

Luento 6: XSLT

AS-0.110 XML-kuvauskielten perusteet

Janne Kalliola

XSL Transformations

- XSLT
 - tyylisivun rakenne
 - mallineet
 - nykyinen solmu
- Edistyneemmät piirteet
 - toisto
 - muuttujat
 - nimetyt mallineet
 - ehdolliset osiot
 - moodit
 - kopiointi
 - järjestäminen
 - numerointi

- XSL Transformations (XSLT) on XML-pohjainen kieli XML-dokumenttien muuttamiseen muodosta toiseen
 - XSLT-tyylisivu kuvaa säännöt, joiden avulla XSLT-prosessori käsittelee sisään luettavaa dokumenttia (input document) ja tuottaa ulos tulevaa dokumenttia (output document)
 - XSLT-tyylisivu voi lisätä ja poistaa elementtejä ja attribuutteja sekä käsitellä leipätekstiä
- Tuotettava dokumentti on joko XML:a tai tavallista tekstiä
 - yleensä lopputuloksena on toinen XML-dokumentti
 - jos halutaan tuottaa HTML:a, tuotetaan käytännössä XHTML:a

Tyylisivun rakenne (1/2)

- XSLT-tyylisivu koostuu joukosta määrittäjiä ja malleineista (template)
 - template voidaan suomentaa myös sapluunaksi
- Määrittäksillä vaikutetaan tyylisivun tuottamaan dokumenttiin liittyviä asioita tai määritellä muuttujia myöhempää käyttöä varten
- Mallineet sisältävät dokumentin muunnossa käytettävät säännöt, sisään luetun dokumentin käsittelyohjeita ja tuotetun dokumentin elementtejä
 - sana "malline" kuvaa toimintaa melko hyvin:
 - osa tuotettavan dokumentin elementeistä on määritelty pysyvästi
 - näiden elementtien väliin tuotetaan lisää sisältöä käsittelyohjeiden pohjalta

- Tyylisivun kaikki elementit ovat elementin `<xsl:stylesheet>` sisällä
 - kaikki XSLT:n elementit, ts. komennot, ovat nimiavaruudessa <http://www.w3.org/1999/XSL/Transform>
 - nimiavaruus lyhennetään yleensä xsl:

```
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
...
</xsl:stylesheet>
```
 - Tyylisivu sisältää ylätason määrittelyt ja mallineet
 - määrittelyt ohjaavat tyylisivun toimintaa
 - tyhjien merkkien (whitespace) poistaminen tai säästäminen
 - sisentäminen (indent)

- Malline koostuu kahdesta osasta
 - mallineen laukaiseva sääntö
 - mallineen sisältö
- Sääntö määritetään XPath-lausekkeella
- Mallineen sisällön rakenteen määrää tuotettavan dokumentin rakenne ja sisältö
 - malline voi tuottaa uusia elementtejä ja attribuutteja, muun muassa, tuotettuun dokumenttiin
 - malline voi valita sisään luetun dokumentin elementtejä
 - kutsua uusia mallineita elementtien pohjalta
 - käsitellä elementit itse
 - malline voi sisältää ehtoja ja toistoja

```
<xsl:template match="sect1">
  <h1><xsl:value-of select="title/text()" /></h1>
  <xsl:apply-templates/>
</xsl:template>

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

<xsl:template match="title"/>
```

- XSLT-prosessori käsittelee aina yhtä XML-dokumentin solmua (node) kerrallaan
 - tätä solmua kutsutaan nykyiseksi solmuksi (current node, context node)
- XSLT-tyylisivun mallineista valitaan se, jonka sääntö osuu tarkiten nykyiseen solmuun
 - sääntöjen valintajärjestykseen on erillinen algoritmi
 - vain yksi malline valitaan
 - tämä malline ohjaa toimintaa ja käskyttää XSLT-prosessoria jossakin vaiheessa valitsemaan uuden elementin tai muun XML-dokumentin osasen nykyiseksi solmuksi
 - tämän jälkeen etsitään uusi malline
- Prosessori aloittaa dokumentin käsittelyn juurielementistä
 - nykyinen solmu viittaa juurielementtiin
 - käsittely jatkuu edellä kuvatulla tavalla

- Malline voi käynnistää mallineen haun mille tahansa XML-dokumentin osalle
 - yleisin tapaus on käsitellä nykyisen solmun alla (sisällä) olevat solmut: `<xsl:apply-templates/>`
 - voidaan myös valita vain tietyt solmut käsittelyyn:
`<xsl:apply-templates select="name|street|city|postal-code"/>`
 - valinta voi kohdistua muualla oleviin solmuihin:
`<xsl:apply-templates select="//distant"/>`
- Myös attribuutti, tekstisirpale tai muu XML-dokumentin osa voidaan valita
 - kaikkiin valintoihin löytyy tuki XPath-lausekkeista
 - valinta voi kohdistua useaan solmuun kerralla
 - solmut käsitellään tällöin dokumenttijärjestyksessä (document order)

- XSLT-tyylisivuissa yleensä poimitaan suuri joukko arvoja sisään luettavasta dokumentista ja sijoitetaan tuotettavaan dokumenttiin
 - attribuuttien arvoja siirretään tai kirjoitetaan näkyviin
- Arvojen poiminta tapahtuu elementillä `<xsl:value-of select="xpath-lauseke"/>`
 - elementti poimii XPath-lausekkeen mukaisen arvon ja sijoittaa sen tuotettavaan dokumenttiin
 - mikäli arvoa ei ole, elementti ei tuota mitään

- XSLT mahdollistaa elementtien luonnin dynaamisesti
 - käytetään elementtiä `<xsl:element>`

```
<xsl:element name="{@element-name}">
  <xsl:apply-templates/>
</xsl:element>
```
 - Vastaavasti myös attribuutteja voidaan luoda lennossa
 - elementti on tietenkin `<xsl:attribute>`
 - attribuutti liitetään viimeksi määriteltyyn elementtiin

```
<a>
  <xsl:attribute name="href">
    <xsl:value-of select="substring(@src, 2)"/>
  </xsl:attribute>
  <xsl:value-of select="@title"/>
</a>
```

- Kaikkia attribuutteja ei tarvitse luoda elementillä `<xsl:attribute>`
 - jos attribuutin arvo kopioidaan suoraan toisesta attribuutista, se voidaan tehdä mallineen sisällä aaltosulkujen avulla

```
<a href="{@src}" target="{@target}">
  <xsl:value-of select="@title"/>
</a>
```
 - huomaa, että XSL:n omien elementtien sisällä aaltosulkuja ei tarvita
 - aaltosulkuja ei voida käyttää, mikäli XPath-lauseke on monimutkainen
 - vain attribuuttien arvoja voidaan poimia suoraan
- Attribuutin arvo voi tietysti olla kiinteäkin

```
<a href="http://www.example.com/">Go to example.com</a>
```

Edistyneemmät piirteet

Edistyneemmät piirteet

- Edellä esitelyillä toiminnoilla pystyy muuntamaan XML-dokumentin muodosta toiseen
 - keinot kuitenkin loppuvat, jos sisään luetun dokumentin elementtejä pitää käsitellä useasti tai lopputulos riippuu luetun dokumentin attribuuttien arvoista
 - vastaavasti XSLT-tyylisivun toimintaa saatetaan haluta ohjata ulkopuolelta
- Ratkaisuja on useita erilaisia
 - toisto
 - muuttujien käyttö
 - ehdolliset osiot mallineissa
 - erilaisten moodien käyttö
 - mallineiden nimeäminen ja kutsuminen ilman säännön laukeamista

- XSLT sisältää hyvin yksinkertaisen toistorakenteen
 - se voi toistaa tietyn mallineen osan kaikille tietyllä XPath-lausekkeella valituille solmuille
 - ei ole mahdollisuutta toistaa asioita esimerkiksi viittä kertaa
 - toistojen määrä riippuu aina sisään luettavasta dokumentista
- Toisto toteutetaan elementillä `<xsl-for:each>`
 - elementissä on XPath-lauseke sääntönä ja elementin sisällä oleva XSLT-koodi toistetaan kaikille säännön valitsemille elementeille:

```
<xsl-for-each select="name">
  <xsl:value-of select="@first">.<xsl:value-of
select="@last">@hut.fi
  <br/>
</xsl-for-each>
```

- Tyylisivulla voi olla käytössä muuttujia
 - muuttujat asetetaan elementillä `<xsl:variable name="...">` ja niiden arvoa ei voi muuttaa myöhemmin
 - muuttujat ovat käytössä koko tyylisivulla
 - muuttujiin viitataan syntaksilla `$nimi`
 - muuttujiin voidaan viitata myös `{$nimi}` mallineen elementtien sisällä, vrt. attribuutit
- Muuttujat ovat näppäriä laadittaessa helposti säädettäviä tyylisivuja
 - muokattavat tiedot sijoitetaan muuttujiin, joita säätämällä tyylisivun toiminta tai tuotettava dokumentti muuttuu
- Vaikka muuttujan arvoa ei voi muuttaa, se voidaan määrittää uudelleen alemmalla tasolla
 - uusi arvo piilottaa aiemman arvon
 - aiempi arvo on käytössä palattaessa takaisin ylemmälle tasolla

- Joskus malline voi olla hyvin monimutkainen
 - erityisesti ehtolauseet pidentävät mallinetta
 - mallineessa voi olla useita vaihtoehtoisia toteutuksia
- Malline voi tällöin kutsua toista mallinetta
 - kutsuttavalle mallineelle on määritelty nimi:
`<xsl:template name="call-me">...</xsl:template>`
 - tällöin ei saa määrittää sääntöä
 - mallineen kutsu tapahtuu elementillä `<xsl:call-template name="..." />`
 - kutsussa nykyinen solmu ei muutu
 - kutsuttu malline käsittelee samaa tietoa kuin kutsujakin
 - kutsutun mallineen suorituksen jälkeen palataan suorittamaan kutsuvaa mallinetta

- Kutsuttavalla mallineelle voidaan lähettää parametreja
 - muistuttaa pitkälti proseduraalisten ohjelmointikielten funktio- tai metodikutsuja
 - parametrit ovat mallineen sisäisiä muuttujia, joiden arvoa kutsuja voi säätää
 - viittaus parametrien arvoihin on sama kuin muuttujiin:
`$parametri`
- Parametrit kirjoitetaan kutsun sisään
`<xsl:call-template name="call-me">`
`<xsl:with-param name="test">...</xsl:with-param>`
`</xsl:call-template>`

- Kutsuttavan mallineen täytyy määrittää parametrit:

```
<xsl:template name="call-me">  
  <xsl:param name="test">...</xsl:param>  
  ...  
</xsl:template>
```

- samalla kutsuttava malline asettaa parametreille oletusarvot
 - kaikkia parametreja ei ole siis pakko asettaa

- Myös tyylisivulle voidaan määrittää parametreja
 - käytetään vastaavaa syntaksia kuin edellä, mutta parametrimäärittely sijoitetaan tyylisivun päätasolle

```
<xsl:stylesheet version="1.0"  
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform">  
  <xsl:param name="show-titles">true</xsl:param>  
  <xsl:param name="author"/>  
</xsl:stylesheet>
```

- Parametrit annetaan XSLT-prosessorille ja se välittää ne edelleen tyylisivulle
 - komentoriviargumentit
 - ohjelmointikielissä metodikutsulla

- XSLT:ssa on kaksi rakennetta ehdollisten osioiden laatumiseksi
 - molemmissa käytetään samaa testirakennetta
 - testinä on XPath-lauseke ja se läpäistään, jos
 - lausekkeessa oleva ehto toteutuu
 - lauseke evaluoituu ei-tyhjäksi joukoksi (ei erillistä ehtoa)
 - ehtoina ovat normaalit =, <, > ja !=
 - < ja > pitää kirjoittaa XML-syntaksin takia < ja >

- If-lauseke on yksinkertaisempi ehdoista
 - sen sisällä oleva mallineen osa suoritetaan ehdon toteutuessa
 - lausekkeessa ei ole **else if** tai edes **else** -osaa

```
<xsl:if test="@target">  
  <xsl:attribute name="target">  
    <xsl:value-of select="@target"/>  
  </xsl:attribute>  
</xsl:if>
```

- Choose-lauseke mahdollistaa usean ehtolausekkeen yhdistämisen
 - vain yksi ehtolausekkeista suoritetaan
 - ensimmäisen onnistuneen testin jälkeen jatketaan choose-lausekkeen jälkeisistä elementeistä
 - lausekkeessa voi olla myös erillinen vaihtoehto tilanteella, jossa yksikään testi ei toteutunut

```
<xsl:choose>
  <xsl:when test="@number-type = '1'">
    <xsl:call-template name="number-numbers"/>
  </xsl:when>
  <xsl:when test="@number-type='a'">
    <xsl:call-template name="number-letters"/>
  </xsl:when>
  <xsl:otherwise>
    <xsl:call-template name="no-numbers"/>
  </xsl:otherwise>
</xsl:choose>
```


- Joskus sama XML-dokumentin osa halutaan käsitellä useita kertoja eri konteksteissa
 - hyvä esimerkki on sisällysluettelon automaattinen laatiminen
 - sisällysluettelo rakentuu otsikoista, jotka täytyy esittää myös leipätekstin sisällä
- Mallineeseen voidaan liittää moodi (mode)

```
<xsl:template match="header[@level=1]" mode="toc">
```

```
...
```

```
</xsl:template>
```

- mallinetta ei voida valita, ellei myös moodia ole asetettu käynnistettäessä mallineen etsintää:

```
<xsl:apply-templates mode="toc"/>
```

- XSLT mahdollistaa myös kopiointin sisään luettavasta dokumentista tuotettavaan dokumenttiin
 - kopiointissa dokumenttisiirpale (document fragment) kopioidaan sellaiseen
 - ei tarvita erillisiä sääntöjä jokaiselle osalle
 - toisaalta ei voida myöskään vaikuttaa lopputulokseen
- Kopiointi on osa mallinetta
 - kopiointi suoritetaan elementillä `<xsl:copy-of>`
 - elementti voi määritellä myös XPath-lausekkeen, joka määrää kopioitavat elementit

```
<xsl:copy-of select="..." />
```
 - jos lauseketta ei määritellä, kopioidaan nykyinen solmu

- XSLT tarjoaa mahdollisuuden järjestää sisään luettavan dokumentin elementit uuteen järjestykseen ennen mallineen valintaa
 - käytetään elementtiä `<xsl:sort select="...">`
 - järjestäminen tapahtuu `select`-lausekkeen tuottamien arvojen pohjalta
 - elementille voi antaa useita lisäattribuutteja järjestämisen ohjaamiseksi
- Järjestäminen tapahtuu joko `<xsl:apply-templates>` tai `<xsl:for-each>` elementtien sisällä

```
<xsl:apply-templates select="sect1/title">
  <xsl:sort select="text()"/>
</xsl:apply-templates>
```
- Järjestämiselementtejä voi olla useita peräkkäin

- XSLT pystyy tuottamaan juoksevia numeroita elementillä `<xsl:number>`
 - elementin attribuuteilla määritellään numeroiden tuottamisen yksityiskohdat
 - numerot tuotetaan yleensä laskemalla nykyistä elementtiä edeltävät, annettua sääntöä vastaavat elementit
 - numerointi aloitetaan yleensä ykkösestä, mutta tätä voidaan säätää
 - laskeminen voi olla monitasoista
 - arabialaisten numeroiden sijaan voidaan tuottaa mm. kirjaimia tai roomalaisia numeroita
 - monitasoisissa numeroinneissa jokaisella tasolla voi olla oma tyyli, esimerkiksi "A.1.1"

- XSL Transformations (XSLT) on XML-pohjainen kieli XML-dokumenttien muuttamiseen muodosta toiseen
 - XSLT-tyylisivu koostuu joukosta määrittämiä ja mallineista
 - tyylisivu kuvaa säännöt, joiden avulla XSLT-prosessori käsittelee sisään luettavaa dokumenttia ja tuottaa ulos tulevaa dokumenttia
 - XSLT-prosessori käsittelee aina yhtä XML-dokumentin solmua kerrallaan
- XPath on olennainen osa XSLT:ta
 - käytetään sekä säännöissä että testeissä
- XSLT tarjoaa laajan joukon elementtejä dokumenttien muokkaamiseen

Kysymyksiä? Kommentteja?