

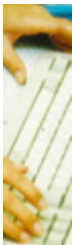


GHS

Erstellen von PDF-Dokumenten für Business-Anwendungen mit XSL-FO



 **JAVAFORUM** 2007
stuttgart am 5. Juli 2007



Ich über mich



Rudi Gast (rgast@ghs-software.de)

seit 2000 angestellt bei der GHS

Tätigkeiten:

- Projektleitung
- Softwareentwicklung
- F&E

ca. 20 Jahre Erfahrung in der Softwareentwicklung



Agenda



- **Einführung**
- **Anforderungen an ein Drucksystem**
- **XSL als Basis**
- **Formatierer FOP**
- **Druckprozess**
- **Erweiterungen**
- **Bewertung**
- **Erfahrungen, Ausblick**
- **Fragen & Diskussion**

Motivation

- **Drucken mit Java API ist zu unflexibel**
 - hohe Aufwände bei Anpassungen und Erweiterungen
 - hochwertige Druckergebnisse nur schwer erzielbar
 - kein kundenspezifischer Druck (Corporate Identity)

- **Drucken über Browser nicht ausreichend**
 - wenig Einfluss auf die Seitengestaltung (-umbruch)
 - Features wie Fuß- und Kopfzeile, Nummerierung, Inhaltsverzeichnis nicht möglich

- **Probleme aus der Praxis**
 - Installationsprobleme
 - Plattformprobleme
 - Abhängigkeit zu weiteren Anwendungen/Treibern
 - unterschiedliche Darstellung der Dokumente durch Office-Anwendungen, z.B. bei RTF Dokumenten



Anforderungen an ein Drucksystem

- **plattformunabhängig**
- **unabhängig von weiteren Anwendungen
(Textverarbeitungssystemen und/oder Druckertreiber)**
- **qualitativ hochwertige Ausdrücke**
- **einfache Wartung und Pflege**
- **kundenspezifischer Druck**
- **kosteneffizient**
- **Vorlagen einfach gestaltbar**

XSL als Basis (1/4)



Was ist XSL?

- **XSL** eXtensible **S**tylesheet **L**anguage
- ein **W3C Standard**
- dient zur **Darstellung von XML Dokumenten**
- ein **neutrales Format, unabhängig von Form und Medium in dem es dargestellt wird.**
- **XSL untergliedert sich in drei Teile**
 - **XSL Transformations (XSLT)**
 - **XML Path Language (XPath)**
 - **XSL Formating Objects (XSL-FO)**

XSL als Basis (2/4) XSLT



- **Ausgabe als XML, HTML oder Text möglich (XML ist Standard)**
- **Template Regeln zur Transformation Quell- in Ergebnisbaum**
- **Knotenwerte**
- **Schleifen**
- **Bedingungen**
- **Sortierung (alphabetisch oder numerisch)**
- **Nummerierung**
- **Variablen und Parameter**
- **Modularisierung (import, include)**

XSL als Basis (3/4) XSL-FO



Sprache für Seitengestaltung und Umbruch

▪ **Seitenlayout**

- Höhe, Breite, Ränder, Druckbereiche, Schreibrichtung, Ausrichtung, Maßeinheiten
- Seitenfolgen definieren und festlegen, usw.

▪ **Blöcke**

- Grundlage der Seitenarchitektur
- unbegrenzt verschachtelbar
- Aufnahme von Elemente
- Festlegung von Font, Schriftgröße, Rahmen, Farben, usw.

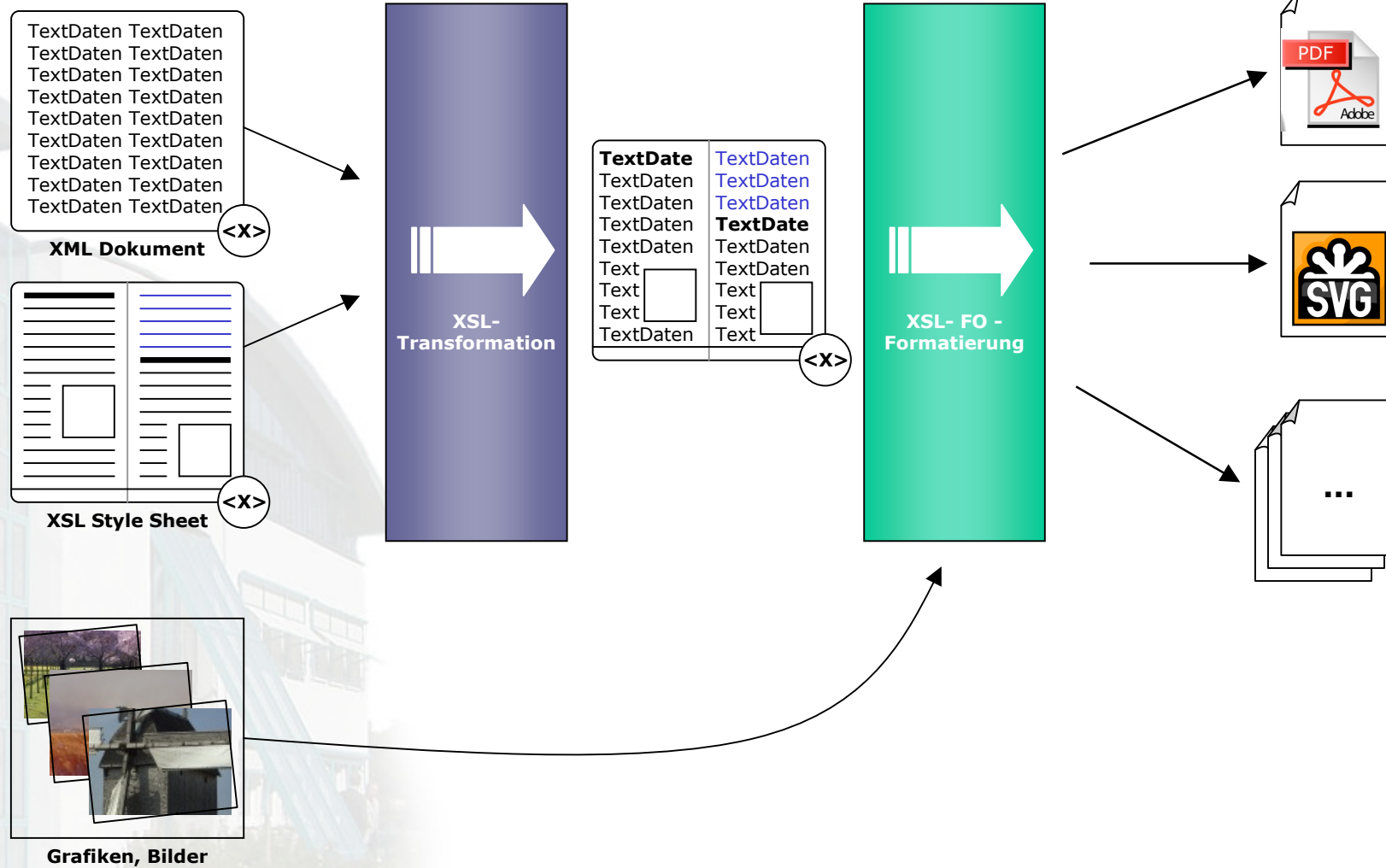
▪ **Rahmen, Ränder, (Innen)Abstände**

▪ **Inzeilige Formatierung**

▪ **Tabellen, geordnete und ungeordnete Listen, Aufzählungen**

▪ **Fußnoten, Querverweise, Hyperlinks**

XSL als Basis (4/4) Das Konzept



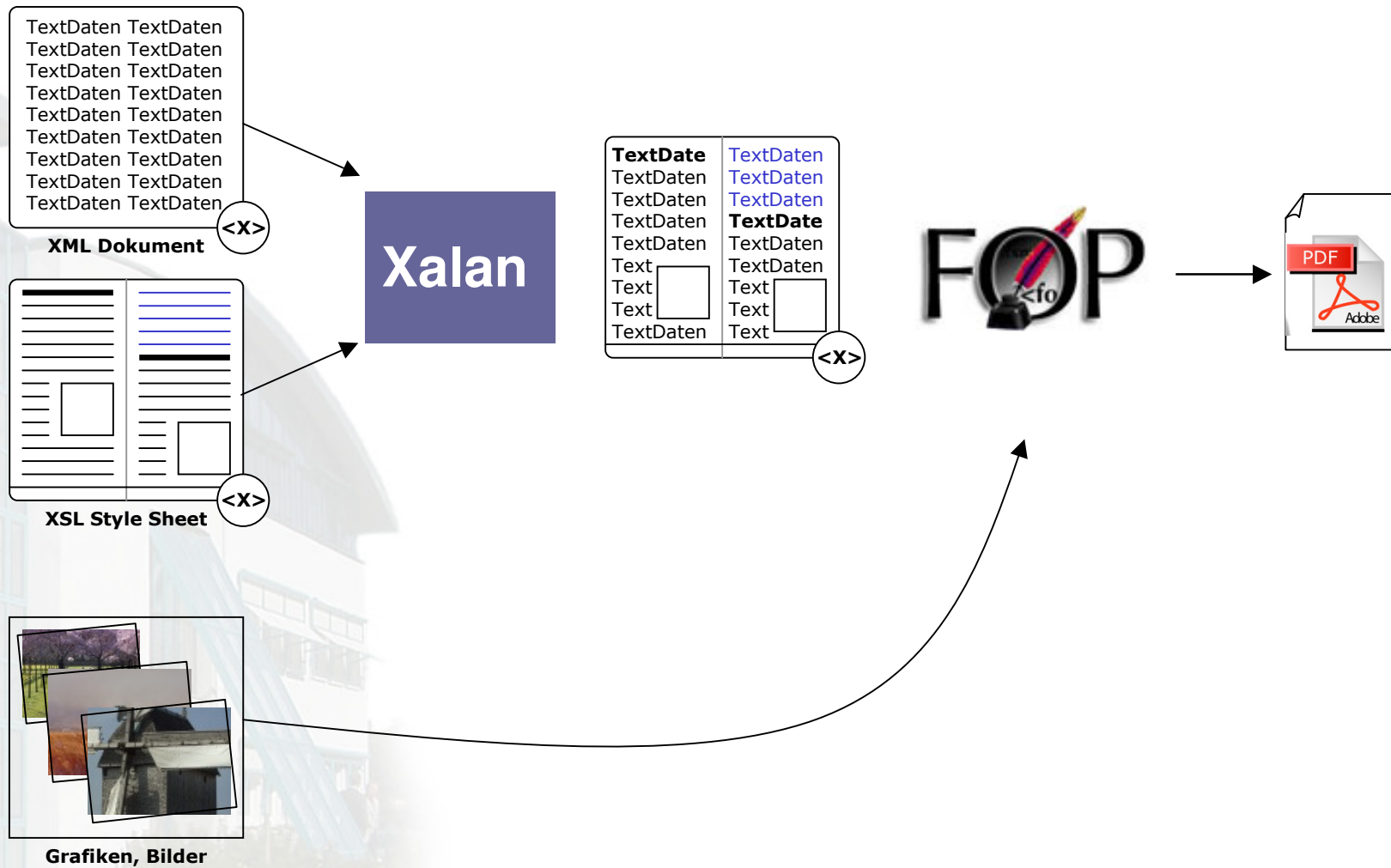
Formatierer FOP



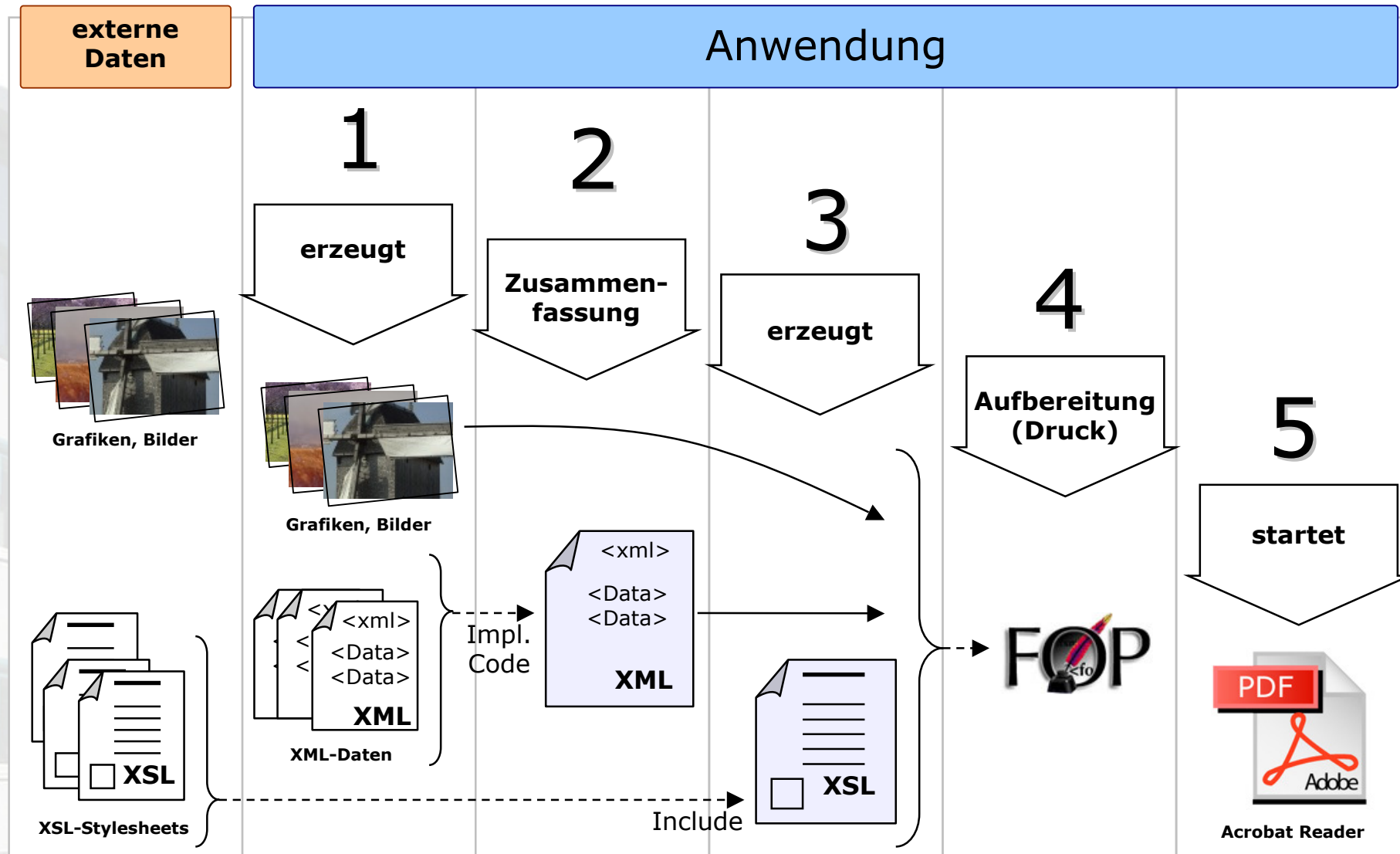
Formating **O**bject **P**rocessor

- **Open Source unterliegt der Apache License, Version 2.0**
- **basiert vollständig auf JAVA**
- **aktuell ist Version 0.93 (Version 1.0 in 2007 geplant)**
- **unterstützt weite Teile der XSL-FO 1.0 Spezifikation und einige Features von 1.1**
s. <http://xmlgraphics.apache.org/fop/compliance.html>
- **benutzt für XSL Transformation Xalan (Apache)**
- **unterstützt primär das PDF-Format**
weitere Formate PCL, PS, SVG, XML, ...

Druckprozess (1/3) Übersicht



Druckprozess (2/3) Ablauf



Druckprozess (3/3) Integration



FOP 0.93

- **Verwendung als externe Applikation**

- über die Batch-Datei fop.bat oder Shell Script
- Umgebungsvariable JAVA_HOME
- Java-Version (1.3 oder neuer)

→ **sinnvoll zum Test der Stylesheets**

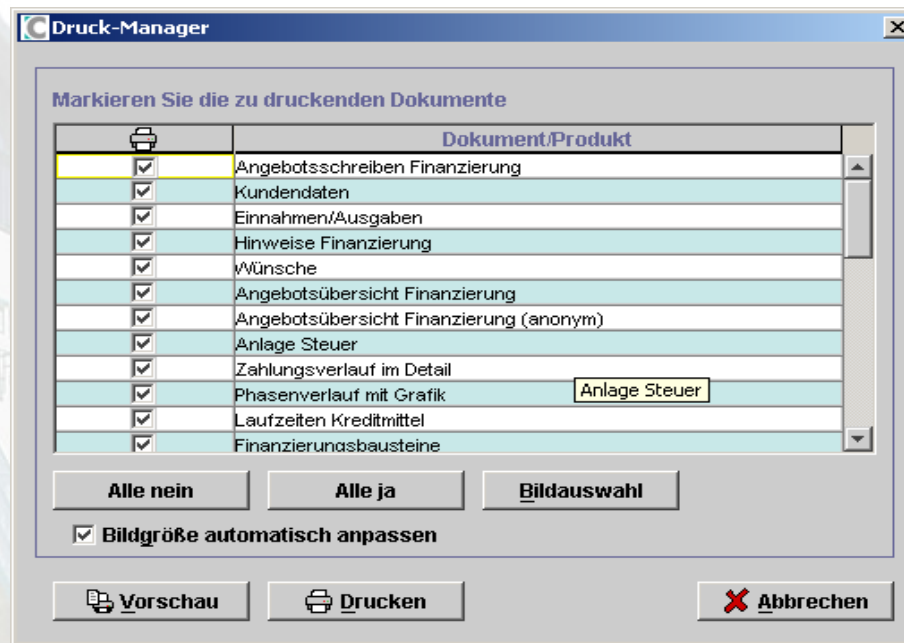
- **Einbindung in die eigene JAVA-Anwendung**

- einfaches kurzes Code-Fragment
- einige Apache JARs müssen integriert werden

Eigene Erweiterungen



- **Stylesheets in Anwendung konfigurierbar (XML)**
 - Keine Änderung der Implementierung
 - beim Austausch von Vorlagen
 - beim Hinzufügen neuer Vorlagen
- **Interaktive Auswahl der konfigurierten Stylesheets**
 - individuelles Dokument generierbar



Bewertung



- **plattformunabhängig**
Drucklösung komplett in JAVA realisiert
- **unabhängig von weiteren Installationen**
nur weitere JAR Archive
- **qualitativ hochwertige Ausdrücke**
reichhaltige Mittel zur Satzgestaltung
wird auch in der Druckindustrie eingesetzt
- **einfache Wartung und Pflege**
Vorlagen können ohne Implementierungsaufwand ausgetauscht
oder hinzugefügt werden
- **kundenspezifische Ausdrücke**
unterschiedliche Stylesheets, gleiche Daten
- **kosteneffizient**
keine weiteren Lizenzkosten mit FOP
- **Vorlagen einfach gestaltbar**
Einarbeitung erforderlich



Erfahrungen, Ausblick



- **Druckprobleme haben sich reduziert**
- **Erstellung der Vorlagen benötigt Zeit und Einarbeitung**
- **Fehlermeldungen von FOP wenig aussagekräftig**
- **Vorlagen können strukturiert werden (Modularisierung)**
→ Wiederverwendung
- **Werkzeuge für die XSL-FO Erstellung verwenden**
Style Vision (Altova), DocDesign (Assentis)
- **Tools zur Umsetzung eines Textsystem-Formats in XSL-FO können die Entwicklung der Stylesheets beschleunigen**

Links und Infos



<http://www.w3.org/Style/XSL/>

sehr umfangreich mit vielen weiteren Links

<http://xmlgraphics.apache.org/fop/>

Informationen zu Apache FOP

<http://www.data2type.de/xml/XML.html>

gute Einführung in deutscher Sprache



Fragen & Diskussion

