ELEC-C8203 Automaatiojärjestelmät 2 - XML

XML merkintäkielten perusteet

Pekka Aarnio

Sisältö

- OSA 1
  1. XML ja Puut
  2. XML Extensible Markup Language
  3. XML-dokumentin rakenne

XML esimerkit

MIKÄ XML?

Mitä on XML?

- XML (Extensible Markup Language) on tiedon tallentamista ja rakenteen kuvaamista koskeva määrittely.

- Kysymyksiä *esim_1-1.xml* dokumentista
  1. Mitä tietoa dokumenttiin on tallennettu?
  2. Minkälainen on tiedon rakenne?
  3. Minkälaista merkkausta (*tags*) on käytetty tiedon rakenteen kuvaamiseen?
  4. Entä selviääkö dokumentista tiedon merkitys?

```
<?xml version="1.0" encoding="UTF-8"?>
<kurssi>
    <koodi>AS-0.1502</koodi>
    <nimi>Automaatio II</nimi>
    <moduuli>
        <luennot>XML:n perusteet</luennot>
        <harjoitukset>harjoitustehtäviä</harjoitukset>
    </moduuli>
    <moduuli>
        <luennot>Kappaletavara-automaatio</luennot>
        <harjoitukset/>
    </moduuli>
    <moduuli>
        <luennot>Prosessiautomaatio</luennot>
        <harjoitukset/>
    </moduuli>
</kurssi>
```

Kuva: *esim_1-1.xml*

Mitä on XML?

- Joitain vastauksia kysymyksiin dokumentista:

1. Dokumentti sisältää tietoa tästä kurssista.
2. Tiedot on esitetty hierarkkisena rakenteena, joka kertoo, että
   - kurssitiedot koostuvat *koodista, nimestä ja moduulitiedoista* ja että
   - Moduulien tiedot kertovat jotain *luennoista ja harjoituksista*
3. Tietojen ja sen rakenteen kuvaamiseen on luotu seuraavat merkinnät: *kurssi, koodi, nimi, moduuli* jne.
4. Tiedon merkitys selviää osittain ihmislukijalle, mikäli merkinnöissä käytetyt käsitteet ovat ymmärrettäviä. Tietokone ei ymmärrä käytettyjen käsitteiden merkitystä.

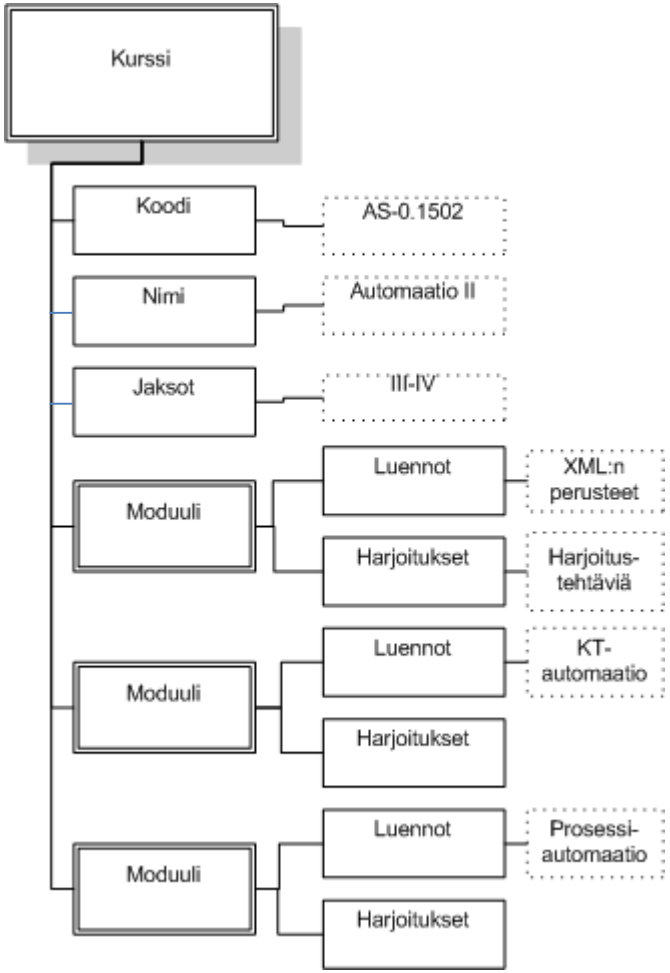
```
<?xml version="1.0" encoding="UTF-8"?>
<kurssi>
    <koodi>AS-0.1502</koodi>
    <nimi>Automaatio II</nimi>
    <moduuli>
        <luennot>XML:n perusteet</luennot>
        <harjoitukset>harjoitustehtäviä</harjoitukset>
    </moduuli>
    <moduuli>
        <luennot>Kappaletavara-automaatio</luennot>
        <harjoitukset/>
    </moduuli>
    <moduuli>
        <luennot>Prosessiautomaatio</luennot>
        <harjoitukset/>
    </moduuli>
</kurssi>
```

Kuva: *esim_1-1.xml*

Huomaa, että:
Tässä XML-dokumentissa esitetyn tiedon rakenne vastaa osittain myös kohteen, kurssin, todellista sisäistä loogista rakennetta!

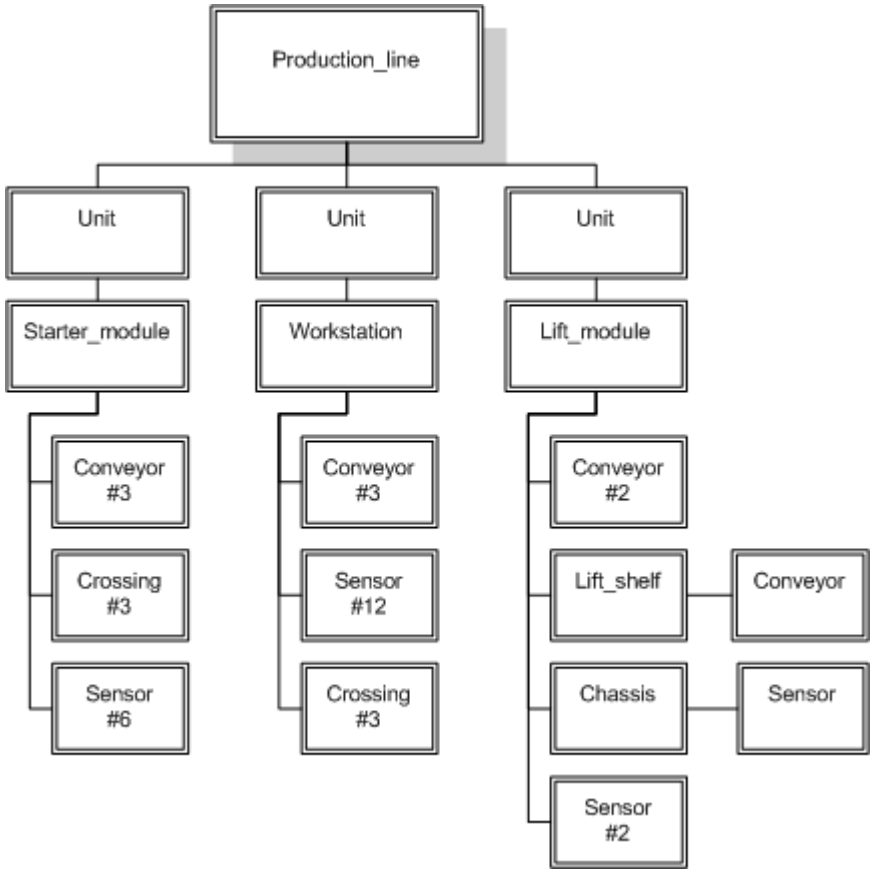
XML - Puurakenne

- XML-merkintäkielillä voidaan kuvata hierarkkisia puumuotoisia tietorakenteita (*tree*)
- Merkinnöillä nimetään puun solmut (*node*)
- Puun lehtisolmut (*leaf nodes*) sisältävät varsinainen tekstitiedon



XML - Puurakenne

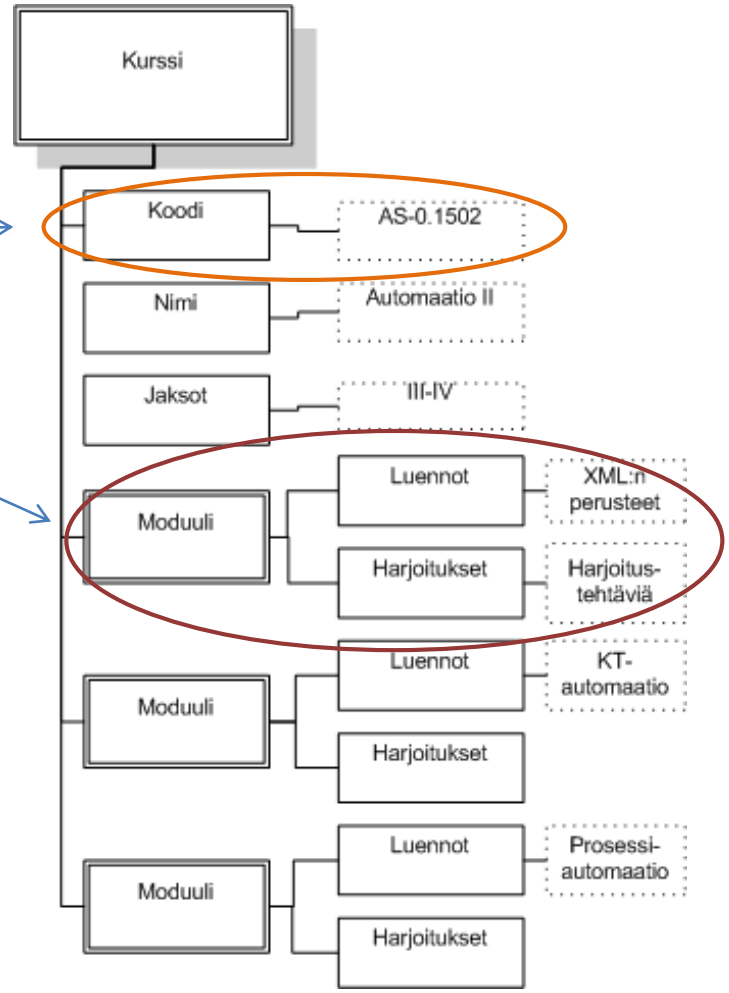
- Puulla on vain yksi juurisolmu (*root node*)
- Solmuilla voi olla monta lapsi-solmua (*child node*) mutta vain yksi isä-solmu (*parent node*)
- Kaaret (*branch*) puun solmujen välillä (*parent-child relation*) voivat vastata mitä tahansa todellista relaatiota esitettävän kohteen osien/objektien välillä
  - osa-kokonaisuus (tai kokonaisuus-osa);
  - luokka-instanssi;
  - ominaisuus;
  - jokin assosiaatio;
  - jne.
- Sama tietosisältö voidaan siis esittää monella erilaisella tavalla puurakenteena (aiheesta lisää attribuuttien yhteydessä)



Kuva: Tuotantolinjan fyysisen rakenteen esitys *kokonaisuus-osa* hierarkiana.

XML - Puurakenne

```
<?xml version="1.0" encoding="UTF-8"?>
<kurssi>
    <koodi>AS-0.1502</koodi>
    <nimi>Automaatio II</nimi>
    <jaksot>III-IV</jaksot>
    <opintopisteet>5</opintopisteet>
    <moduuli>
        <luennot>XML:n perusteet</luennot>
        <harjoitukset>harjoitustehtäviä</harjoitukset>
    </moduuli>
    <moduuli>
        <luennot>Kappaletavara-automaatio</luennot>
        <harjoitukset/>
    </moduuli>
    <moduuli>
        <luennot>Prosessiautomaatio</luennot>
        <harjoitukset/>
    </moduuli>
    <moduuli>
        <excursiot/>
    </moduuli>
</kurssi>
```



XML - Esimerkki

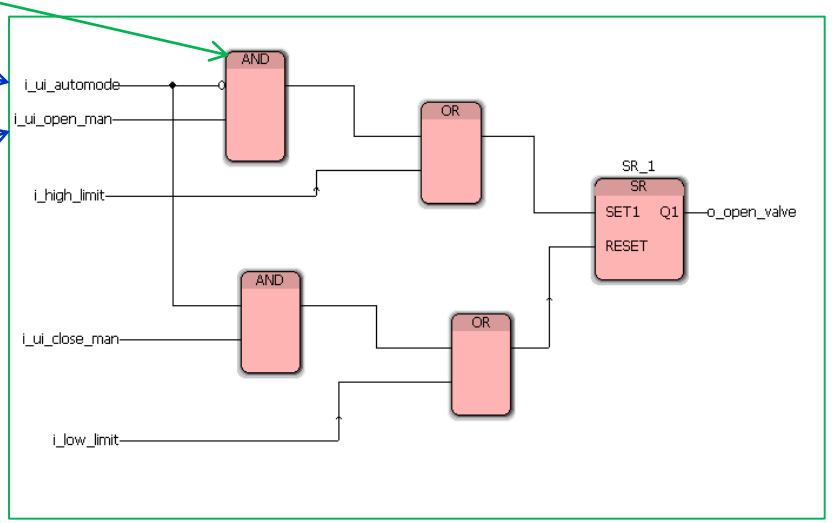
- Tuoteluettelot esitetään usein yksinkertaisena listana
  - Tuotteet listattu ja
  - tuotteen ominaisuudet listattu
  - Järjestämättömänä tai järjestettynä listana

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<CATALOG>
  <PLANT>
    <COMMON>Bloodroot</COMMON>
    <BOTANICAL>Sanguinaria canadensis</BOTANICAL>
    <ZONE>4</ZONE>
    <LIGHT>Mostly Shady</LIGHT>
    <PRICE>$2.44</PRICE>
    <AVAILABILITY>031599</AVAILABILITY>
  </PLANT>

  <PLANT>
    <COMMON>Columbine</COMMON>
    <BOTANICAL>Aquilegia canadensis</BOTANICAL>
    <ZONE>3</ZONE>
    <LIGHT>Mostly Shady</LIGHT>
    <PRICE>$9.37</PRICE>
    <AVAILABILITY>030699</AVAILABILITY>
  </PLANT>

  <PLANT>
    <COMMON>Marsh Marigold</COMMON>
    <BOTANICAL>Caltha palustris</BOTANICAL>
    <ZONE>4</ZONE>
    <LIGHT>Mostly Sunny</LIGHT>
    <PRICE>$6.81</PRICE>
    <AVAILABILITY>051799</AVAILABILITY>
  </PLANT>
    ...
```

XML - Esimerkki

- PLCOpenXML

```
<FBD>
    <block width="8" height="13" instanceName="AND2_BOOL" typeName="AND2_BOOL" localId="3">
        <position x="29" y="6"/>
        <inputVariables>
            <variable formalParameter="IN1" negated="true" hidden="true">
                <connectionPointIn>
                    <relPosition x="-2" y="4"/>
                    <connection refLocalId="16">
                        <position x="27" y="10"/>
                        <position x="22" y="10"/>
                        <position x="16" y="10"/>
                        <position x="14" y="10"/>
                    </connection>
                </connectionPointIn>
            </variable>
            <variable formalParameter="IN2" hidden="true">
                <connectionPointIn>
                    <relPosition x="-2" y="8"/>
                    <connection refLocalId="17">
                        <position x="27" y="14"/>
                        <position x="15" y="14"/>
                    </connection>
                </connectionPointIn>
            </variable>
        </inputVariables>
        <inOutVariables/>
        <outputVariables>
            <variable formalParameter="OUT" hidden="true">
```



XML merkintäkielet

EXTENSIBLE MARKUP LANGUAGE

Merkintäkielet - yleisesti

- Merkintäkieli on tekstipohjainen kieli, jolla voidaan kuvata rikasta joukkoa tietoa hyvin yksinkertaisen säännöstön pohjalta
  - Merkintäkielellä voidaan annotoida (=merkitä) tekstiä siten, että merkinnät ovat eroteltavissa tekstistä
- Yleensä merkintäkielen avulla luodaan tiedostoon haluttu rakenne
  - koneet tulkitsevat tätä rakennetta ennalta määrättyjen sääntöjen avulla
- Yleisimmät merkintäkielet ovat XML:n määrittelyn mukaisia
- HTML ja JSON ovat myös merkintäkieliä
  - HTML-kielellä merkataan miten tieto esitetään web-sivulla; se ei siis kuvaa itse tietosisällön rakennetta.
  - JSON (JavaScript Object Notation) soveltuu tietosisällön kuvaamiseen selaimissa ja web-ympäristössä (Huom: JSON ei ole XML-kieli)

Extensible Markup Language XML

- Extensible Markup Language (XML) on World Wide Web Consortiumin (W3C) suositus elektronisen tiedon esitysmuodoksi (http://www.w3.org/TR/xml/)
  - XML määrittelee ainoastaan tavan esittää tietoa (kieliopin, syntaksin)
  - se ei ota kantaa itse esitettävään tietoon eikä myöskään määrittele primitiivijoukkoa (merkkien nimiä), jolla tietoa pitäisi kuvata
- XML on metakieli
  - XML:lla voidaan määritellä uusia merkintäkieliä, joilla vasta kuvataan itse tietoa ja sen rakennetta
  - puhuttaessa XML-dokumentista yms. tarkoitetaan dokumenttia, joka on XML-määritysten mukainen eli oikeamuotoinen (XML-määritysten mukaisella kielellä merkattu)

W3C - XML:n tavoitteet

- Mahdollisuus käyttää dokumentteja Internetissä
- Rakenne on formaali ja tiivis
- Laaja ohjelmistotuki
- XML:ää käyttävien sovellusten ohjelmointi on helppoa
- Dokumentit ovat lukukelpoisia myös ihmisille
- Dokumentin suunnittelu on nopeaa
- Dokumenttien laadinta on helppoa
- Vapaavalintaisten ominaisuuksien määrä on minimoitu
- XML:n ei tarvitse olla ytimekästä
- Yhteensopivuus SGML:n kanssa

Kielten kompleksisuus on sopiva

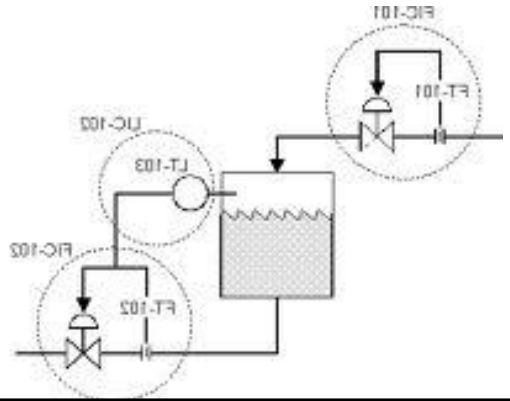
[Origin and Goals: http://www.w3.org/TR/REC-xml/]

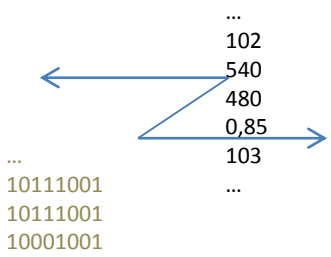
XML:N KÄYTTÖSOVELLUKSET

XML Automaatiossa - Binääridata vs. XML

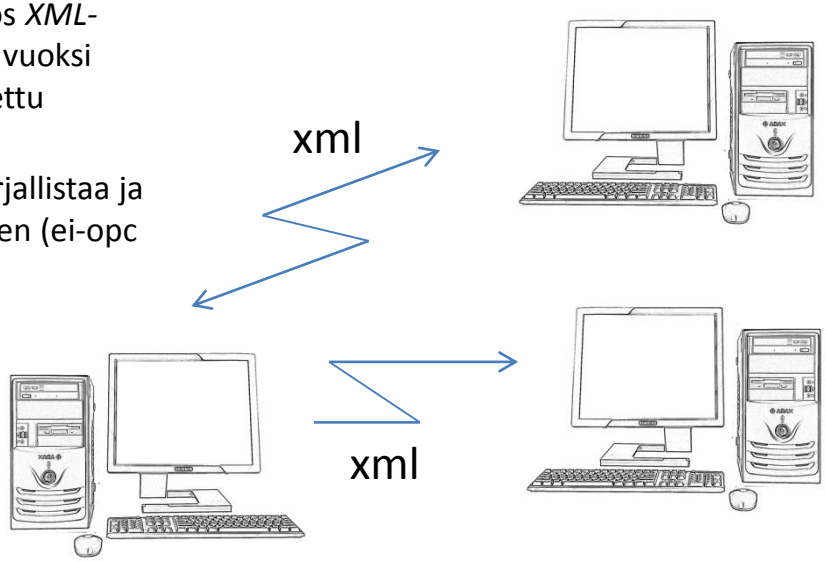
- Laitteiden mittaustietojen lukemisessa ja niiden suorassa ohjauksessa käytetään perinteisesti ns. *binääristä datasiirtoa*
- Tämä on mahdollista sillä ohjaussovellukset on ohjelmoitu juuri ko. tarkoitukseen ja ne tietävät miten datasekvenssit on koodattu binäärimuotoon ja missä järjestyksessä ne lähetetään
- Uusien älykkäiden mitta- ja toimilaitteiden dataa on kuitenkin tarve lukea useaan eri tietojärjestelmään, joiden sovellukset eivät voi tietää miten binääridata on koodattu
- *OPC UA* on uusi automaation tiedonsiirron rajapintastandardi. Vanha OPC XMLDA mahdollistaa laitetietojen siirron myös *XML-muodossa*, mutta erityisesti reaaliaikaisuus-vaatimuksen vuoksi *OPC UA* tiedonsiirto serverin ja client:in välillä on toteutettu binäärisenä.
- Client sovelluksessa mittaustieto voidaan tarvittaessa sarjallistaa ja lähettää *XML- tai JSON- muodossa*, jolloin tieto on muiden (ei-opc ua) sovellusten luettavissa ja ”ymmärrettävissä”

```
<ProcessData>
        <TankData>
                <Pressure unit="kPa">540</Pressure>
                <Temperature unit="K">480</Temperature>
                <Level>85%</Level>
        </TankData>
        ...
</ProcessData>
```

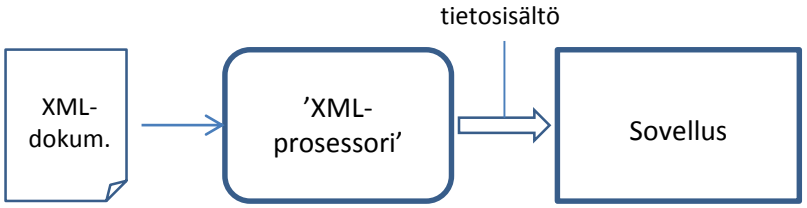




OPC UA binary



XML:n käyttösovellukset



- XML:n käytön perusperiaate:
- XML-dokumenttien käyttö on yleensä kaksivaiheista:
  1. ’XML-prosessori’ lukee XML-dokumentin jostakin lähteestä
  2. sovellus saa ’XML-prosessorilta’ XML-dokumentin sisällön käyttöönsä

- Näin on pyritty helpottamaan sovellusten laatijoiden työtaakkaa ja samalla pitämään huoli, että XML-dokumentteja käsitellään samalla tavalla sovelluksesta toiseen
  - melkein kaikkiin ohjelmointikieliin löytyy yksi tai useampia rajapintoja XML-dokumenttien käsittelyyn
  - näiden rajapintojen takana on yleensä *XML-parseri* (tai -jäsennin), joka lukee ja muokkaa XML-dokumentin johonkin ohjelmointikielelle sopivaan muotoon

- XML ei ota kantaa, mitä sovellus dokumentilla tekee

XML-DOKUMENTIN RAKENNE

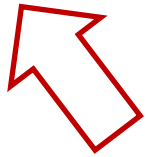
XML:n rakenneosat

- XML-dokumentti voi sisältää seuraavia rakenteita:
  - elementit – muodostavat dokumentin rungon ja rakenteen
  - attribuutit – antavat lisämääreitä elementeille
  - tekstisisältö/leipäteksti – elementtien sisällä olevaa tekstiä
  - kommentit – kommentteja ihmisille tai koneilta väliaikaisesti piilotettuja osioita
  - käsittelyohjeet – ohjeita dokumentin koneelliseen käsittelyyn
  - entiteetit – erikoismerkkien esittämiseen ja dokumenttien liittämiseen toisiinsa

XML 1.0 Spec: http://www.w3.org/TR/xml/

Rakenneosat: Elementit

- Dokumentti koostuu elementeistä (element)
- Jokaisella elementillä on nimi, joka kirjoitetaan kulmasulkujen (< ja >) sisään
- Elementillä on yleensä alkuosa <element> ja loppuosa </element>, joiden väliin kirjoitetaan elementin tekstisisältö/leipäteksti



<Luento> *XML-dokumentin rakenne* </Luento>

<Pvm> *2017-01-19* </Pvm>

- Yksiosaisia elementtejä kutsutaan tyhjiksi elementeiksi (empty element)
  - Merkitään lisäämällä kauttaviiva elementin nimen perään tai
  - Alkuosa ja loppuosa välittömästi peräkkäin

<Harjoitukset/>

< Harjoitukset ></Harjoitukset >

Huom: Elementtien sijaan puhutaan myös tageista (tag), joilla tarkoittaa välillä elementtiä, välillä elementin alku- tai loppuosaa (start/end tag)!

Huom: Tyhjä elementti ei ole merkityksetön. Se ei siis ole sama kuin ei lainkaan ko. elementtiä!

Rakenneosat: Elementtien nimet

- Elementtien nimille on tiettyjä sääntöjä
  - Nimi voi alkaa ainoastaan kirjaimella tai alaviivalla
  - Nimessä on sallittu käyttää kirjaimia, numeroita, ala- ja tavuviivoja ja pisteitä (huom: ei välilyöntejä (white space))
  - Nimen kirjainten koolla on merkitystä (case sensitive)

HUOM: Vaikka XML on *unicodea*, niin kannattaa välttää skandinaavisia merkkejä ja muita erikoiskirjaimia, sillä kaikki ohjelmat eivät välttämättä osaa tulkita niitä oikein!

Dokumentin rakenne: Elementtien sisäkkäisyys

- Elementit voivat sisältää toisia elementtejä ja/tai leipätekstiä
- Dokumentin ulointa elementtiä kutsutaan juurielementiksi (*root element*)
- Sisempiä elementtejä kutsutaan lapsielementeiksi (*child element*) ja ulompaa isäelementiksi (*parent element*)
- Sisäkkäisyydestä muodostuu dokumentin rakenne
  - sisäkkäisyys ei voi mennä ristiin

- Esimerkki:

< Moduuli >…< Luennot >…</Luennot >…
< Harjoitukset > …</Harjoitukset > … < /Moduuli>

Rakenneosat: Tekstisisältö

- Elementit voivat sisältää leipätekstiä sisällään
- lapsielementit ja leipäteksti/tekstisisältö voivat vaihdella vapaasti:

<Moduuli> *XML-moduuli* <Luento/> *Harjoitukset* </Moduuli>

- Teknisesti, jokainen erillinen leipätekstisirpale sijoitetaan omaan leipätekstisolmuun XML-puussa
- yllä olevassa esimerkissä on kaikkiaan neljä solmua:
  - elementti ”Moduuli"
  - teksti ” *XML-moduuli* "
  - elementti ”Luento"
  - teksti ” *Harjoitukset* "

Selkeämmin esitettynä sisennyksiä käyttäen

```
<Moduuli>
    XML-moduuli
    <Luento/>
    Harjoitukset
</Moduuli>
```

Rakenneosat: CDATA-alue

- Leipäteksti voi olla lähes mitä tahansa tekstiä ja erikoismerkkejä (kirjaimia, numeroita, merkkejä)
- Leipätekstissä ei voi kuitenkaan käyttää XML:n syntaksissa varattuja merkkejä:
  - **<**, **>** ja **&** merkkejä EI SAA KÄYTTÄÄ
  - näitä varten on olemassa korvausmerkinnät (<, > ja &), *(& = et-merkki; ampersand)*
  - Entiteettejä &entiteetti; käytetään lyhennys- tai korvausmerkintänä tietylle merkille tai merkkijonolle

HUOM: Mikä tahansa merkki voidaan ilmoittaa entiteettinä muodossa &#x*2014*; missä luku on merkin *Unicode* heksadesimaalimuodossa!

- Silloin kuin entiteettien käyttö ei ole mahdollista (ks. HUOM), voidaan käyttää CDATA-aluetta (CDATA section), joka voi sisältää mitä tahansa merkkejä
  - alue aloitetaan merkeillä **<![CDATA[** ja päätetään merkkeihin **]]>**
  - XML-prosessori ei jäsennä/parsi CDATA-aluetta (Unparsed Character Data)

HUOM: XML-muotoinen koodi, jota ei haluta XML-prosessorin parsittavaksi, täytyy sisällyttää CDATA-alueeseen (XML-data, joka ei kuulu siirrettävän XML-dokumentin rakenteeseen vaan on sen datakuormaa)!

Rakenneosat: Tyhjät merkit

- Tyhjiksi merkeiksi (white spaces) luetaan välilyönnit, tabulaattorit, rivinvaihdot yms.
- XML-dokumentti yleensä tulostetaan siistiin muotoon jokainen elementti omalla rivillään
  - tällöin elementtien väliin voi syntyä tyhjiä merkkejä sisältäviä leipätekstisirpaleita
  - XML-käsittelijän voi ohjata hävittämään pelkästään tyhjistä merkeistä koostuvat leipätekstisirpaleet

HUOM: Tyhjät merkit, erityisesti niiden ilmestyminen ja katoaminen, aiheuttavat useasti ongelmia dokumenttien käsittelyssä (XSL muunnokset)!

Rakenneosat: Attribuutit

- Elementteihin voidaan lisätä tietoa (tekstisisältöä) myös attribuuteilla
- Attribuutti on *avain-arvopari*
  - attribuutin nimeä vastaa sille annettu arvo, joka esitetään lainaus- tai heittomerkeissä
  - arvon sisältö voi olla mitä tahansa tekstiä ja entiteettejä
  - jokin arvo on välttämätön; vähintäänkin tyhjä merkkijono: ””
  - yhdessä elementissä voi olla useita attribuutteja
  - nimen tulee olla yksikäsitteinen

```
<empty-element attr1="value" attr2=”value2”/>

<Kurssi koodi=” ELEC-C1220" jaksot=””> Automaatio II </Kurssi>
```

Dokumentin rakenne: Attribuutti vs. lapsielementti

- Attribuutti voidaan aina korvata lapsielementillä
  - attribuutin nimestä saadaan elementin nimi
  - attribuutin arvosta saadaan elementin tekstisisältö
- Valinta attribuutin ja lapsielementin välillä on dokumentin rakennetta koskeva päätös (usein makukysymys)
  - Elementin tietosisältöön liittyvä metatieto (=tietoa tiedosta) kannattaa esittää attribuutin arvona
  - Lapsielementin käyttö mahdollistaa myöhemmän laajentamisen ja sallii useita arvoja
  - Attribuuttiarvon esitysmuoto voidaan määritellä tarkasti (myös tyhjät merkit)
  - Attribuuttien käsittely on yleensä helpompi ohjelmoijille
  - XML-käsittelijä standardoi tyhjät merkit attribuuttien arvoissa (säilyttää), muttei elementeissä

```
<kurssi koodi=”ELEC-C1220" jaksot=”III-IV”> Automaatio 2 </kurssi>
```

vs.

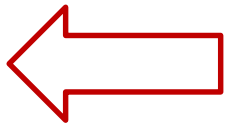
```
<kurssi>
  <koodi> ELEC-C1220 </koodi>
  <jaksot> III-IV <jaksot/>
  <nimi> Automaatio 2 </nimi>
</kurssi>
```

```
<ProcessData>
    <TankData>
        <Pressure unit="kPa">540</Pressure>
        <Temperature unit="K">480</Temperature>
        <Level>85%</Level>
    </TankData>
    ...
</ProcessData>
```

Rakenneosat: Alkumäärittely, käsittelyohjeet ja kommentit

- XML-dokumentin alkumäärittely ei ole prosessointiohje vaikka se on samaa muotoa

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



- Käsittelyohjeet vastaavat kommentteja koneelliselle käsittelijälle

<?prosessori attr1=”???”?>

- Niillä voidaan antaa ohjeita myös erilaisille koneelliselle käsittelijöille

<?xml-stylesheet type="*text/xsl*” ref=”*production_line_to_aml.xsl*”?>

- Kommentteihin voi kirjoittaa selventävää tietoa dokumentin lukijalle (ihmiselle)

<!-- *Dokumentti on päivitetty 2018-01-11* -->

HUOM: Kommentti on näppärä tapa sulkea osa dokumentista konekäsittelyn ulkopuolelle!

%%Hyvämuotoinen XML

- XML-dokumentti on hyvämuotoista (well-formed), kun
  - dokumentti alkaa XML-määrittelyllä
  - dokumentissa on vain yksi juurielementti, jonka lapsia kaikki muut elementit ovat
  - Isä-elementeillä on alku- ja loppuosa
  - tyhjät elementit on merkitty loppukauttaviivalla
  - elementit ovat sisäkkäin, eivät ristikkäin
  - dokumentissa ei käytetä merkkejä **<** ja **&** kuin elementtien ja entiteettien alussa
- Jos dokumentti ei täyty em. vaatimuksia, se ei ole virallisesti XML:a

Well-formed validator by w3c: http://validator.w3.org/#validate_by_input

Validator enabling also well-formedness checking :
https://www.freeformatter.com/xml-validator-xsd.html

Lähdeviitteet

- Kirjoja:
  - (XML:n perusteet ja Xpath: luvut 1, 3 ja 4 kirjasta) Goldberg, K.,H.2009. XML-Visual QuickStart Guide. 2.e. Peachpit Press XML. : Introduction, Ch1, Ch2, Ch3, Ch4 ja Ch9-Ch15.
  - (XML:n perusteet ja Xpath: luvut 1, 2 ja 7 ja Appendix B: XPath Functions kirjasta) E-book: Beginning XML, (5th ed.) by Fawcett, J., et al., John Wiley & Sons, Inc. Ch 1-3, Ch5, Ch7, Ch8.
- Muita lähteitä:
  - Tutoriaalit: http://www.w3schools.com/
  - XPATH esimerkkejä: http://www.w3schools.com/xpath/xpath_examples.asp
  - Well-formed validator by w3c: http://validator.w3.org/#validate_by_input
  - W3C:n Määrittelyt:
    - http://www.w3.org/XML/Core/
    - http://www.w3.org/Style/XSL/;
    - http://www.w3.org/XML/Schema#dev
- Tools
  - Online tool: xpath, xslt, schema validate: http://www.xpathtester.com
  - Online tool: xpath, xslt, schema validate, reg-exp: http://www.freeformatter.com
  - Command line tool: Xmlstarlet command line XML toolkit:
    - Download for windows: http://xmlstar.sourceforge.net/download.php
  - XML editors
    - EditX Lite free version: http://www.freexmleditorsite.com/download.html
    - MindFusion XML Viewer on Ilmainen editori. Sopii ainakin XSLT-muunnoksien harjoitteluun.