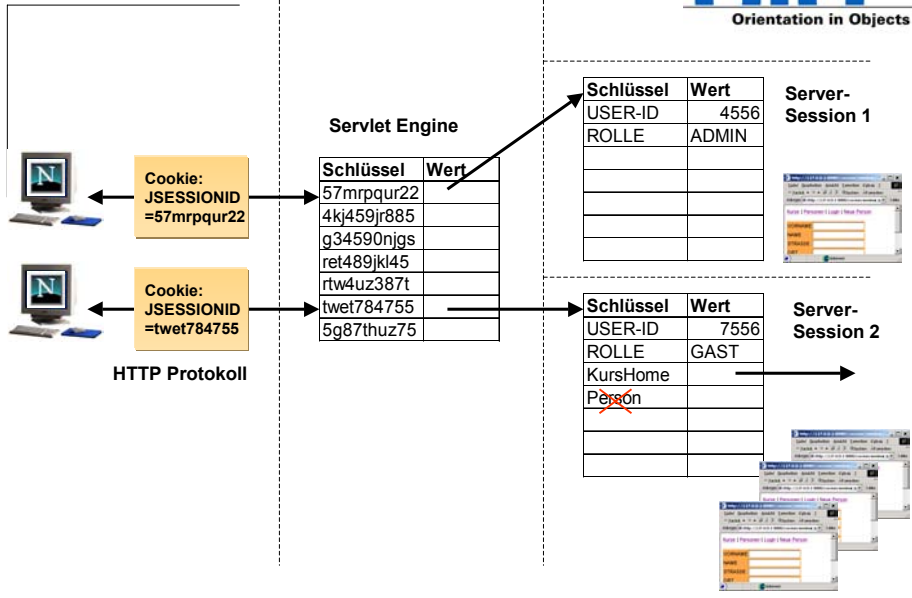
XML-basierte Web Entwicklung mit Cocoon 2 © 2001-2002 Orientation in Objects GmbH

## Sessions - Hidden Field in Formularen

The screenshot shows a browser window with the same form as in the previous slide. A dashed arrow points from the 'abschicken' button to the code block below.

```
<input type="hidden" name="PER_ID" value="2">
```

## Sessions - ID im Cookie



## Sessions - SSL

- SSL ist Verbindungsorientiert
  - TCP-Verbindung wird nicht nach Abruf jeder Seite abgebaut
  - Wegen Sicherheit kein Abbau der Verbindung
- SSL stellt Sessionmechanismus schon zur Verfügung

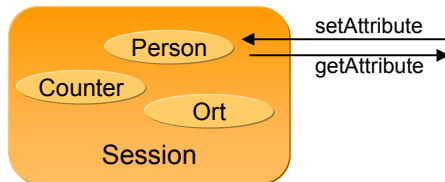
## Das Sessionobjekt

- Wert In der Session ablegen

```
request.getSession().setAttribute („Ort1“, o);  
request.getSession().setAttribute („Ort2“, s);
```

- Wert aus Session lesen

```
Object o=request.getSession().getAttribute („Ort1“);  
String s=request.getSession().getAttribute („Ort2“).toString();
```



## Inhalt

- XML based Webpublishing
- Konfiguration mit Sitemap
- Dynamische Seiten
- Request, Sessions & Parameter
- **Logicsheets**
- Beispiel Anwendungsentwicklung
- Das Pipelinemodell im Detail
- Datenbank Zugriff
- Multichannel Applications
- PDF Erzeugung
- Erzeugung von Grafiken (SVG)
- Cocoon Internas

## Was sind LogicSheets?

- Dienen der Abstraktion innerhalb von XSP
- Java-Code wird durch Tags ersetzt
- Wiederverwendung verschiedener Funktionen
- Bessere Lesbarkeit der Seite

```
<util:time-of-day>
```

## Funktionsweise eines Logicheets

```
<util:time-of-day>
  <util:parameter name="format">
    <request:get-parameter name="time-format" default="hh:mm:ss"/>
  </util:parameter>
</util:time-of-day>
```

```
<xsp:expr>
  SimpleDateFormat.getInstance().format(
    new Date(),
    <request:get-parameter name="time-format" default="hh:mm:ss"/>
  )
</xsp:expr>
```

```
<xsp:expr>
  SimpleDateFormat.getInstance().format(
    new Date(),
    XSPRequestHelper.getParameter("name", "hh:mm:ss")
  )
</xsp:expr>
```



## Session als Logicsheet

- Definition der Verwendung

```
<xsp:page language="java"
xmlns:xsp="http://apache.org/xsp"
xmlns:session="http://apache.org/xsp/session/2.0">
```
- Wert In der Session ablegen

```
<session:set-attribute name="Ort">
Mannheim
</session:set-attribute>
```
- Wert aus Session lesen

```
<session:get-attribute name="Ort"/>
```
- ...

## XSP: Logicsheets (Taglibs)

- ESQL
- MAIL
- REQUEST
- RESPONSE
- LOG
- SESSION
- UTIL

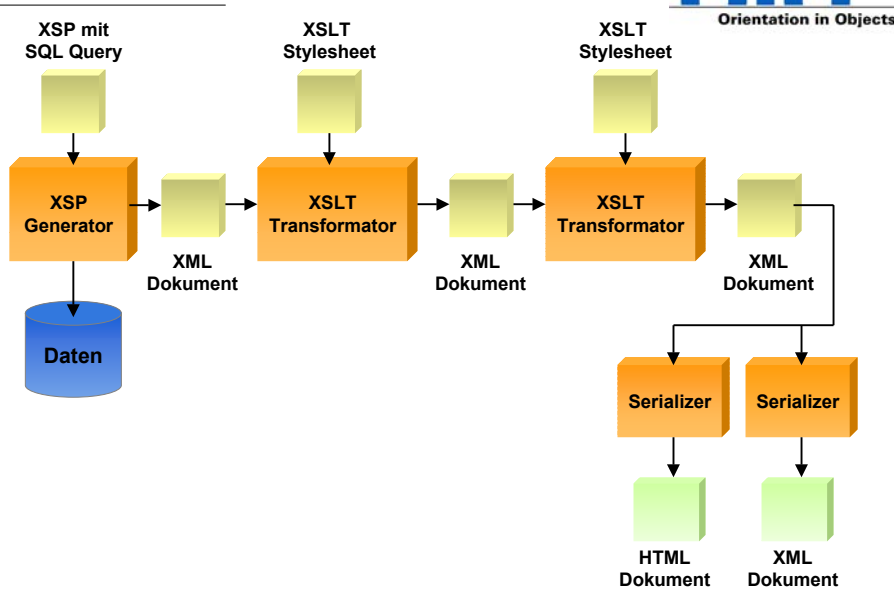
### 3rd Party Logicsheets (Taglibs)

Name	Zweck
DB Prism	Emuliert Oracle Web Application Server (für Cartridges)
LDAP	Zugriff auf Verzeichnisdienste
Auth	Authentifizierung und Passwortabfrage
SOAP	XML-basierte RPCs
IMAP	Mailversand über IMAP Protokoll

## Inhalt

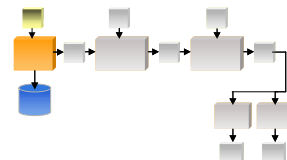
- XML based Webpublishing
- Konfiguration mit Sitemap
- Dynamische Seiten
- Request, Sessions & Parameter
- Logicsheets
- **Beispiel Anwendungsentwicklung**
- Das Pipelinemodell im Detail
- Datenbank Zugriff
- Multichannel Applications
- PDF Erzeugung
- Erzeugung von Grafiken (SVG)
- Cocoon Internas

## Ablauf



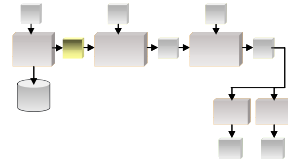
## XSP mit SQL Query

```
...  
<page><content>  
<seminare>  
  <esql:connection>  
  <esql:pool>seminar</esql:pool>  
  <esql:execute-query>  
    <esql:query>select id, name from seminar</esql:query>  
    <esql:results>  
      <esql:row-results>  
        <seminar>  
          <esql:get-columns/>  
        </seminar>  
      </esql:row-results>  
    </esql:results>  
  </esql:execute-query>  
</esql:connection>  
</seminare>  
</content></page>  
...
```



## XML Output der XSP

```
<?xml version="1.0" encoding="UTF-8" ?>
<page>
  <content>
    <seminare>
      <seminar>
        <id>X.EINF</id>
        <name>XML Einführung</name>
      </seminar>
      <seminar>
        <id>X.JAVA</id>
        <name>XML mit Java</name>
      </seminar>
    </seminare>
  </content>
</page>
```



## XSLT umwandeln in Tabelle

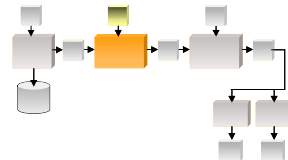
```
<?xml version="1.0" encoding="utf-8"?>
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

  <xsl:template match="seminare">
    <tabelle>
      <xsl:apply-templates/>
    </tabelle>
  </xsl:template>

  <xsl:template match="seminar">
    <zeile>
      <data><xsl:value-of select="name"/></data>
    </zeile>
  </xsl:template>

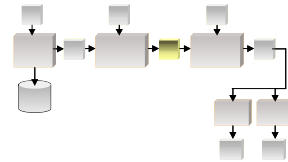
  ...

</xsl:stylesheet>
```



## XML Output mit Tabelle

```
<page>
  <content>
    <tabelle>
      <zeile>
        <data>XML Einführung</data>
      </zeile>
      <zeile>
        <data>XML mit Java</data>
      </zeile>
    </tabelle>
  </content>
</page>
```



## XSLT zur HTML Generierung

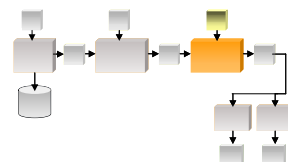
```
<?xml version="1.0"?>

<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

  <xsl:template match="page">
    <html>
      <body>
        <xsl:apply-templates select="content"/>
        <br/>
        generated by Cocoon
      </body>
    </html>
  </xsl:template>

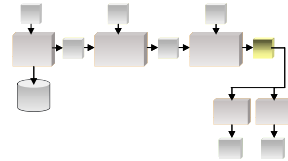
  ...

</xsl:stylesheet>
```



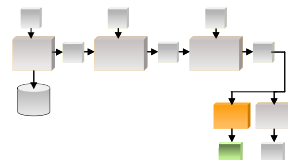
## XML Output

```
<html>
  <body>
    <table>
      <tr>
        <td>XML Einf&uuml;hrung</td>
      </tr>
      <tr>
        <td>XML mit Java</td>
      </tr>
    </table>
    <br/>
    generated by Cocoon
  </body>
</html>
```



## HTML Output

```
<html>
  <body>
    <table>
      <tr>
        <td>XML Einf&uuml;hrung</td>
      </tr>
      <tr>
        <td>XML mit Java</td>
      </tr>
    </table>
    <br>
    generated by Cocoon
  </body>
</html>
```



## Beispiel Anwendung

The screenshot displays two browser windows. The left window shows a page titled 'Bioenergetisches Salzteigbacken' with a 'Teilnehmer' section listing names: Gaby Biegel, Steffen Sander, and Simon Mann. The right window shows a page titled 'Seminar' with a table of seminars.

Datum	Seminar	Art
14. - 16.5.	Bioenergetisches Salzteigbacken	
14. - 17.5.	Yoga Einführung	
21. - 22.5.	Tauchen	
3. - 5.6.	Bioenergetisches Salzteigbacken	
14. - 17.6.	Yoga Einführung	
25. - 29.6.	Drachenfliegen	
4. - 6.7.	Bioenergetisches Salzteigbacken	
16. - 17.7.	Tauchen	
23. - 26.7.	Yoga Einführung	

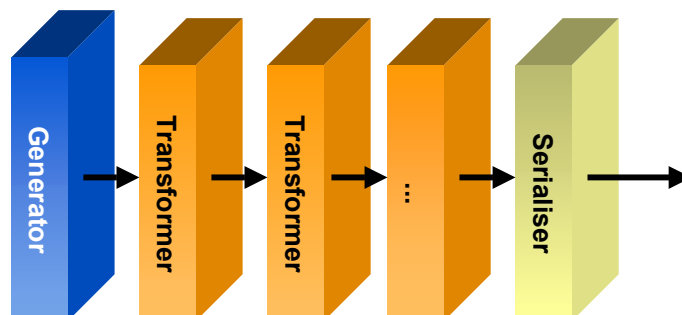
## Teilweise Installation der Anwendung



## Inhalt

- XML based Webpublishing
- Konfiguration mit Sitemap
- Dynamische Seiten
- Request, Sessions & Parameter
- Logicsheets
- Beispiel Anwendungsentwicklung
- **Das Pipelinemodell im Detail**
- Datenbank Zugriff
- Multichannel Applications
- PDF Erzeugung
- Erzeugung von Grafiken (SVG)
- Cocoon Internas

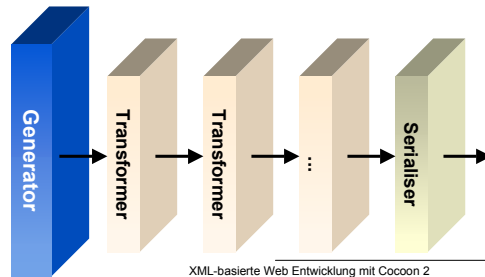
## Pipeline Model





## Standard Generatoren

Generator	Beschreibung
DirectoryGenerator	Erzeugt XML Verzeichnisliste
FileGenerator	Liebt XML File ein und parst es
HTMLGenerator	Liebt HTML File ein und generiert daraus SAX Ereignisse im XHTML Format
ImageDirectoryGenerator	Um Bilder erweiterter DirectoryGenerator
PhpGenerator	Liebt von PHP ein über PHP Servlet Schnittstelle
ServerPagesGenerator	Liebt XML File und interpretiert Inhalt als eXtensible Server Page
StatusGenerator	Liefert aktuelle Informationen des Systems

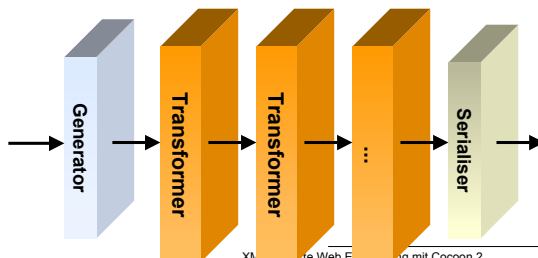


XML-basierte Web Entwicklung mit Cocoon 2

© 2001-2002 Orientation in Objects GmbH

## Standard Transformer

Transformer	Zweck
XSLT Transformer	Für XSLT Transformationen
Log Transformer	Kann in eine Pipeline eingeklinkt werden. Protokolliert die SAX Ereignisse
SQL Transformer	Für SQL Abfragen
i18n Transformer	Für Internationalisierung
XIncludeTransformer	Einbetten von XML Dokumenten

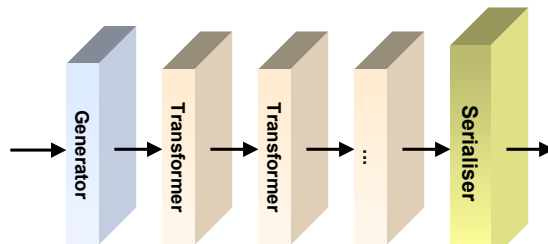


XML-basierte Web Entwicklung mit Cocoon 2

© 2001-2002 Orientation in Objects GmbH

## Standard Serializer

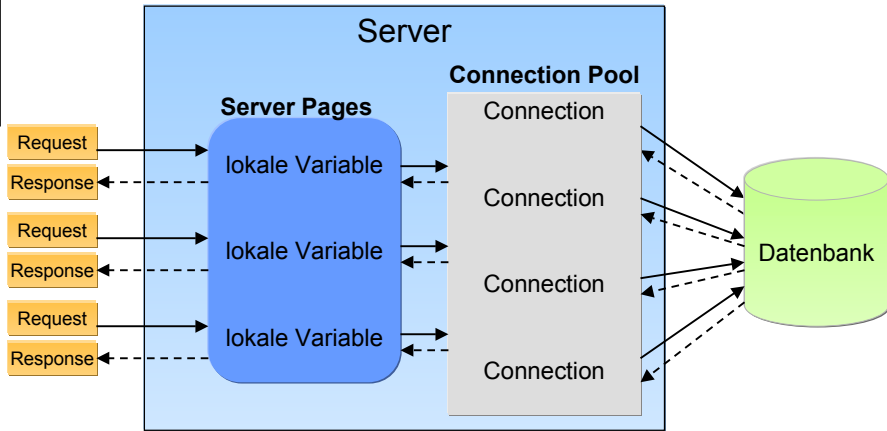
Serializer	Erzeugt
HTMLSerializer	HTML Dokumente
XMLSerializer	XML Dokumente
FOPSerializer	PDF Dokumente
SVGSerializer	SVG Grafik
TextSerializer	Textfile



## Inhalt

- XML based Webpublishing
- Konfiguration mit Sitemap
- Dynamische Seiten
- Request, Sessions & Parameter
- Logicsheets
- Beispiel Anwendungsentwicklung
- Das Pipelinemodell im Detail
- **Datenbank Zugriff**
- Multichannel Applications
- PDF Erzeugung
- Erzeugung von Grafiken (SVG)
- Cocoon Internas

# Zugriff auf Datenbanken



# Hypersonic Database Manager

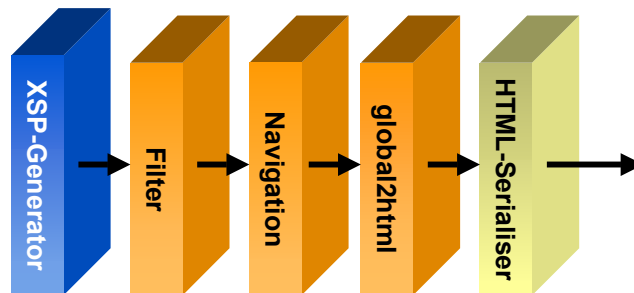
The screenshot shows the Hypersonic Database Manager interface. The title bar reads 'Hypersonic Database Manager'. The menu bar includes 'File', 'View', 'Command', 'Recent', and 'Options'. The main window displays a table with the following data:

NAME	PASSWORT	VORNAME	TEL	ORT	MAIL	STRASSE	PER_ID	ROI
Vogel	t1	Peter	(0621) 589966	68309 Mannheim	vogel.peter@t-online.de	Weinheimer Str. 8	1	GAS
Biegel	t4	Gaby	(06151) 5189798	53250 Bonn	gaby@freemailer.com	Mozartplatz 1	4	GAS
Sander	t5	Steffen	(06151) 15748	53250 Bonn	steffen@mailfornothing.com	Trierer Str. 27	5	GAS
Zeller	t6	Nicole	(0621) 15888	68312 Mannheim	zeller@netservices.com	Adenauer-Ring 1	6	GAS
Mann	t8	Simon	(0721) 558898	76138 Karlsruhe	simon@freaka.de	Amsehweg 78	8	GAS
Rieger	t7	Yvonne	(030) 745109799	10189 Berlin	yr@freemailer.com	Hauptstr. 4	7	GAS
Rauh	t2	Manfred	(069) 186468	60315 Frankfurt	rauh@freemailer.com	Westendstr. 34	2	ADM
Heller	t3	Silke	(07131) 995566	74076 Heilbronn	hs@hellerweb.de	Neckarweg 38	3	GAS

## Datenbankzugriff mit ESQL

```
<esql:connection>
  <esql:pool>seminar</esql:pool>
  <esql:execute-query>
    <esql:query>select * from SEMINARART</esql:query>
    <esql:results>
      <seminararten>
        <esql:row-results>
          <seminarart>
            <esql:get-columns/>
          </seminarart>
        </esql:row-results>
      </seminararten>
    </esql:results>
  </esql:execute-query>
</esql:connection>
```

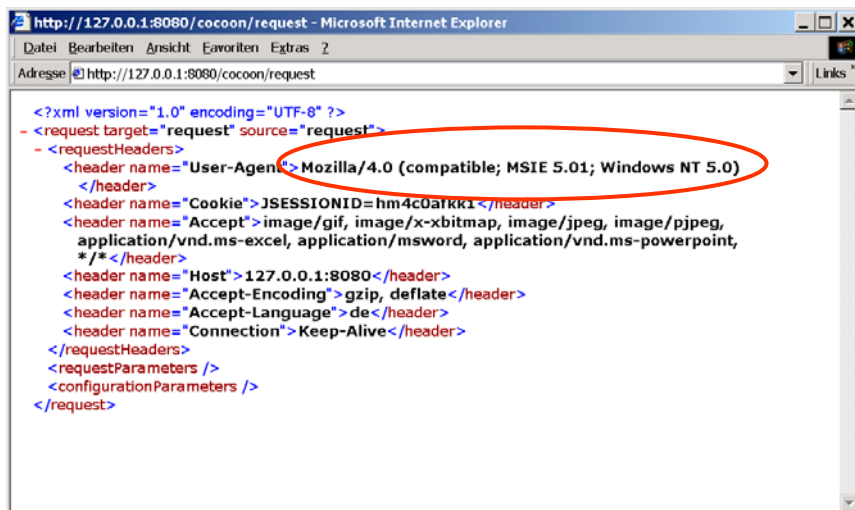
## Pipeline Aufbau



```
<map:match pattern="seminar/**/*.xsp">
  <map:generate type="serverpages" src="seminar/xsp/{1}.xsp"/>
  <map:transform src="seminar/style/{1}-filter.xsl"/>
  <map:transform src="seminar/style/navigation.xsl"/>
  <map:transform src="seminar/style/global2html.xsl"/>
  <map:serialize type="html"/>
</map:match>
```

- XML based Webpublishing
- Konfiguration mit Sitemap
- Dynamische Seiten
- Request, Sessions & Parameter
- Logicsheets
- Beispiel Anwendungsentwicklung
- Das Pipelinemodell im Detail
- Datenbank Zugriff
- **Multichannel Applications**
- PDF Erzeugung
- Erzeugung von Grafiken (SVG)
- Cocoon Internas

## HTTP Request User-Agent



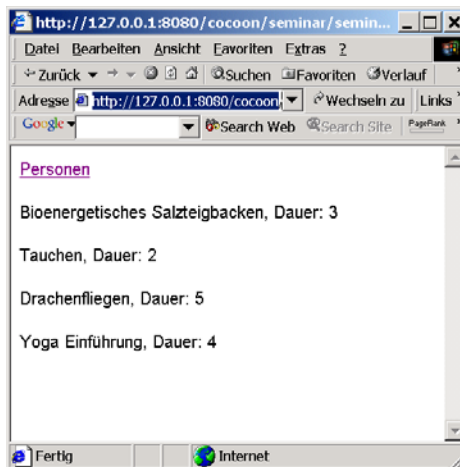
```
<?xml version="1.0" encoding="UTF-8" ?>
- <request target="request" source="request">
- <requestHeaders>
  <header name="User-Agent">Mozilla/4.0 (compatible; MSIE 5.01; Windows NT 5.0)</header>
  <header name="Cookie">JSESSIONID=hm4c0atkk1</header>
  <header name="Accept">image/gif, image/x-xbitmap, image/jpeg, image/pjpeg,
application/vnd.ms-excel, application/msword, application/vnd.ms-powerpoint,
*/*</header>
  <header name="Host">127.0.0.1:8080</header>
  <header name="Accept-Encoding">gzip, deflate</header>
  <header name="Accept-Language">de</header>
  <header name="Connection">Keep-Alive</header>
</requestHeaders>
<requestParameters />
<configurationParameters />
</request>
```

## Selectoren

```
<requestHeaders>
  <header name="User-Agent">Nokia-WAP-Toolkit/2.1</header>
  <header name="Date">Tue, 01 May 2001 13:59:11
GMT</header>
  <header name="Accept">text/vnd.wap.wml,
text/vnd.wap.wmlscript,application/vnd.wap.wmlc,
application/vnd.wap.wmlscriptc,text/x-vCard,
text/x-vcard,text/x-vcal,text/x-vCalendar,
text/x-vcalendar,image/vnd.wap.wbmp,
application/vnd.wap.wbxml,
application/vnd.wap.multipart.mixed,multipart/mixed,
image/gif,image/jpeg</header>
  <header name="Pragma">no-cache</header>
  <header name="Host">127.0.0.1:8080</header>
  <header name="Accept-Language">de</header>
  <header name="Accept-Charset">UTF-8, ISO-8859-1, ISO-
10646-UCS-2</header>
  <header name="Cache-Control">no-cache</header>
</requestHeaders>
```

### Request Header des Nokia Wap Toolkits

## Multichannel Applications



## Anwenden des Multichannelstylesheets



## Generierung im Batch

### **Vorteile:**

- Schnelle Bearbeitung von Abrufen
- Konventioneller Web Server ist ausreichend
- Robuster im Betrieb

### **Nachteile:**

- Keine dynamische Seitengenerierung

- XML based Webpublishing
- Konfiguration mit Sitemap
- Dynamische Seiten
- Request, Sessions & Parameter
- Logicsheets
- Beispiel Anwendungsentwicklung
- Das Pipelinemodell im Detail
- Datenbank Zugriff
- Multichannel Applications
- **PDF Erzeugung**
- Erzeugung von Grafiken (SVG)
- Cocoon Internas

- XSL-FO stellt die zweite Hälfte der XSL Spec.
- besteht aus 56 definierten FO-Elementen und über 100 Attributen
- Formatting Objects beheben die Nachteile einer Bildschirmausgabe, die vom Browser o.ä. abhängt und liefert für die Druckausgabe optimierte Dokumente
- FOP ist eine Implementierung der ASF



## FO Namespace

- Verwendung des Kürzels **fo**:
- Namespacedeklaration:  
`xmlns:fo="http://www.w3.org/1999/XSL/Format"`
- Kombination fo: und xsl: Namespaces möglich :)

## Aufbau eines FO-Dokuments

```
<fo:root xmlns:fo="http://www.w3.org/1999/XSL/Format">
  <fo:layout-master-set>
    <fo:simple-page-master name="my-page">
      <fo:region-body/>
    </fo:simple-page-master>
  </fo:layout-master-set>
  <fo:page-sequence master-name="my-page">
    <fo:flow flow-name="xsl-region-body">
      <fo:block>Hier steht Text...</fo:block>
    </fo:flow>
  </fo:page-sequence>
</fo:root>
```

## Eintrag in die Sitemap

```
<map:pipeline>
  <map:match pattern="seminar/kurse.pdf">
    <map:generate type="..." src="..." />
    ...
    <map:serialize type="fo2pdf" />
  </map:match>
</map:pipeline>
```

## Übung

- Erstellen einer FO Seite



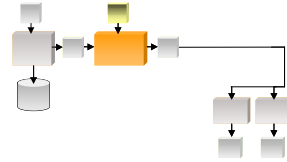
# Stylesheet für XSL:FO

```
<xsl:template match="/">
  <fo:root xmlns:fo="http://www.w3.org/1999/XSL/Format">
    <!-- Hier Seitenformat festlegen (fehlt!) -->

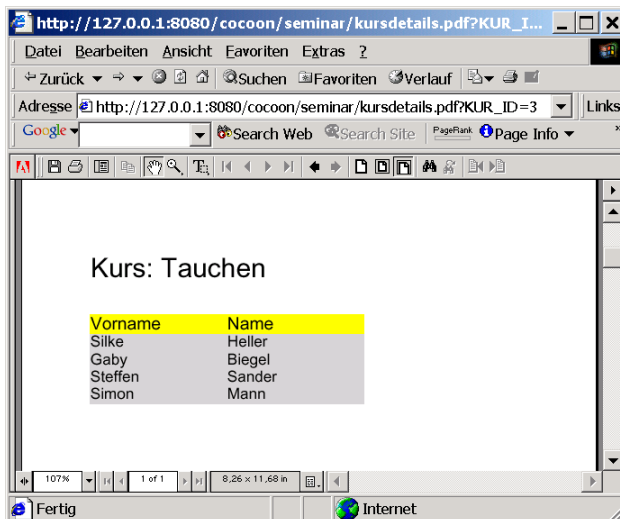
    <fo:page-sequence master-name="all">
      <fo:static-content flow-name="xsl-region-after"/>
      <fo:flow flow-name="xsl-region-body">
        <fo:block font-size="36pt" space-before.optimum="24pt"
          text-align="center">Seminare</fo:block>
        <xsl:apply-templates/>
      </fo:flow>
    </fo:page-sequence>
  </fo:root>
</xsl:template>

<xsl:template match="seminar">
  <fo:block font-size="12pt" space-before.optimum="12pt" text-align="left">
    <xsl:value-of select="name"/></fo:block>
</xsl:template>

...
```



# Ausgabe in PDF



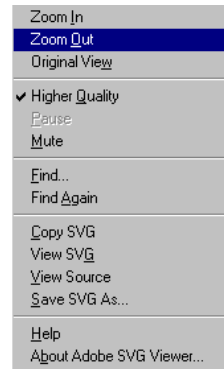
- XML based Webpublishing
- Konfiguration mit Sitemap
- Dynamische Seiten
- Request, Sessions & Parameter
- Logicsheets
- Beispiel Anwendungsentwicklung
- Das Pipelinemodell im Detail
- Datenbank Zugriff
- Multichannel Applications
- PDF Erzeugung
- **Erzeugung von Grafiken (SVG)**
- Cocoon Internas

## Was ist SVG - Scalable Vector Graphics?

- Standardisiertes Grafikformat
- Aktueller Stand:
  - SVG 1.0 W3C 4. September 2001
- 2D Vektor und mixed Vektor/Raster Grafik
- XML Anwendung
- SVG Grafiken können animiert, interaktiv und dynamisch sein

## SVG Viewer & Editoren

- Adobe
  - [www.adobe.de/svg](http://www.adobe.de/svg)
- IBM SVG View
  - <http://www.alphaworks.ibm.com/tech/svgview>
- Apache Batik Project
  - <http://xml.apache.org/batik>
- Jasc WebDraw
  - <http://www.jasc.com/products/webdraw>



## Hello World als SVG



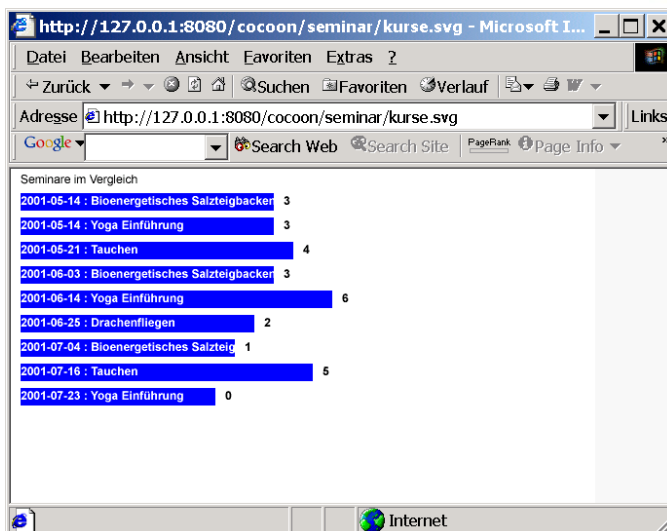
The text "Hello Scalable Vector Graphics" is displayed in a blue, wavy font that follows the curve of a red path. The path is a simple arc that starts and ends at the same point, forming a shallow curve.

- Beispiel ist eine SVG Grafik (importiert in PowerPoint)
- Ausrichtung eines Textes an einem Pfad

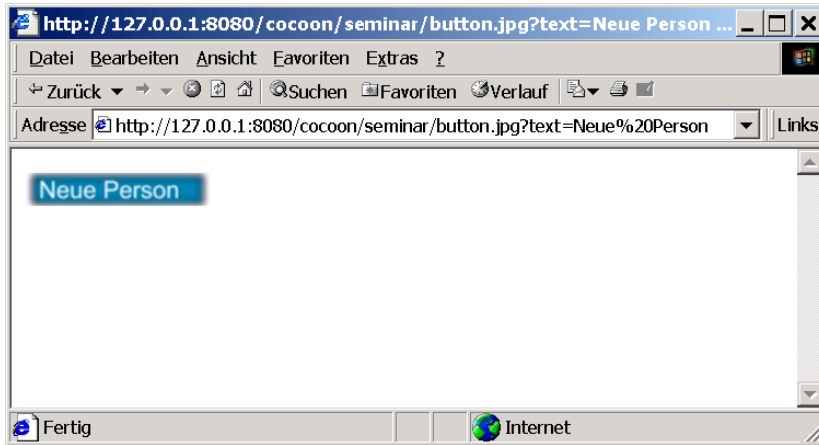
## Sourcecode zu „Hello World“

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 20001102//EN"
  "http://www.w3.org/TR/2000/CR-SVG-20001102/DTD/svg-20001102.dtd">
<svg width="10cm" height="3cm" viewBox="0 0 1000 300">
  <defs>
    <path id="MyPath"
      d="M 100 200 C 200 100 300 0 400 100
        C 500 200 600 300 700 200 C 800 100 900 100 900 100"/>
  </defs>
  <desc>Hello World</desc>
  <use xlink:href="#MyPath" style="fill:none; stroke:red" />
  <text style="font-family:Verdana; font-size:60; fill:blue">
    <textPath xlink:href="#MyPath">
      Hello Scalable Vector Graphics
    </textPath>
  </text>
</svg>
```

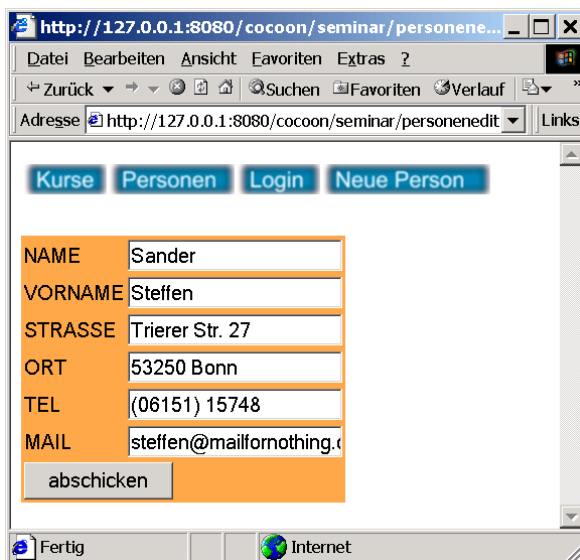
## Dynamisches SVG



# Dynamischer Image Button



# Dynamisches Menu



## Inhalt

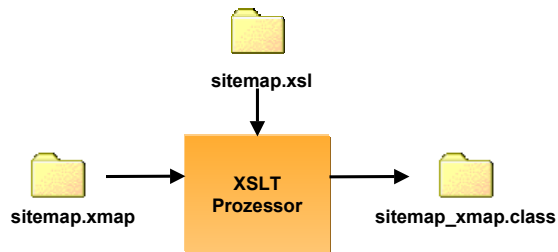
- XML based Webpublishing
- Konfiguration mit Sitemap
- Dynamische Seiten
- Request, Sessions & Parameter
- Logicsheets
- Beispiel Anwendungsentwicklung
- Das Pipelinemodell im Detail
- Datenbank Zugriff
- Multichannel Applications
- PDF Erzeugung
- Erzeugung von Grafiken (SVG)
- **Cocoon Internas**

## Performance

- Cocoon2 verwendet SAX
- Darstellung kleiner Informationsmengen
- Caching
- Jikes
- XSLT Optimierungen
  - z.B. Gruppierungen
  - Keys (Indizies)



## Generierung der Sitemap Klasse



## Auszug aus xsp.xml

```
/* User Imports */
<xsl:for-each select="xsp:structure/xsp:include">
import <xsl:value-of select="."/;>;
</xsl:for-each>

/**
 * Generated by XSP. Edit at your own risk, :-)
 */
public class <xsl:value-of select="@file-name"/> extends XSPGenerator {

...

/* User Class Declarations */
<xsl:apply-templates select="xsp:logic"/>

...
```

## XML based Webdevelopment



### Vorteile

- Trennung von Daten, Logik und Style
- Multichannelfähigkeit
- Gute Aufteilung der Arbeiten im Team
- Kurze Entwicklungszeiten
- Flexibilität

### Nachteile

- Overhead
- Mehr Komponenten
  - Verwaltung, Management
- Geschäftslogik ist nicht in Geschäftsobjekten gekapselt

## Literatur und Quellen



- Specification JSP 1.1  
<http://java.sun.com/products/jsp/download.html>
- Specification XSL 1.0 (candidate recommendation)  
<http://www.w3.org/TR/xsl/>
- Specification XSLT 1.0 (recommendation)  
<http://www.w3.org/TR/xslt.html>
- Specification XSLT 1.1 (working draft)  
<http://www.w3.org/TR/xslt11/>

## Literatur und Quellen

- Apache Software Foundation  
<http://xml.apache.org>
- Adding XML Capabilities with Cocoon (Stefano Mazzocchi)  
<http://xml.apache.org/cocoon/>
- Lutris open source enterprise  
<http://www.lutris.com>
- Enhydra Open Source for E-Business  
<http://www.enhydra.org>
- AxKit XML Application Server  
<http://www.axkit.org>
- Bean Taglibrary  
<http://www.oio.de/jax.htm>
- WML Specification  
<http://www.oasis-open.org/cover/xml.html>

## Fragen?

