Webserverprogrammering

WSP - dag 10

XML (Extended Meta Language)
XML Namespaces
Dokumenttype/skemadefinitioner - DTD og XSD
Behandling af XML (DOM, SAX StAX, XSLT)
(Syndikering og RSS med XML)
JAXP - XML ⇆ Java-objekter

Projekthjælp

XML

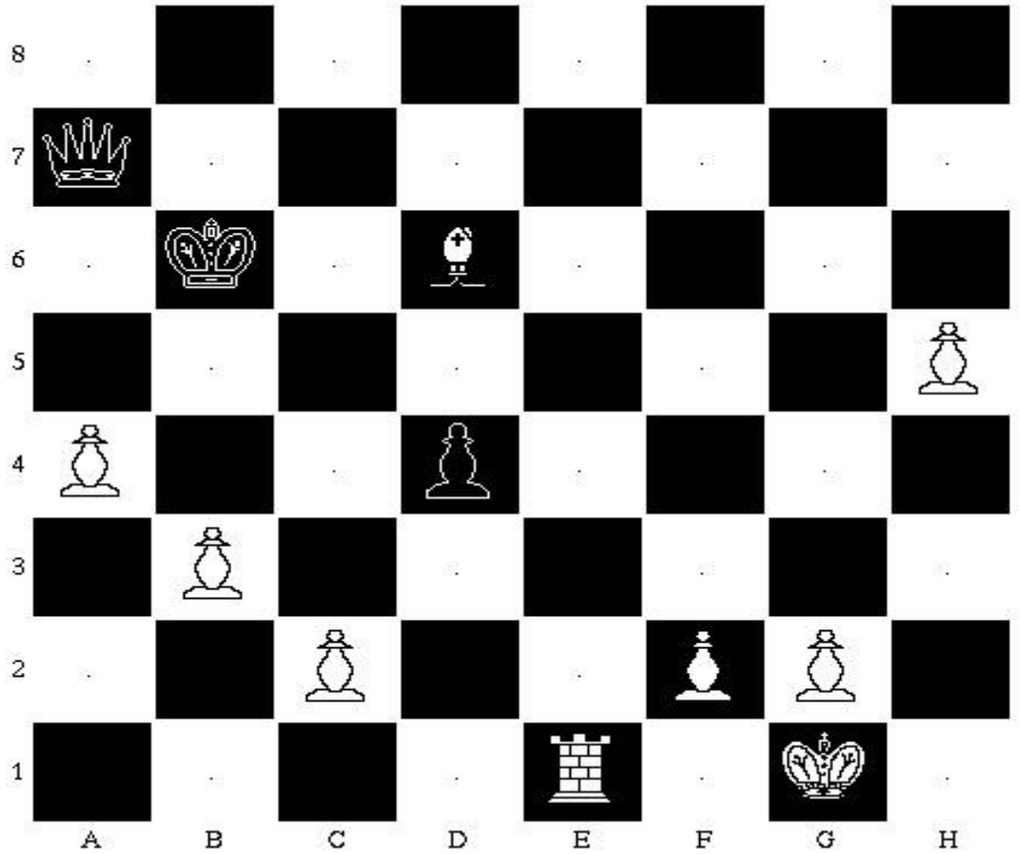
Det mindst ringe filformat !

XML (*Extended Meta Language*)

- XML repræsentation af skakspil
- SAX, DOM og XSLT sammenlignet
- Anvendelse af SAX
- Anvendelse af DOM
- Anvendelse af XSLT
- Eksempel: Nyhedsfødning (RSS)

XML repræsentation af skakspil

Skaksituation



kilde: http://java.sun.com/developer/technicalArticles/xml/JavaTechandXML/

XML beskrivelse af skaksituation

```
<CHESSBOARD>
 <WHITEPIECES>
  <KING><POSITION COLUMN="G" ROW="1"/></KING>
  <BISHOP><POSITION COLUMN="D" ROW="6"/></BISHOP>
  <ROOK><POSITION COLUMN="E" ROW="1"/></ROOK>
  <PAWN><POSITION COLUMN="A" ROW="4"/></PAWN>
  <PAWN><POSITION COLUMN="B" ROW="3"/></PAWN>
  <PAWN><POSITION COLUMN="C" ROW="2"/></PAWN>
  <PAWN><POSITION COLUMN="F" ROW="2"/></PAWN>
  <PAWN><POSITION COLUMN="G" ROW="2"/></PAWN>
  <PAWN><POSITION COLUMN="H" ROW="5"/></PAWN>
 </WHITEPIECES>
 <BLACKPIECES>
  <KING><POSITION COLUMN="B" ROW="6"/></KING>
  <QUEEN><POSITION COLUMN="A" ROW="7"/></QUEEN>
  <PAWN><POSITION COLUMN="A" ROW="5"/></PAWN>
  <PAWN><POSITION COLUMN="D" ROW="4"/></PAWN>
 </BLACKPIECES>
</CHESSBOARD>
```

kilde: http://java.sun.com/developer/technicalArticles/xml/JavaTechandXML/

DTD for skaksituation (Chessboard.dtd)

```
<!ELEMENT CHESSBOARD (WHITEPIECES, BLACKPIECES)>
<!ENTITY % pieces
  "KING, QUEEN?, BISHOP?, BISHOP?,
  ROOK?, ROOK?, KNIGHT?, KNIGHT?,
  PAWN?, PAWN?, PAWN?, PAWN?,
  PAWN?, PAWN?, PAWN?, PAWN?"
>
<!ELEMENT WHITEPIECES (%pieces;)>
<!ELEMENT BLACKPIECES (%pieces;)>
<!ELEMENT POSITION EMPTY>
<!ATTLIST POSITION
  COLUMN (A|B|C|D|E|F|G|H) #REQUIRED
  ROW (1|2|3|4|5|6|7|8) #REQUIRED
>
<!ELEMENT KING (POSITION)>
<!ELEMENT QUEEN (POSITION)>
<!ELEMENT BISHOP (POSITION)>
<!ELEMENT ROOK (POSITION)>
<!ELEMENT KNIGHT (POSITION)>
<!ELEMENT PAWN (POSITION)>
```

DTD for flere skaksituationer (Chessboards.dtd)

```
<!ELEMENT CHESSBOARDS (CHESSBOARD*)>
<!ENTITY % chessboard SYSTEM "Chessboard.dtd">
%chessboard;
```

kilde: http://java.sun.com/developer/technicalArticles/xml/JavaTechandXML/

XML beskrivelse af skaksituationer (bruger Chessboards.dtd)

```
<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE CHESSBOARDS SYSTEM "dtd/Chessboards.dtd">

<CHESSBOARDS>
  <CHESSBOARD>
    <WHITEPIECES>
      <KING><POSITION COLUMN="G" ROW="1" /></KING>
      <BISHOP><POSITION COLUMN="D" ROW="6" /></BISHOP>
      <ROOK><POSITION COLUMN="E" ROW="1" /></ROOK>
      <PAWN><POSITION COLUMN="A" ROW="4" /></PAWN>
      <PAWN><POSITION COLUMN="B" ROW="3" /></PAWN>
      <PAWN><POSITION COLUMN="C" ROW="2" /></PAWN>
      <PAWN><POSITION COLUMN="F" ROW="2" /></PAWN>
      <PAWN><POSITION COLUMN="G" ROW="2" /></PAWN>
      <PAWN><POSITION COLUMN="H" ROW="5" /></PAWN>
    </WHITEPIECES>
    <BLACKPIECES>
      <KING><POSITION COLUMN="B" ROW="6" /></KING>
      <QUEEN><POSITION COLUMN="A" ROW="7" /></QUEEN>
      <PAWN><POSITION COLUMN="A" ROW="5" /></PAWN>
      <PAWN><POSITION COLUMN="D" ROW="4" /></PAWN>
    </BLACKPIECES>
  </CHESSBOARD>
  <CHESSBOARD>...</CHESSBOARD>
</CHESSBOARDS>
```

XML skema (xsd/Chessboard.xsd)

```
<?xml version="1.0" encoding="UTF-8"?>

<xsd:schema
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://mde.sun.com/Chessboard"
  xmlns="http://mde.sun.com/Chessboard"
  elementFormDefault="qualified">

 <xsd:element name="CHESSBOARD">
  <xsd:complexType>
   <xsd:sequence>
    <xsd:element name="WHITEPIECES" type="pieces" />
    <xsd:element name="BLACKPIECES" type="pieces" />
   </xsd:sequence>
  </xsd:complexType>
 </xsd:element>

 <xsd:complexType name="pieces">
  <xsd:sequence>
   <xsd:element name="KING" type="piece"
     minOccurs='1' maxOccurs='1'/>
   <xsd:element name="QUEEN" type="piece"
     minOccurs='0' maxOccurs='1'/>
   <xsd:element name="BISHOP" type="piece"
     minOccurs='0' maxOccurs='2'/>
   <xsd:element name="ROOK" type="piece"
     minOccurs='0' maxOccurs='2'/>
   <xsd:element name="KNIGHT" type="piece"
     minOccurs='0' maxOccurs='2'/>
   <xsd:element name="PAWN" type="piece"
     minOccurs='0' maxOccurs='8'/>
  </xsd:sequence>
 </xsd:complexType>

 <xsd:complexType name="piece">
  <xsd:sequence>
   <xsd:element name="POSITION"
     minOccurs='1' maxOccurs='1'>
    <xsd:complexType>
     <xsd:attribute name="COLUMN" use='required'>
      <xsd:simpleType>
       <xsd:restriction base="xsd:string">
        <xsd:pattern value="[A-H]"/>
       </xsd:restriction>
      </xsd:simpleType>
     </xsd:attribute>
     <xsd:attribute name="ROW" use='required'>
      <xsd:simpleType>
       <xsd:restriction base="xsd:positiveInteger">
        <xsd:minInclusive value="1"/>
        <xsd:maxInclusive value="8"/>
       </xsd:restriction>
      </xsd:simpleType>
     </xsd:attribute>
    </xsd:complexType>
   </xsd:element>
  </xsd:sequence>
 </xsd:complexType>
</xsd:schema>
```

XML skema for flere (xsd/Chessboards.xsd)

```
<?xml version="1.0" encoding="UTF-8"?>

<xsd:schema
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://mde.sun.com/Chessboards"
  xmlns="http://mde.sun.com/Chessboards"
  xmlns:cb="http://mde.sun.com/Chessboard">

 <xsd:import namespace="http://mde.sun.com/Chessboard"
   schemaLocation='Chessboard.xsd' />

 <xsd:element name="CHESSBOARDS">
  <xsd:complexType>
   <xsd:sequence>
    <xsd:element ref="cb:CHESSBOARD"
      maxOccurs='unbounded'/>
   </xsd:sequence>
  </xsd:complexType>
 </xsd:element>
</xsd:schema>
```

XML beskrivelse af skaksituationer (bruger xsd/Chessboards.xsd)

```
<?xml version="1.0" encoding="UTF-8"?>

<cbs:CHESSBOARDS
  xmlns:cbs="http://mde.sun.com/Chessboards"
  xmlns="http://mde.sun.com/Chessboard"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation='http://mde.sun.com/Chessboard  xsd/Chessboard.xsd
                      http://mde.sun.com/Chessboards  xsd/Chessboards.xsd'>
<CHESSBOARD>
 <WHITEPIECES>
  <KING><POSITION COLUMN="G" ROW="1" /></KING>
  <BISHOP><POSITION COLUMN="D" ROW="6" /></BISHOP>
  <ROOK><POSITION COLUMN="E" ROW="1" /></ROOK>
  <PAWN><POSITION COLUMN="A" ROW="4" /></PAWN>
  <PAWN><POSITION COLUMN="B" ROW="3" /></PAWN>
  <PAWN><POSITION COLUMN="C" ROW="2" /></PAWN>
  <PAWN><POSITION COLUMN="F" ROW="2" /></PAWN>
  <PAWN><POSITION COLUMN="G" ROW="2" /></PAWN>
  <PAWN><POSITION COLUMN="H" ROW="5" /></PAWN>
 </WHITEPIECES>
 <BLACKPIECES>
  <KING><POSITION COLUMN="B" ROW="6" /></KING>
  <QUEEN><POSITION COLUMN="A" ROW="7" /></QUEEN>
  <PAWN><POSITION COLUMN="A" ROW="5" /></PAWN>
  <PAWN><POSITION COLUMN="D" ROW="4" /></PAWN>
 </BLACKPIECES>
</CHESSBOARD>
<CHESSBOARD>...</CHESSBOARD>
</cbs:CHESSBOARDS>
```

SAX, DOM og XSLT sammenlignet

SAX og DOM

SAX	DOM
Event based model	Tree data structure
Serial access (flow of events)	Random access (in-memory data structure)
Low memory usage (only events are generated)	High memory usage (the document is loaded in memory)
To process parts of the document (catching relevant events)	To edit the document (processing the in-memory data structure)
To process the document only once (transient flow of events)	To process multiple times (document loaded in memory)

kilde: http://java.sun.com/developer/technicalArticles/xml/JavaTechandXML/

Processing Phase	SAX	DOM	XSLT
XML input processing			
Parsing and validating	Built in	Built in or based on SAX	Based on SAX or DOM
Recognizing/searching	Catching events with event handlers	Searching the tree with tree walkers	Xpath patterns
Extracting	Catching events	Getting attribute values, node content: API methods	Getting attribute values, node contents: Xpath statements
Mapping/binding	Creating business objects from the extracted information	Creating business objects from the extracted information	If ever, through DOM or SAX (pipelining)
XML output processing			
Constructing	No default support but can be done by generating a properly balanced sequence of method calls to event handlers	Implicitly part of the model: API factory methods	Implicitly part of the model: XSL statements
Serializing	No default support but can be done with a custom event handler	Implementation specific support, or through XSLT identity transformation	Implicitly part of the model: XSL output method statement

kilde: http://java.sun.com/developer/technicalArticles/xml/JavaTechandXML/

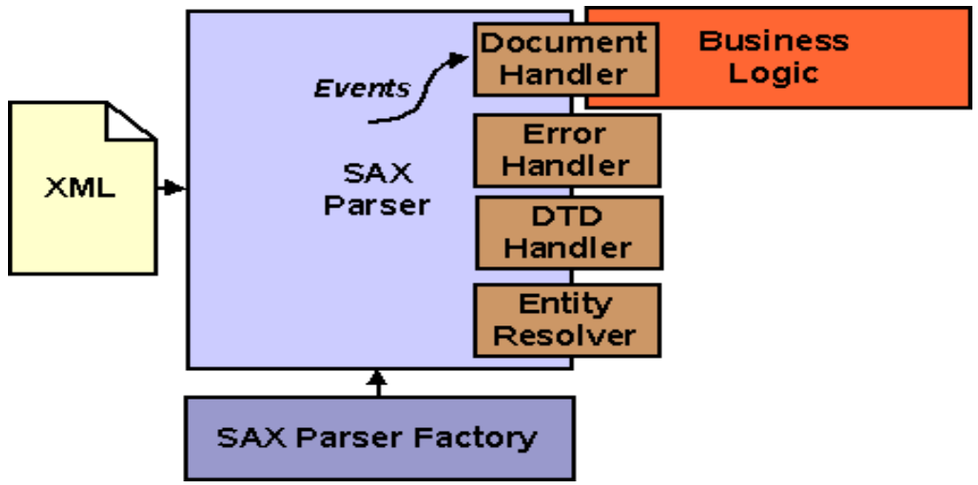
Anvendelse af SAX

SAX-proceduren

1. Opret en SAX ’parser’ fabrik
2. Konfigurér ’parser’ fabrik
3. Opret ny ’parser’ vha. fabrik
4. Sæt dokumenthåndtering, fejlhåndtering, DTD-håndtering og ’resolver’ for ’parser’
5. Gennemfør ’parsing’ af XML dokument(er)

SAX-baseret behandling

org.xml.sax.XMLReader – Interface for SAX ‘parser’; registrering af alle ‘handlers’ og ‘resolver’
org.xml.sax.ContentHandler – Interface for modtagelse af hændelser (dokument start/slut, element start/slut, data)
org.xml.sax.ErrorHandler – Interface for modtagelse af fejl (ikke kritisk, kritisk og advarsel)
org.xml.sax.DTDHandler – Interface for modtagelse af DTD-hændelser
org.xml.sax.EntityResolver – Interface for eksterne referencer
org.xml.sax.DefaultHandler – Default implementation af de 4 foregående interfaces



SAX-baseret behandling (ChessboardSAXPrinter.java, 1)

```
import org.xml.sax.*;
import org.xml.sax.helpers.*;
import javax.xml.parsers.*;

public class ChessboardSAXPrinter {
 private SAXParser parser;

 public ChessboardSAXPrinter(boolean validating)
   throws Exception {
  SAXParserFactory factory
    = SAXParserFactory.newInstance();
  factory.setValidating(validating);
  parser = factory.newSAXParser();
  ...
  return;
 }

 public void print(String fileName, PrintStream out)
   throws SAXException, IOException {
  ...
  parser.parse(fileName, ...);
  return;
 }
```

SAX-baseret behandling (ChessboardSAXPrinter.java, 2)

```
 public static void main(String[] args) {
  ...
  for (int k = 0; k < r; k++) {
    // r: number of runs
   ChessboardSAXPrinter saxPrinter
     = new ChessboardSAXPrinter(validating);
   long time = System.currentTimeMillis();
   for (int i = 0; i < n; i++) {
     // n: number of document processed per run
    saxPrinter.print(args[0], out);
   }
   // print out the average time (s)
   // to process a document during the current run
   System.err.print(
     (((double) (System.currentTimeMillis()
                 - time)) / 1000 / n) + "\t");
  }
  ...
 }
}
```

SAX-baseret behandling (ChessboardSAXPrinter.java, 3)

```
import org.xml.sax.*;
import org.xml.sax.helpers.*;
import javax.xml.parsers.*;

public class ChessboardSAXPrinter {
 private SAXParser parser;
 private PrintStream out;

 public class ChessboardHandler extends DefaultHandler {
  private boolean whitePiece = false;

  public void startElement(String name,
     AttributeList attrs) {
   if (name.equals("WHITEPIECES")) {
    whitePiece = true;
   } else if (name.equals("BLACKPIECES")) {
    whitePiece = false;
   } else if (name.equals("KING")
```

SAX-baseret behandling (ChessboardSAXPrinter.java, 4)

```
                    || name.equals("QUEEN")
                    || name.equals("BISHOP")
                    || name.equals("ROOK")
                    || name.equals("KNIGHT")
                    || name.equals("PAWN")) {
      out.print((whitePiece ? "White" : "Black")
                  + " "+ name.toLowerCase() + ": ");
    } else if (name.equals("POSITION")) {
      if (attrs != null) {
        out.print(attrs.getValue("COLUMN"));
        out.println(attrs.getValue("ROW"));
      }
    }
    return;
  }
  ...
}
...
}
```

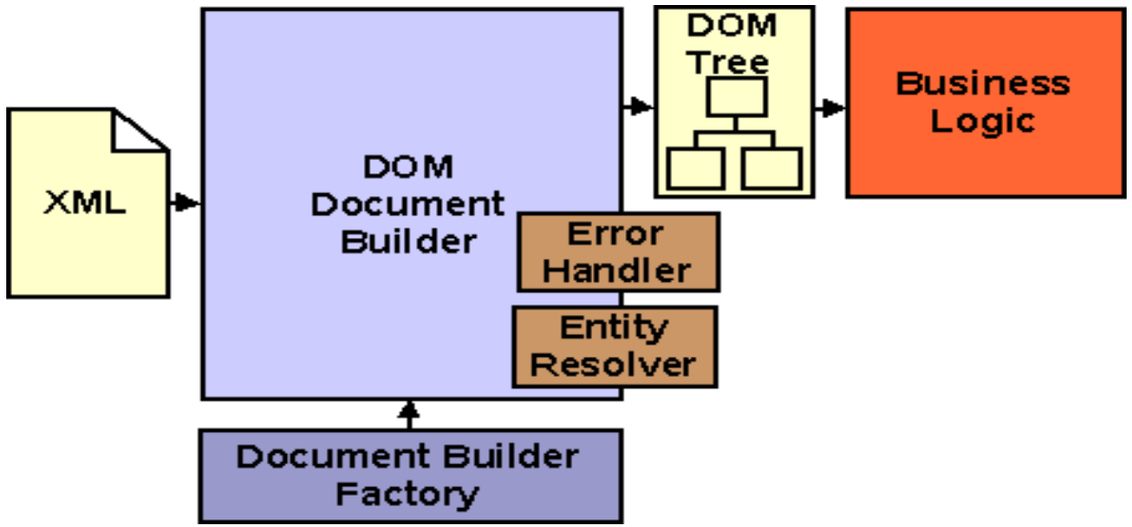
Anvendelse af DOM

DOM-proceduren

1. Opret en DOM dokumentbygnings fabrik
2. Konfigurér fabrikken
3. Opret ny dokumentbygger vha. fabrik
4. Sæt fejlhåndtering og ’resolver’ for underliggende ’parser’
5. Gennemfør ’parsing’ af XML dokument(er) til generering af DOM træ

DOM-baseret behandling

org.w3c.dom.Node – Interface for enkelt knude i træ; metoder for tilgang, flytning, indsættelse, fjernelse af barneknuder; metoder for tilgang til forældreknude og dokument
org.w3c.dom.Document – Interface for knude for hele dokumentet
org.w3c.dom.Element – Interface for knude for XML element
org.w3c.dom.Text – Interface for knude for data til et element



DOM-baseret behandling (ChessboardDOMPrinter.java, 1)

```
import org.w3c.dom.*;
import org.xml.sax.*;
import javax.xml.parsers.*;

public class ChessboardDOMPrinter {
 private DocumentBuilder builder;

 public void print(String fileName, PrintStream out)
   throws SAXException, IOException {
  Document document = builder.parse(fileName);
  NodeList nodes_i
    = document.getDocumentElement().getChildNodes();
  for (int i = 0; i < nodes_i.getLength(); i++) {
   Node node_i = nodes_i.item(i);
   if (node_i.getNodeType() == Node.ELEMENT_NODE
       && ((Element) node_i).getTagName()
          .equals("CHESSBOARD")) {
    Element chessboard = (Element) node_i;
    NodeList nodes_j = chessboard.getChildNodes();
```

DOM-baseret behandling (ChessboardDOMPrinter.java, 2)

```
for (int j = 0; j < nodes_j.getLength(); j++) {
 Node node_j = nodes_j.item(j);
 if (node_j.getNodeType() == Node.ELEMENT_NODE) {
  Element pieces = (Element) node_j;
  NodeList nodes_k = pieces.getChildNodes();
  for (int k = 0; k < nodes_k.getLength); k++) {
   Node node_k = nodes_k.item(k);
   if (node_k.getNodeType() == Node.ELEMENT_NODE) {
    Element piece = (Element) node_k;
    Element position
      = (Element) piece.getChildNodes().item(0);
    out.println((pieces.getTagName()
                    .equals("WHITEPIECES")
                 ? "White " : "Black ")
                + piece.getTagName().toLowerCase()
                + ": "
                + position.getAttribute("COLUMN")
                + position.getAttribute("ROW"));
   }
  }
 }
}
}
```

DOM-baseret behandling (naiv ChessboardDOMPrinter.java)

```
public void print(String fileName, PrintStream out)
  throws SAXException, IOException {
 Document document = builder.parse(fileName);
 NodeList positions
   = document.getElementsByTagName("POSITION");
 for (int i = 0; i < positions.getLength(); i++) {
  Element position = (Element) positions.item(i);
  Element piece = (Element) position.getParentNode();
  Element pieces = (Element) piece.getParentNode();
  out.println(
    (pieces.getTagName().equals("WHITEPIECES")
     ? "White " : "Black ")
    + piece.getTagName().toLowerCase() + ": "
    + position.getAttribute("COLUMN")
    + position.getAttribute("ROW"));
 }
 return;
}
```

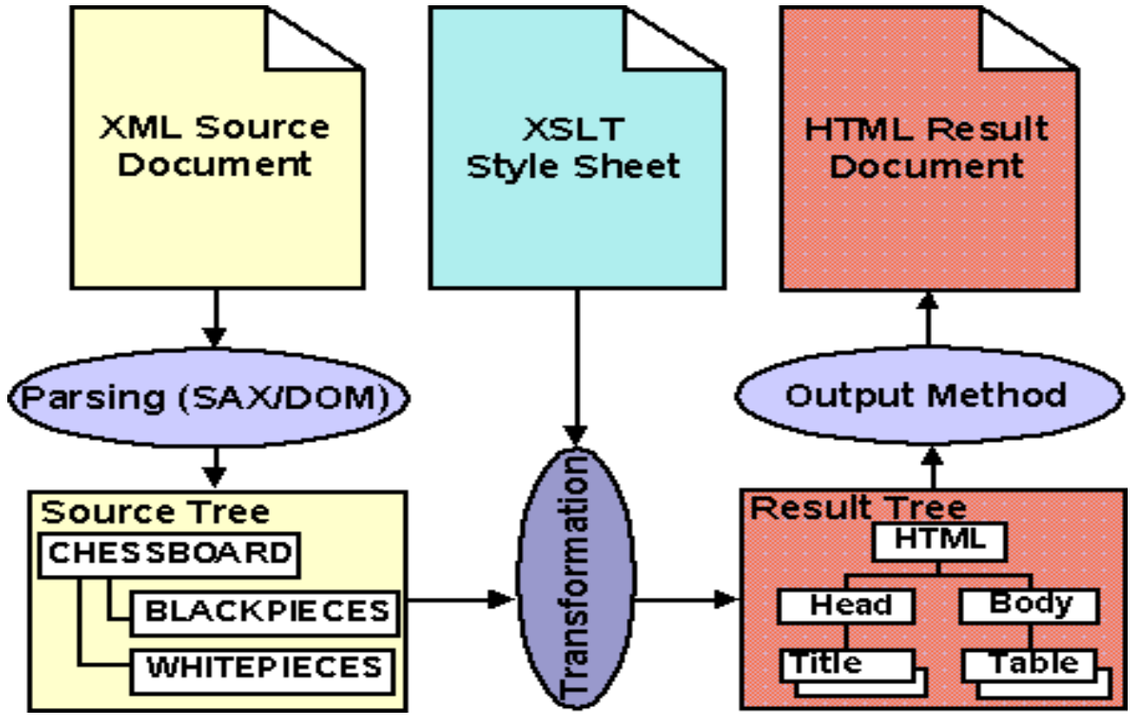
Anvendelse af XSLT

XSLT-proceduren

1. Opret en transformeringsfabrik
2. Konfigurér fabrikken
3. Opret ny transformer med bestemt ’style sheet’ vha. fabrikken
4. Sæt fejlhåndtering og URI ’resolver’
5. Gennemfør anvendelse af ’style sheet’ på XML dokument(er) og generér DOM træ(er), SAX hændelser eller skrive til output

XSLT-baseret behandling

(Vi bruger `Transformer` klassen)



XSLT-baseret behandling

```
import javax.xml.transform.*;

String styleSheetFile;
String fileToProcess;
OutputStream out;
Properties properties;
...
TransformerFactory factory
  = TransformerFactory.newInstance();
Transformer transformer = factory.newTransformer(
  new SAXSource(new InputSource(styleSheetFile)));
for (Enumeration i = properties.propertyNames();
     i.hasMoreElements();) {
 String name = (String) i.nextElement();
 transformer.setParameter(name,
   "\'" + properties.getProperty(name) + "\'");
}
transformer.transform(
  new SAXSource(new InputSource(fileToProcess)),
  new StreamResult(out));
...
```

XSLT-baseret behandling

```
<?xml version="1.0" encoding="UTF-8"?>

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

 <xsl:strip-space elements="*" />

 <xsl:output method="text"/>

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

XSLT-baseret behandling

```
 <xsl:template match="WHITEPIECES/*">
  <xsl:value-of select="concat('White ', name(),
     ': ', POSITION/@COLUMN, POSITION/@ROW)" />
  <xsl:text>
</xsl:text>
 </xsl:template>

 <xsl:template match="BLACKPIECES/*">
  <xsl:value-of select="concat('Black ', name(),
     ': ', POSITION/@COLUMN, POSITION/@ROW)" />
  <xsl:text>
</xsl:text>
 </xsl:template>

</xsl:stylesheet>
```

XML Namespaces

- Hvad hvis et XML-dokument er har en blanding af tags fra forskellige dokumenttypedefinitioner ?
  - Hvad hvis de samme tags bruges ??!?
- Løsning: tagnavne kan have et præfiks
  - <table>
    <tr>

    erstattes med

  - <h:table xmlns:h="http://www.w3.org/TR/html4/">
    <h:tr>
- XMLNamespacet (her h:) kan defineres i starten af XML-dokumentet eller løbende

XML-information i et fotografi

```
$ exiftool -xmp billede.tif -b

<rdf:RDF xmlns:rdf='http://www.w3.org/1999/02/22-rdf-syntax-ns#'>

 <rdf:Description rdf:about='' xmlns:dc='http://purl.org/dc/elements/1.1/'>
  <dc:format>image/jpeg</dc:format>
  <dc:subject>
   <rdf:Bag>
    <rdf:li>Lagt ind af EL Redaktion Modtaget 20090616</rdf:li>
   </rdf:Bag>
  </dc:subject>
 </rdf:Description>

 <rdf:Description rdf:about='' xmlns:exif='http://ns.adobe.com/exif/1.0/'>
  <exif:DateTimeDigitized>2009-02-28T11:35:54+02:00</exif:DateTimeDigitized>
  <exif:Flash rdf:parseType='Resource'>
   <exif:Fired>False</exif:Fired>
   <exif:Mode>2</exif:Mode>
   <exif:RedEyeMode>False</exif:RedEyeMode>
  </exif:Flash>
 </rdf:Description>

 <rdf:Description rdf:about='' xmlns:fwu='http://ns.fotoware.com/iptcxmp-user/1.0/'>
  <fwu:UserDefined227>17-06-2009 Clipping Path Transparens</fwu:UserDefined227>
 </rdf:Description>

 <rdf:Description rdf:about='' xmlns:photoshop='http://ns.adobe.com/photoshop/1.0/'>
  <photoshop:ColorMode>3</photoshop:ColorMode>
  <photoshop:History>E=ACT D=2009-06-16 T=16:15:56 U=elle%20modeassistent V=Til%20Billede%20Input, E=ACT
D=2009-06-17 T=14:19:19 U=jorg%20reinke V=Til%20Transparens</photoshop:History>
 </rdf:Description>

 <rdf:Description rdf:about='' xmlns:tiff='http://ns.adobe.com/tiff/1.0/'>
  <tiff:Compression>1</tiff:Compression>
  <tiff:ImageLength>3504</tiff:ImageLength>
  <tiff:ImageWidth>2336</tiff:ImageWidth>
```

Eksempel: RSS (nyhedsfødning)

- Indholdssyndikering (eng.: content syndication)
  - hente information (f.eks. nyhedoverskrifter) fra en eller flere hjemmesider og vise det i sin egen hjemmeside.
- RSS
  - står (afhængigt af hvem man spørger :-) for
    - Really Simple Syndication,
    - Rich Site Summary eller
    - RDF Site Summary.
  - Teknologien udsprang af RDF (Resource Description Framework), et system Netscape oprindelig introducerede.
- Implementation
  - Med specialiseret RSS-taglib
  - Med XML-transformering
    - I Java, JSTL, ...

```
<?xml version="1.0" encoding="iso-8859-1"?>
<rdf:RDF
 xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
 xmlns="http://purl.org/rss/1.0/"
 xmlns:taxo="http://purl.org/rss/1.0/modules/taxonomy/"
 xmlns:dc="http://purl.org/dc/elements/1.1/"
 xmlns:syn="http://purl.org/rss/1.0/modules/syndication/"
 xmlns:admin="http://webns.net/mvcb/"
>

<channel rdf:about="http://www1.dr.dk/nyheder/">
   <title>DR Nyheder</title>
   <link>http://www1.dr.dk/nyheder/</link>
   <description>Nyheder Danmarks Radio</description>
   <dc:language>da</dc:language>

   <items>
    <rdf:Seq>
     <rdf:li rdf:resource="http://www.dr.dk/nyheder/udland/article.j
     <rdf:li rdf:resource="http://www.dr.dk/nyheder/udland/article.j
     <rdf:li rdf:resource="http://www.dr.dk/nyheder/indland/article.
     <rdf:li rdf:resource="http://www.dr.dk/nyheder/udland/article.j
    </rdf:Seq>
   </items>
</channel>

<item rdf:about="http://www.dr.dk/nyheder/udland/ar
   <title>Voldsomme uroligheder i Mellemøsten</title
   <link>http://www.dr.dk/nyheder/udland/article.jht
   <description>Der har været uroligheder ved Gaza-s
</item>

<item rdf:about="http://www.dr.dk/nyheder/udland/ar
   <title>Syv dømt til døden i Rwanda</title>
   <link>http://www.dr.dk/nyheder/udland/article.jht
   <description>I Rwanda er syv personer blevet dømt
vidne til folkemordet for 10 år siden.</description>
</item>

<item rdf:about="http://www.dr.dk/nyheder/indland/article.jhtml?articleID=152958">
   <title>Brigadegeneral langer ud efter regeringen</title>
   <link>http://www.dr.dk/nyheder/indland/article.jhtml?articleID=152958</link>
   <description>Regeringen ser ud til at være vendt 180 grader på cirka tre måneder i
spørgsmålet om en reform af forsvaret.</description>
</item>
```

Implementation med JSTL

```
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<%@ taglib uri="http://java.sun.com/jsp/jstl/xml" prefix="x"%>
<html>
<head><title>Syndikering med JSTL og XML</title></head>
<body>

<c:import var="rssKilde" url="http://rss.asdf.dk/dr.rss"/>
<x:parse var="rss" xml="${rssKilde}"/>

<h1>
  <x:out select="$rss//*[name()='channel']/*[name()='description'][1]"
         escapeXml="false"/>
</h1>

Læs flere nyheder
<a href="<x:out select="$rss//*[name()='channel']/*[name()='link'][1]"/>"

<ul>
  <x:forEach select="$rss//*[name()='item']">
    <li>
      <x:out select="./*[name()='title']" escapeXml="false"/><br>
      <font size="-1">
        <x:out select="./*[name()='description']" escapeXml="false"/>
        <a href="<x:out select="./*[name()='link']"/>">mere info</a><br><br>
      </font>
    </li>
  </x:forEach>
</ul>

</body>
</html>
```

XML - det mindst ringe alternativ

- XML-parsning/generering er besværligt...
-

StAX - Streaming API for XML

- "As fast as SAX, almost as convenient as DOM"
- XML-parsning foregår a la med en iterator (ved at iterere igennem elementerne)

```
public interface XMLStreamReader {
  public int next();                // retur f.eks. START_ELEMENT, eller END_DOCUMENT
  public boolean hasNext();
  public String getText();
  public String getLocalName();     // tagnavn uden præfiks
  public String getNamespaceURI();
  ...
}

XMLInputFactory factory = XMLInputFactory.newInstance();
XMLStreamReader reader = factory.createXMLStreamReader(new FileReader(file));
...
while (true) {
  int event = parser.next();
  if (event == XMLStreamConstants.END_DOCUMENT) {
     parser.close();
     break;
  }
  if (event == XMLStreamConstants.START_ELEMENT) {
      System.out.println(parser.getLocalName());
  }
}
```

StAX - Streaming API for XML

- XML-skrivning understøttes også, med lignende API

```
public interface XMLStreamWriter {

  public void writeStartElement(String localName)
  public void writeEndElement()
  public void writeCharacters(String text)

...
}
```

- StAX er nem og næsten lige så hurtig til SAX
  - Varianter findes der er hurtigere end SAX
- Oprindeligt Java, nu også C-implementationer

StAX - Streaming API for XML

Feature	StAX	SAX	DOM	TrAX
API Type	Pull, streaming	Push, streaming	In memory tree	XSLT Rule
Ease of Use	High	Medium	High	Medium
XPath Capability	No	No	Yes	Yes
CPU and Memory Efficiency	Good	Good	Varies	Varies
Forward Only	Yes	Yes	No	No
Read XML	Yes	Yes	Yes	Yes
Write XML	Yes	No	Yes	Yes
Create, Read, Update, Delete	No	No	Yes	No

Kilde: http://java.sun.com/webservices/docs/1.6/tutorial/doc/

Nem generering af XML fra Java-objekter

-
- http://javabog.dk/JSP/kapitel11.jsp#afsn11.2.2