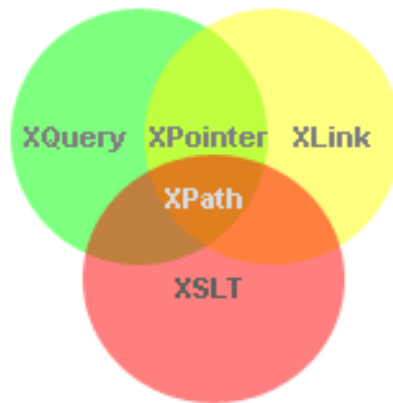


## XPATH Perusteet - teoria



Kuva 1. XPath on tärkeä osa useita eri tekniikoita

### 1. XPathin tehtävä

XPath (XML Path Language) on kyselykieli, jolla XML-dokumentista voidaan poimia yksittäisiä elementtejä, attribuutteja, tekstisisältöjä tai niiden joukkoja (dokumentin osia). Näitä saatuja tuloksia joko käytetään hyväkseen suoraan tai lähtökohtana erilaisille toimenpiteille, esimerkiksi XSLT-muunnoksille. XPath on tärkeä koska monet XML-tekniikat hyödyntävät sitä (ks. Kuva 1).

### 2. XPath-lauseke

XPath-lausekkeen rakenne

- XPath-lauseke koostuu joukosta *askelia (step)*
- askelista muodostuu *polku (path)* alkupisteestä (*current node*) loppujoukkoon
- askeleet erotetaan toisistaan kauttaviivoilla */step/step/step*
- jokainen askel muokkaa valittua joukkoa (*set*)
- jos askel päättyy tyhjään joukkoon, palautetaan tyhjä joukko
- askel määrää *siirtymän (axis)*, *solmutestin (node test)* ja mahdollisesti joukon *ehtoja (predicates)*

Askel

Askeleen rakenne: `axis::node-test[predicates][predicates...]`

**`axis::node-test[predicates][predicates...]`**

**Step esimerkki:** `child::title[attribute::lang='fi']` valitsee kaikki lapsielementit, joiden nimi on 'title' ja joiden 'lang' attribuuttien arvo on 'fi'.

### 3. XPath: Lausekkeen evaluointi

Kannattaa muistaa, että XPath-lauseke evaluoidaan aina puumuotoisessa XML:ssa. Elementit, attribuutit ja tekstisisällöt ovat puun solmuja (node)

Lauseke evaluoidaan askel askeleelta:

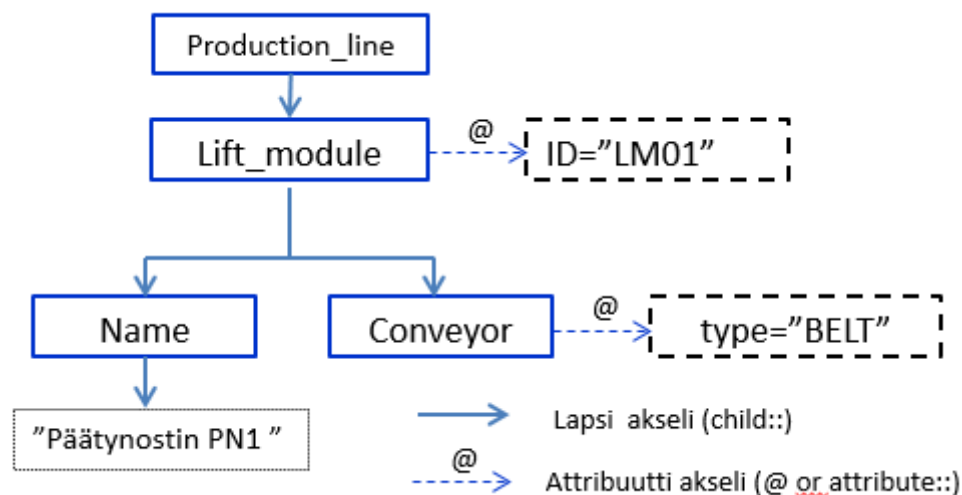
- ensimmäinen askel otetaan esimerkiksi ns. nykyisestä elementistä (current node/ context node) tai dokumentin juuresta / tai juurielementistä
- seuraava askel otetaan siitä, mihin edellinen jäi (se on siis uusi kontekstisolmu)
- jos askelta ei pystytä ottamaan, lauseke palauttaa tyhjän joukon

Seuraavan esimerkin lauseke poimisi xml-dokumentista (kuva 2) sen 'Lift\_module' elementin, jonka ID-attribuutin arvo on 'LM01', kaikkien 'Conveyor'-elementtien 'type'-attribuuttien arvot (esim. 'BELT')

ESIM: `/child::Production_line/child::Lift_module[attribute::ID='LM01']/child::Conveyor/attribute::type`

Lyhennetty syntaksi sallii yleisimmän siirtymän, lapsi-siirtymän (child::) poisjättämisen sekä attribute:: siirtymän lyhennetyin merkinnän '@' käytön, jolloin edellisen lyhennetty versio olisi:

ESIM: `/Production_line/Lift_module[@ID='LM01']/Conveyor/@type`



Kuva 2. Esimerkki XML dokumentin puurakenteesta ja erilaisista poluista siinä.

### 4. Esimerkkejä peruslausekkeista

Esimerkkejä perussiirtymistä lausekkeissa:

- Nykyisen elementin title-lapsielementtien valinta: **child::title** tai lyhennettynä **title**
- Nykyinen elementti voidaan toki merkitä alkuun: **self::node()/child::title** tai **./title**
- Nykyisen elementin src-attribuutin valinta: **attribute::src** tai **@src**

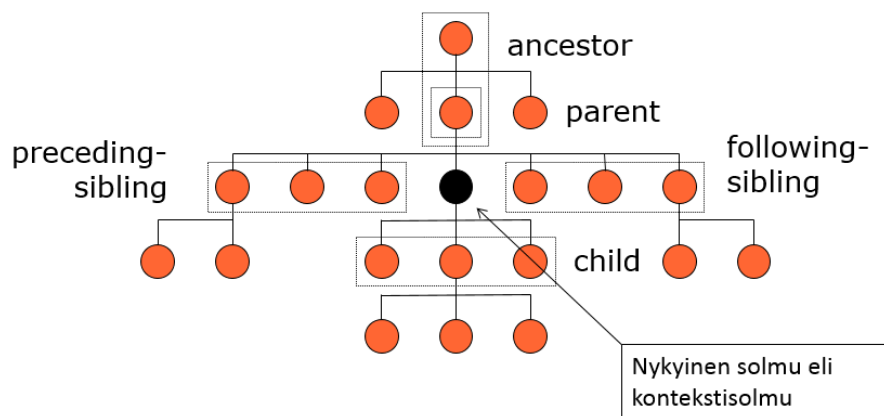
Huomaa, että seuraavat lausekkeet palauttavat eri joukon:

- **self::node()/child::node()** – palauttaa kaikki lapsisolmut, myös tekstisisällöt
- **./\*** – palauttaa vain nykyisen elementin lapsielementit, mutta ei sen tekstisisältöä

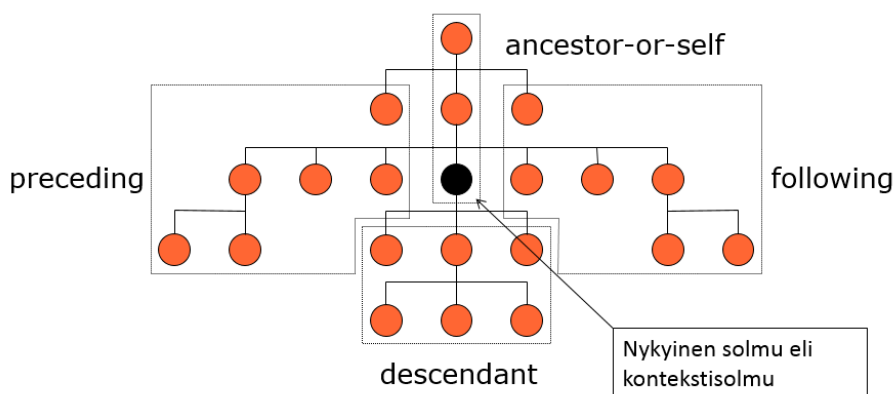
## 5. XPath: Siirtymät XML-puussa

XPath siirtymät mahdollistavat liikkumisen kaikkiin suuntiin XML-puussa:

- **parent** tai **..** (kaksi pistettä) valitsee isäelementin
- **child** tai tyhjä valitsee lapsisolmut
- **/** valitsee dokumentin juuren (jos polun ensimmäisenä merkkinä)
- **descendant** valitsee kaikki lapsielementit, myös lapsien lapset jne.
- **descendant-or-self** tai **//** valitsee myös nykyisen solmun
- **ancestor** valitsee kaikki esi-isät
- **ancestor-or-self** valitsee myös nykyisen solmun
- **following** ja **preceding** valitsevat kaikki dokumentissa seuraavat tai edeltävät solmut
- **following-sibling** ja **preceding-sibling** valitsevat seuraavat tai edeltävät sisarsolmut
- **self** tai **.** (piste) valitsee nykyisen solmun



Kuva 3: Havainnekuva1 erilaisten siirtymien (Axis) valitsemista puun solmujoukoista (node set)



Kuva 4: Havainnekuva2 erilaisten siirtymien (Axis) valitsemista puun solmujoukoista (node set)

## 6. Solmutesti -Siirtymien tarkentaminen

Siirtymälle voidaan asettaa ehtoja solmutestillä. Askel valitsee ainoastaan ne solmut, jotka ovat siirtymän päässä nykyisestä joukosta ja jotka täyttävät solmutestin ehdot. Solmutestillä voidaan valita: tietyt nimetyt elementit, pelkkä leipäteksti, kommentit, kaikki solmut/noodit (eli myös tekstisisällöt) tai kaikki elementit ( Merkki \* palauttaa kaikki elementit):

1. following-sibling::*image*
2. child::*text()* tai *text()*
3. parent::node()/child::*comment()* tai *../comment()*
4. child::*node()*
5. descendant-or-self::\* tai *//\**

## 7. Ehdot

Valintaa voidaan tarkentaa edelleen määrittelemällä haluttu määrä ehtoja valittaville solmuille (*predicates*). Ehdot määritetään hakasuluissa solmutestin jälkeen [*ehto*]. Useita ehtoja voidaan määrittellä peräkkäin, joista jokaisen on toteuduttava.

ESIM: *child::title[attribute::lang='fi']* tai *title[@lang='fi']*

## 8. XPath – operaattorit ja Funktiot

Xpath lausekkeissa voidaan myös käyttää useita aritmeettisia ja loogisia operaattoreita sekä vertailu-operaattoreita (Ks. Tarkemmin kalvosetti: Xpath.pdf s24).

Usein hyödyllinen operaattori on | joka palauttaa kahden solmujoukon unionin.

Esimerkiksi *//book | //cd* palauttaisi kaikki book ja cd elementit.

XPath määrittää myös joukon funktioita, joita voidaan käyttää lausekkeiden ehdoissa tai lausekkeiden sijaan palauttamaan jotakin tietoa dokumentista. funktiot ovat tyyppitettyjä ja osa funktioista voi ottaa solmujoukon (node set) parametrikseen. (Ks. Tarkemmin kalvosetti: Xpath.pdf)

Esimerkiksi XPath lauseke: */bookstore/book[@category = 'web']/title[string-length() > 12]* voisi palauttaa seuraavan tuloksen

RESULT: *<title lang="en">XQuery Kick Start</title>*

## 9. Joitakin hyödyllisiä XPath- funktioita

*XPath:Merkkijonofunktiot*

- *concat(s1, s2, ...)* yhdistää argumentit yhdeksi merkkijonoksi. Argumenttien täytyy olla merkkijonoja
- *contains(haystack, needle)* palauttaa true, jos toinen argumentti löytyy ensimmäisestä
  - esim. *contains('XML','XM')* – palauttaa: true
- *substring-before(s1, s2)* palauttaa merkkijonon, joka päättyy siihen, mistä s2 löytyy s1:stä
  - esim. *substring-before('Punainen', 'a')* – palauttaa: 'Pun'

### Solmujoukkofunktiot

(huom: 'ns':llä on merkitty jotain XPath-lauseketta, joka palauttaa solmujoukon)

- *count(ns)* laskee solmujoukon solmujen lukumäärän
- *local-name(n)* palauttaa solmun paikallisen nimen (siis ilman nimiavaruutta)
- *sum(ns)* laskee solmujoukon summan, solmut muutetaan numeroiksi

### Nykyiseen solmuun liittyvät funktiot

- *position()* palauttaa nykyisen solmun järjestysnumeron nykyisessä solmujoukossa
- *last()* palauttaa nykyisen solmujoukon viimeisen solmun järjestysnumeron

## Lisää esimerkkejä XPath kalvosetissä

XPATH kalvosetti, *XML\_XPath\_extra.pdf*, sisältää paljon esimerkkejä XPath lausekkeista ja listoja erityyppisistä Xpath funktioista

- Xpath:n siirtymä- ja solmutestityypit: kalvot 17-22
- XPath – operaattorit ja funktiot: kalvot 23-33
- Lisää Xpath esimerkkejä: kalvot 34-39

## Lähteitä

- Tutoriaalit: <http://www.w3schools.com/>
- XPATH esimerkkejä: [http://www.w3schools.com/xpath/xpath\\_examples.asp](http://www.w3schools.com/xpath/xpath_examples.asp)
- W3C:n Määrittelyt: <http://www.w3.org/XML/Core/>

## Online tools:

- Online tool: xpath, xslt, schema validate: <http://www.xpathtester.com>
- Online tool: xpath, xslt, schema validate, reg-exp: <http://www.freeformatter.com>
- Well-formed validator by w3c: [http://validator.w3.org/#validate\\_by\\_input](http://validator.w3.org/#validate_by_input)