

En introduktion til XSLT

Bent Guldbjerg Christensen

Email: bentor@daimi.au.dk

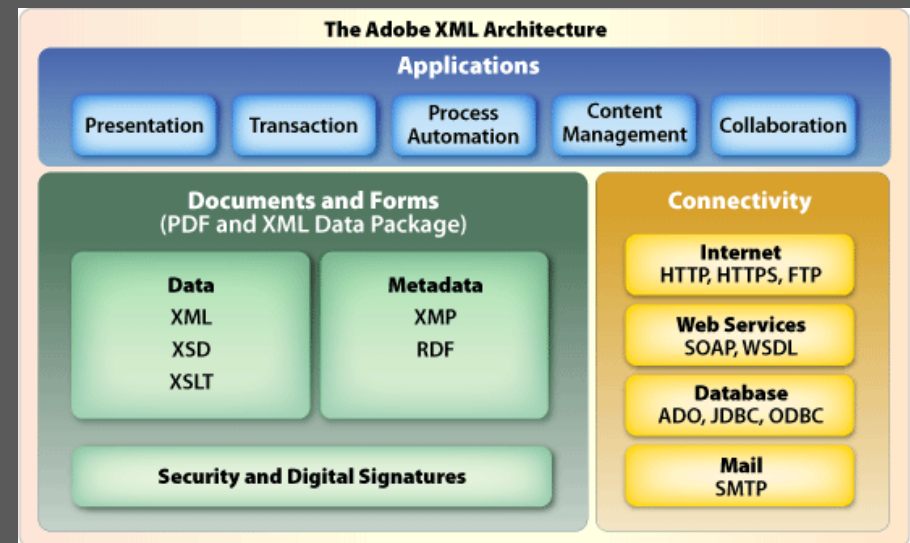
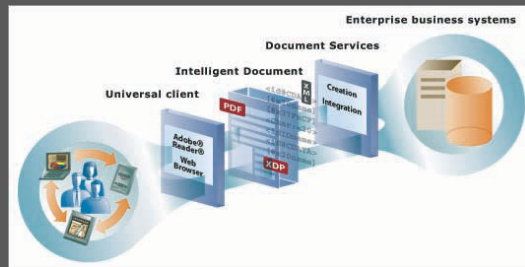
Datalogisk Institut, Aarhus Universitet

Plan

- Motivation for XSLT
 - Hvad kan XSLT bruges til?
 - XML og DTD basalt set
- XSLT
 - Hvor udføres XSLT?
 - Et "Hello World!" eksempel
 - XPath: Navigation i XML
 - Programmering med XSLT
- XSLT i praksis
 - Eksempler på brug af XSLT: Togbasen, DRs Nyheder
- Sammenfatning
 - Hvor finder jeg mere?

Hvad kan XSLT bruges til?

- XSLT står for: Extensible Stylesheet Language for Transformations – dvs. et sprog til at beskrive transformationer.
- XSLT kan bruges til at transformere XML dokumenter til noget andet... F.eks. HTML, PDF, SVG...
- Da XML ofte bruges som udvekslingsformat for "rådata", indgår XSLT transformationerne som en essentiel del af den videre forarbejdelse, og er især nyttig i tilfælde, hvor informationen ændrer sig ofte.
- Her er et eksempel fra Adobes informationsarkitektur, hvor XML og XSLT indgår som vigtige elementer:



XML og DTD basalt set

- Extensible Markup Language (XML) blev specificeret i 1998 for tilbyde en simpel mekanisme til at strukturere, gemme og levere information. XML er baseret på principper fra to tidligere markup sprog, nemlig SGML og HTML:
 - Standard Generalised Markup Language (SGML) blev ratificeret med ISO (8879) i 1986, og er siden forblevet stortset uændret. SGML specifikationen beskriver et avanceret og komplekst markup sprog for definition af data formater. Et metasprog!
 - HTML blev udviklet i 1990 som hypertext dokument formatet til World Wide Web. Dette nye markup sprog havde et meget simpelt design sammenlignet med SGML. I kontrast til SGML, som er et metasprog, er HTML designet med et fast markup tags.
- Så hvad er XML data? Det kan betragtes som en almindelig tekstfil med tags. De forskellige tags definerer XML elementer og elementerne udgør XML dokument strukturen.

XML og DTD basalt set

- Definitionen på et XML dokument ifølge specifikationen:
- En tekstfil er et XML dokument, hvis og kun hvis det er well-formed. For at være well-formed skal følgende tre regler overholdes:
 - Dokumenterne skal begynde med en XML deklaration (dokument prolog). Som denne: `<?xml version="1.0" ?>`
 - Der er et rod element, som indeholder alle andre elementer.
 - Alle elementer skal være rigtig indlejret, dvs. elementer må ikke overlappe. F.eks.: `<tag1> <tag2>Indhold</tag2> </tag1>`

XML og DTD basalt set

- Eksempel på et well-formed XML dokument:

```
<?xml version="1.0" encoding="utf-8?">
  <class>
    <student studentno="4">
      <name>Lise</name>
      <email>lise@daimi.au.dk</email>
      <affiliation>Daimi</affiliation>
    </student>

    <student studentno="5">
      <name>Peter</name>
      <email>peter@daimi.au.dk</email>
      <affiliation>Daimi</affiliation>
    </student>
  </class>
```

XML og DTD basalt set

- XML standarden definerer, hvordan man konstruerer en dokument type deklaration (DTD), som beskriver grammatikken for en klasse af XML dokumenter. Mere præcist: En DTD dikterer den logiske struktur af XML dokumenter.
- Dette udtrykkes med en række deklarations elementer, som definerer “data” elementer og deres indbyrdes strukturelle relation. DTD elementerne deklarerer nye data elementer og hvilken type indhold de nye elementer må indeholde.
- Et XML dokument er valid, hvis og kun hvis, XML dokumentet har tilknyttet en DTD og dokumentet overholder betingelserne beskrevet i denne DTD.

XML og DTD basalt set

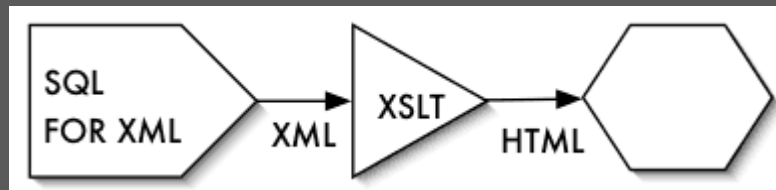
- Et eksempel på en DTD, som det tidligere XML dokument overholder:

```
<?xml version="1.0" encoding="utf-8"?>
<!ELEMENT class (student*)>
<!ELEMENT student (name,email,affiliation)>
<!ATTLIST student studentno CDATA #REQUIRED>
<!ELEMENT name (#PCDATA)>
<!ELEMENT email (#PCDATA)>
<!ELEMENT affiliation (#PCDATA)>
```

- XML Schema er en anden meget udbredt standard til at beskrive den logiske struktur af XML dokumenter, men vil ikke blive gennemgået her.

XSLT

- Hvor udføres XSLT?
 - XSLT bruges meget ofte i en server/klient-arkitektur, hvor en server tilbyder dynamisk information til flere forskellige klienter. Transformationerne kan blive udført på både serveren og på klienten.
 - XSLT bruges ofte i forbindelse med WWW, hvor en database indeholder dynamisk information man ønsker at vise på en webside:
 - Databasen eksporterer information i et XML dokument,
 - XML dokumentet transformeres herefter på webserveren med et XSLT stylesheet til HTML og sendes til webbrowseren,
 - HTML filen vises i webbrowseren.



XSLT

- Et "Hello World!" eksempel XSLT stylesheet:

```
<?xml version="1.0" encoding="utf-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:output method="text" indent="no"/>

  <xsl:template match="/">
    <xsl:for-each select="/class/student">
      <xsl:value-of select="name/."/>
      <xsl:text> (</xsl:text><xsl:value-of select="email/."/><xsl:text> )</xsl:text>
      <xsl:text> is affiliated with </xsl:text>
      <xsl:value-of select="affiliation"/>
      <xsl:value-of select="$newline"/>
    </xsl:for-each>
  </xsl:template>

  <xsl:variable name="newline">
    <xsl:text>
  </xsl:text>
  </xsl:variable>
</xsl:stylesheet>
```

XSLT

- Resultatet af at udføre transformation vist på foregående slide på XML dokumentet med information om de to studerende Lise og Peter bliver til:

Lise (lise@daimi.au.dk) is affiliated with Daimi

Peter (peter@daimi.au.dk) is affiliated with Daimi

XSLT

- Hvilke mekanismer har man i XSLT til at beskrive transformationer?
 - XPath til navigering og udvælgelse af XML elementer
 - Traditionelle programmeringsmekanismer: variable, tests (if, choose), løkker (for-each) og procedurer (templates).

XSLT

- XML Path (XPath) definerer en syntaks til at adressere dele af et XML dokument. XPath udtryk bliver altid evalueret i forbindelse med et XML dokument. Et eksempel:

`/class/student[1]`

- Simple tests (prædikater) på en samling af XML elementer kan bruges til at udvælge bestemte elementer. Der er defineret en række funktioner i XPath specifikationen til at lave test på: samlinger af elementer, tekst og tal. Prædikater er altid angivet i [].
- Eksemplet herunder illustrerer tekst funktionen `contains()`, som evaluerer til true, hvis det første tekst argument indeholder det andet tekst argument, ellers evaluerer den til false. Her er det første tekst argument indholdet af elementet `name`:

`/class/student[1]/name/[contains(.,'t')]`

XSLT

- Programmeringsmekanismer:

- If:

```
<xsl:if test="/class/student[1]/name/[contains(.,'t')]">  
  <xsl:text>The test evaluated to true</xsl:text>  
</xsl:if>
```

- Choose (udvidet if):

```
<xsl:choose>  
  <xsl:when test="/class/student[1]/name/[contains(.,'t')]">  
    <xsl:text>The 1st test evaluated to true</xsl:text>  
  </xsl:when>  
  <xsl:when test="/class/student[1]/name/[contains(.,'l')]">  
    <xsl:text>The 2nd test evaluated to true</xsl:text>  
  </xsl:when>  
  <xsl:otherwise>  
    <xsl:text>None if the test evaluated to true</xsl:text>  
  </xsl:otherwise>  
</xsl:choose>
```

XSLT

- Programmeringsmekanismer fortsat:

- For-each:

```
<xsl:for-each select="/class/student">  
  <xsl:value-of select="position()" />  
  <xsl:text>. </xsl:text>  
  <xsl:value-of select="name/." />  
  <xsl:text>  
  </xsl:text>  
</xsl:for-each>
```

- For-each løkken (i passende indpakning) vil resultere i følgende output:

1. Lise
2. Peter

XSLT

- Et XSLT stylesheet er struktureret i en samling templates:

- Templates:

```
<xsl:template name="foo">  
  <xsl:text>Template foo has been called</xsl:text>  
  <xsl:call-template name="bar"/>  
</xsl:template>
```

- En anden måde at aktivere (trigger) templates:

```
<xsl:apply-templates/>  
  
<xsl:template match="name">  
  <xsl:text>The name is: </xsl:text>  
  <xsl:value-of select="."/>  
</xsl:template>
```


XSLT

- Kontrolflow i et XSLT stylesheet:

```
<xsl:template match="/">
  <xsl:text>Here's the beginning.</xsl:text>
  <xsl:call-template name="printStudents"/>
  <xsl:text>Here's the end.</xsl:text>
</xsl:template>
```

```
<xsl:template name="printStudents">
  <xsl:for-each select="/class/student">
    <xsl:value-of select="position()"/>
    <xsl:text>. </xsl:text>
    <xsl:value-of select="name/."/>
    <xsl:text>
  </xsl:for-each>
</xsl:template>
```

XSLT i praksis

- Togbasen: Et XML dokument over modeltog med tilhørende XSLT stylesheet. Togbasen findes i to versioner – en hvor transformationen sker på serveren og en hvor transformationen sker i webbrowseren:

Webserver: <http://www.daimi.au.dk/~fah/berith/togdb/index.html>
Rådata: <http://www.daimi.au.dk/~fah/berith/togdb/togdb.xml>
Stylesheet: <http://www.daimi.au.dk/~fah/berith/togdb/xml/togdb.xsl>
DTD: <http://www.daimi.au.dk/~fah/berith/togdb/xml/togdb.dtd>

```
<?xml version="1.0" encoding="iso-8859-1"?>  
<!DOCTYPE togbase PUBLIC "-//Togdb //DTD Web Application 2.3//EN" "http://www.daimi.au.dk/~fah/berith/togdb/xml/togdb.dtd">
```

```
<?xml-stylesheet href="xml/togdb.xsl" type="text/xsl"?>
```

```
<togbase>  
  <tog id="1" artikelNr="2067">  
    <beskrivelse>Køf II 269</beskrivelse>  
    <type>DSB traktor 269</type>  
    <farve>Grøn</farve>  
    <aksler>2</aksler>  
    <selskab>DSB</selskab>  
    <art>Rangertraktor</art>  
    <underart>Diesel</underart>  
    <fabrikat>Arnold</fabrikat>  
    ...  
  </tog>  
  <tog>  
    ...  
  </tog>  
</togbase>
```



XSLT i praksis

- DR Nyheder benytter RSS-feeds, som er et XML baseret format til nyheder. Et RSS-feed læses/vises ofte med et dedikeret program, en RSS-reader.
- DR har dog også valgt at man skal have mulighed for at læse RSS-feed'et i en webbrowser, så på samme måde som Togbasen er der tilknyttet et XSLT stylesheet som kan transformere RSS-feed'et til et HTML dokument (som webbrowser kan vise):

Rådata: <http://www.dr.dk/Nyheder/Service/feeds/Alleenyheder.xml>

Stylesheet: <http://www.dr.dk/nyheder/htm/service/rss/rssfeed.xsl>



Sammenfatning

- XSLT kan bruges til at transformere XML dokumenter til noget andet... F.eks. HTML, PDF, SVG...
- Da XML ofte bruges som udvekslingsformat for "rådata", indgår XSLT som en essentiel del af den videre forarbejdelse, og er især nyttig i tilfælde, hvor informationen ændrer sig ofte.
- I en traditionel server/klient-arkitektur (såsom WWW) kan XSLT transformationer benyttes både på server- og klient-siden.
- Hvor finder jeg mere?
 - Specifikationen fra W3C kan findes på: <http://www.w3.org/TR/xslt>
 - En udmærket tutorial kan findes på: <http://www.w3schools.com/xsl/>