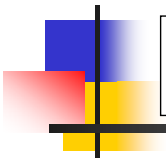
XSP – eXtensible Server Pages

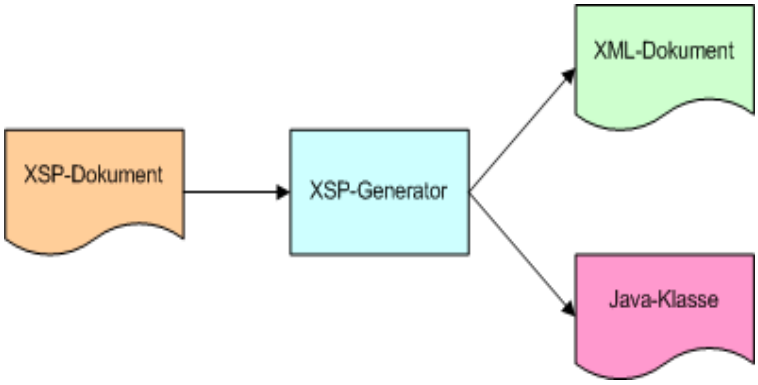
Thomas KALSBERGER



Einführung

- Extensible Serverpages (XSP‘s)
  - wie normale Serverpage
  - ist zusätzlich gültiges XML Dokument
  - liefert als Ergebnis XML Dokument, welches weiter transformiert werden kann (z.B. HTML, PDF, PHP,...)
  - basiert auf GP markup2code transformation engine
  - XSP Parser wandelt Logik in Javacode um
- Drei Kernbereiche
  - Dynamic Markup Language
  - Programming Language
  - Program Generator

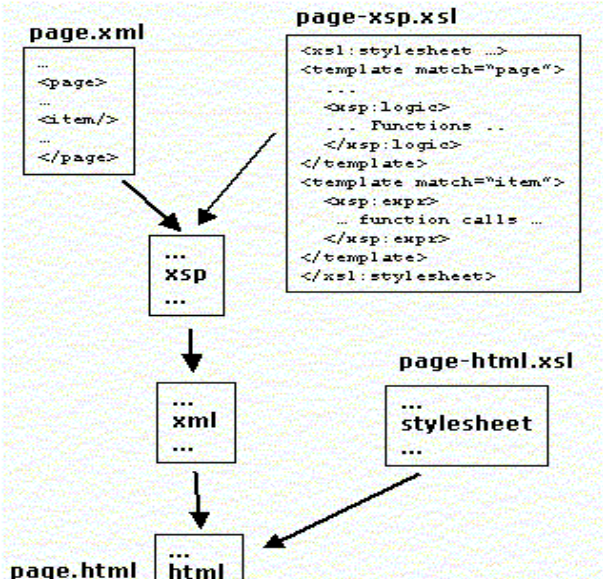
1. Eingebettete Logik



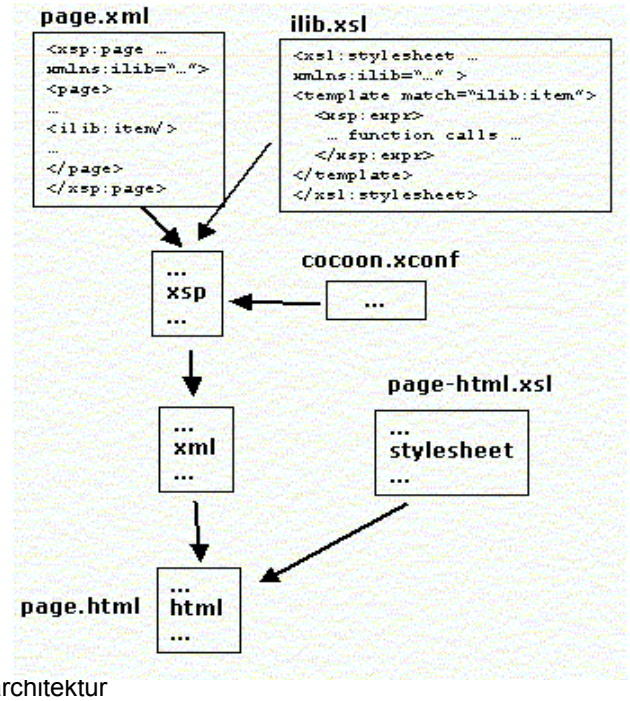
XSP Generator trennt Logik und Content des XSP Dokuments

Erstellungsmöglichkeiten 2

2. Eingebundenes Logicsheet



3. Logicsheet als Taglibrary

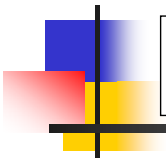


Erstellungsmöglichkeiten 4

- Eingebettete Logik
  - Code ist in XML Page enthalten
  - Keine Trennung von Content und Logic
  - Gut für kleine Projekte, schlecht füt große
- Eingebettetes Logicsheet
  - Code separat in XSL File
  - Effektive Trennung von Content und Logic
  - Bevorzugte Erstellungsweise
- Logicsheet als Taglibrary
  - Wiederverwendbare Library die in Cocoon registriert ist (cocoon.xconf)
  - Effektive Trennung von Content, Logic und

- Zentrale Steuerungseinheit, enthält:
  - matchers
  - generators
  - transformers
  - readers
  - serializers
  - selectors
  - processing pipelines with match patterns
  - ...

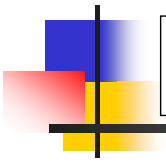
Wird in Programm transformiert und dann in ein ausführbares Programm kompiliert



Logicsheets

- XSL oder XSLT Dateien
  - mit speziellem Namensbereich
  - Primärer Mechanismus um Programmlogik in XSP's zu integerieren
  - Output ist XSP Datei
  - Müssen in cocoon.xconf registriert werden
  - Verwendet von Generator
  - Vielzahl vordefinierte logicsheets
    - request.xsl
    - response.xsl
    - session.xsl
    - cookie.xsl

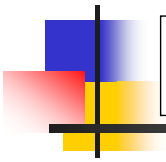
- **xsp:attribute**
  Erstellt ein Attribut. Um ein Attribut zu erstellen, muss dieses Element "xsp:param" enthalten, das wiederum den Namen "name" trägt. Der Wert von "xsp:param" wird zum Namen des Attributes. Die, mit "xsp:expr" definierte Ausgabe wird zum Wert des Attributes.
- **xsp:comment**
  Erlaubt es, innerhalb der XSP-Datei einen Kommentar zu definieren.
- **xsp:content**
  Fügt der Ausgabe Zeichen hinzu. Wird dieses Element innerhalb anderer Elemente definiert, so wird der Inhalt von "xsp:content" als String zurückgeliefert.



XSP Elemente 2

- **xsp:element**
  Mit diesem XSP-Tag können beliebige XML-Elemente dynamisch erstellt werden. Im Rumpf von "xsp:element" muss mit "xsp:param" der Name des Elements übergeben werden (name="name"). Mit "xsp:attribute" können dem Element beliebige viele Attribute hinzugefügt werden.
- **xsp:exit-page**
  Alles, was im Rumpf dieses Elementes definiert wird, wird am Ende der Bearbeitung ausgeführt.
- **xsp:expr**
  Alles, was im Rumpf dieses Elementes angegeben wurde, wird in die Ausgabe geschrieben. Bei ADT‘s wird automatisch die toString() Methode aufgerufen.

- xsp:include
  Dieses Element wird in Verbindung mit "xsp:stucture" verwendet, um Importanweisungen auszuführen. Falls Java als XSP-Sprache verwendet wird, so werden hiermit alle Importanweisungen definiert.
- xsp:init-page
  Dieses Element ist das Gegenstück zu "xsp:exit-page". Alle im Rumpf gemachten Anweisungen werden ausgeführt, bevor die eigentliche XSP-Seite geparst wird.
- xsp:logic
  Erlaubt es, im Rumpf jede beliebige Anweisung zu deklarieren, die auch im Rumpf einer Klasse der gewählten XSP-Sprache verwendet werden dürfte. (Für Vergleichsoperatoren < und > die entsprechenden < und > verwenden oder Rumpf in ein <![CDATA[]> Element einschliessen).



XSP Elemente 4

- xsp:page
  Der Rumpf dieses Elementes beinhaltet alle XSP-Elemente.
- xsp:param
  Verwendet um innerhalb anderer Elemente Parameter definieren zu können. Der Rumpf des Elementes stellt den Wert des Parameters dar.
- xsp:pi
  Processing Instruction:Dabei muss diesem Element im Rumpf "xsp:param" übergeben werden, welcher als Name "target" enthält. Im Rumpf von "xsp:param" wiederrum muss das Ziel der PI enthalten.
- xsp:structure
  Dieses Element leitet immer eine beliebige Menge von "xsp:inlcude" ein.

- Beispiele