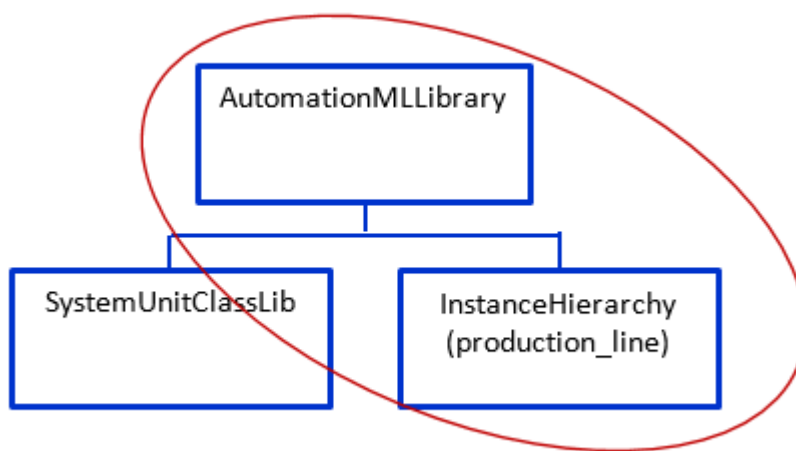


## Tehtävä 3: XSLT Muunnokset

### 1. XML-AML -muunnos: AutomationMLLibrary ja Instanssihierarkia

AutomationML:n (AML) mukainen laitteistokuvauksen XML-dokumentti (AutomationMLLibrary) koostuu kahdesta osasta: laitteiston rakennehierarkiasta (InstanceHierarchy) ja tyyppihierarkiasta (SystemUnitClassLib). AML mallin XML-dokumentin juurielementillä on siis nämä kaksi lapsielementtiä kuvan 1 mukaisesti. (Ks. tarkemmin *XML\_AutomationML\_extra.pdf*)



Kuva 1. AutomationMLLibrary dokumentti koostuu tyypillisesti kahdesta osasta rakennehierarkiasta (InstanceHierarchy) ja tyyppihierarkiasta (SystemUnitClassLib). Tässä tehtävässä keskitytään rakennehierarkian generointiin (ympäröity osa).

Tässä tehtävässä tehdään/täydennetään XSL-muunnosdokumentti (tyylisivu *AL2AML\_transform.xsl*), joka muuntaa tehtävässä 1 toteutetun *assembly\_line\_base.xml* -dokumentin (AL.xml) AutomationML-määrittelyjen mukaiseksi xml-dokumentiksi. Pääpaino on tässä tehtävässä rakennehierarkian (InstanceHierarchy) muodostamisessa. Muunnoksen tuloksen tulisi muistuttaa jossain määrin annettua esimerkkidokumenttia *AML\_result\_example.xml*, jonka katkelma on esitetty kuvassa 2.

### 2. Ohjeet rakennehierarkian generointiin:

Tulosdokumentin juurielementin tulee olla "AutomationMLLibrary" ja sen ainoan lapsielementin nimen tulee olla "InstanceHierarchy". "InstanceHierarchy"-elementillä on lapsinaan ja lapsenlapsinaan vain "InternalElement"-elementtejä ja sen nimiattribuutti (Name) saa tehtävässä arvon "assembly\_line". AML rakennehierarkiassa jokainen objekti/osa esitetään "InternalElement"-nimisenä elementtinä ja kokonaisuus-osa suhde esitetään sisäkkäisinä "InternalElement"-elementteinä. Jokaisen "InternalElement"-elementin attribuutit ovat: "Name", "ID" ja "RefBaseSystemUnitPath".

Instanssihierarkia muodostetaan lähdedokumentin *assembly\_line\_base.xml* elementeistä. Elementtejä "*description*" ja "*name*" ei kuitenkaan kuulu lisätä hierarkiaan, koska ne eivät ole todellisia laitteiston osia.

### Vaihe 1 InternalElement hierarkian generointi

Tehtävän ensimmäisessä vaiheessa kannattaa kirjoittaa tarvittavat sapluunat/templatet lähtödokumentin elementtien käsittelyyn. Kukin template generoi *InternalElementin* ja kutsuu tarvittavia ali-templateja (*xsl:apply-templates*). Tässä vaiheessa tyylisivun tulee siis generoida vain *InternalElement* hierarkia (kuva 2), mutta ei vielä attribuutteja.

### Vaihe 2 Attribuuttien arvojen generointi

Toisessa vaiheessa kehitetään tyylisivua niin, että kunkin *InternalElementin* generoineeseen templateen lisätään ko. elementissä tarvittavien attribuuttien generointi. Instanssihierarkian elementtien attribuuttien arvot generoidaan muunnosdokumentilla seuraavia sääntöjä noudattaen:

1. Kohdedokumentin "*InternalElement*":n "*Name*"-attribuutti:
  - Kohdedokumentin "*InternalElement*":n "*Name*"-attribuutti saa arvokseen ko. rakenneosan nimen ("*name*"-elementin arvon) lähdedokumentissa (AL.xml), jos se on siinä esitetty.
  - Esim. `<InternalElement Name="stack_magazine_module">`
  - Muussa tapauksessa "*Name*"-attribuutin arvo muodostetaan AL.xml:n ko. rakenneosan elementtimestä (*local-name()*) täydennettynä elementin paikkamerolla (*position()*), eli kuinka mones tämän niminen isän lapsi-elementti on kyseessä (Ks. JÄRJESTYS\*). Apuna voi käyttää esim. XPath:n *local-name(self::node())* ja *position()*-funktioita.
  - Esim. `<InternalElement Name="sensor_1">`
2. "*ID*"-attribuutti:
  - Kohteen ID-attribuutin arvoksi tulee rakenneosan ID-attribuutin arvo lähdedokumentissa, jos se on objektille olemassa.
  - Muussa tapauksessa sille luodaan kiinteä arvo "NA000" tehtävän yksinkertaistamiseksi (esim. 'component' elementit)
3. Attribuutti "*RefBaseSystemUnitPath*":
  - Tämä attribuutti on viittaus (IDREF-tyyppiä) luokkakirjaston yhteen laiteluokkaan, johon tämän rakenneosan/laitteen katsotaan kuuluvan.
  - Attribuutin arvo on polkumäärittely, joka koostuu kahdesta osasta: "*ALineSystemUnitClassLib*" on luokkakirjaston nimi ja kauttaviivalla (/) erotettu loppuosa on laiteluokan lyhytnimi tässä kuvitteellisessa AL-kirjastossa.
  - Laiteluokan lyhytnimen generointi:
    - Jos rakenneosalla on *type*-attribuutti, muodostetaan laiteluokan nimi yhdistämällä *type*-attribuutin arvo vastaavan elementin nimen kanssa.
    - Muussa tapauksessa käytetään pelkkää elementin nimeä.
  - Esimerkit:
    - `RefBaseSystemUnitPath="ALineSystemUnitClassLib/station"`
    - `RefBaseSystemUnitPath="ALineSystemUnitClassLib/OPTICALsensor"`

```

<?xml version="1.0" encoding="utf-8"?>
<!-- Malliesimerkki AL2AML-muunnoksen tulostiedostosta: aml_result_example.xml -->
<AutomationMLLibrary>
  <InstanceHierarchy Name="assembly_line">
    <InternalElement Name="distribution_station" ID="ST001" RefBaseSystemUnitPath="ALineSystemUnitClassLib/station">
      <InternalElement Name="stack_magazine_module" ID="M001" RefBaseSystemUnitPath="ALineSystemUnitClassLib/module">
        <InternalElement Name="double_acting_cylinder" ID="A001" RefBaseSystemUnitPath="ALineSystemUnitClassLib/LINEAR_MOVEactuator"/>
        <InternalElement Name="sensor_1" ID="S001" RefBaseSystemUnitPath="ALineSystemUnitClassLib/OPTICALsensor"/>
        <InternalElement Name="sensor_2" ID="S002" RefBaseSystemUnitPath="ALineSystemUnitClassLib/INDUCTIVEsensor"/>
        <InternalElement Name="sensor_3" ID="S003" RefBaseSystemUnitPath="ALineSystemUnitClassLib/INDUCTIVEsensor"/>
        <InternalElement Name="magazine" ID="NA000" RefBaseSystemUnitPath="ALineSystemUnitClassLib/PASSIVEcomponent"/>
      </InternalElement>
    <InternalElement Name="changer_module" ID="M002" RefBaseSystemUnitPath="ALineSystemUnitClassLib/module">
      <InternalElement Name="rotary_drive" ID="A002" RefBaseSystemUnitPath="ALineSystemUnitClassLib/ROTARY_MOVEactuator"/>
      <InternalElement Name="suction_cup" ID="A003" RefBaseSystemUnitPath="ALineSystemUnitClassLib/PNEUMATIC_GRIPactuator"/>
      <InternalElement Name="sensor_1" ID="S004" RefBaseSystemUnitPath="ALineSystemUnitClassLib/MECHANICALsensor"/>
      <InternalElement Name="sensor_2" ID="S005" RefBaseSystemUnitPath="ALineSystemUnitClassLib/MECHANICALsensor"/>
    </InternalElement>
  </InternalElement>
</InstanceHierarchy>
</AutomationMLLibrary>

```

Kuva 2. katkelma *aml\_result\_example.xml* rakennehierarkiasta. (Huomaa, että juurielementti on AutomationMLLibrary)

### 3. Vihjeitä:

1. Halutun AML rakenteen hahmottaneen parhaiten tutkimalla annettua esimerkkiedostoa *aml\_result\_example.xml* tai kuvaa 2.
2. Huomaa xsl-tiedoston alussa juurielementin lapsi-elementti `<xsl:output method="xml" encoding="utf-8" indent="yes" media-type="text/xml"/>`. Method on xml, koska tuloksena generoidaan xml-dokumentti ja indent on 'yes', jotta muunnos generoi sisennetyn (ainakin rivinvaihdot) xml-dokumentin.
3. Tärkeitä xsl-elementtejä tehtävän kannalta ovat (toteutuksestasi riippuen) esim. `xsl:template`, `xsl:element`, `xsl:attribute`, `xsl:value-of`, `xsl:text` (`xsl:for-each`, `xsl:sort`)
4. Katso myös vinkkejä attribuuttien arvojen määrittämisestä yhdistämällä useita arvoja merkkijonoksi teoriaosan luvusta 8. *Muita hyödyllisiä xsl:elementtejä ja vinkkejä*
5. Täydennettävän tyylisivun *AL2AML\_transform\_pohja.xsl* alussa on määritelty pari muuttujaa. Vinkkejä siitä kuinka niitä voi hyödyntään löytyy teoriaosan luvusta 8. *Muita hyödyllisiä xsl:elementtejä ja vinkkejä*

\*JÄRJESTYS: Tässä numeroidaan siis isä-elementin lapsia eli eri isillä voi olla samannimisiä lapsia. Esim. 'stack\_magazine\_module':lla on lapset 'sensor\_1', 'sensor\_2' ja 'sensor\_3' ja 'changer\_module':lla on lapset 'sensor\_1' ja 'sensor\_2' (nimet eivät siis ole yksikäsitteisiä)

### 4. XSLT muunnoksen ajaminen kahdella eri tavalla

#### Muunnos Notepad++ ohjelmalla

Avaa *assembly\_line\_base.xml* tiedosto Notepad++ editorilla (Jos se on asennettuna). Valitse menusta *Plugins/XML tools/xsl transform*, jolloin 'XML Transformation Settings' dialogi aukeaa. Navigoi (...) ja avaa tiedosto *AL2AML\_transform.xsl* ja paina 'Transform' painiketta. Muunnoksen tuloksena generoidun kooditekstin tulisi aueta uuteen Notepad++:n välilehteen.

Tallenna lopullisen muunnoksen tulos tehtäväkansioon xml-muotoiseksi tiedostoksi *aml\_result\_notepad.xml*.

Tekstieditori: Notepad++ with XML-plugin

Lähtötiedostot:

- Kohde XML: *Assembly\_line\_base.xml* (= tehtävässä 1 kirjoitettu perusmalli)
- XSL: *AL2AML\_transform.xml* (tässä tehtävässä kirjoitettu tiedosto)

Muunnos online XSLT ohjelmalla

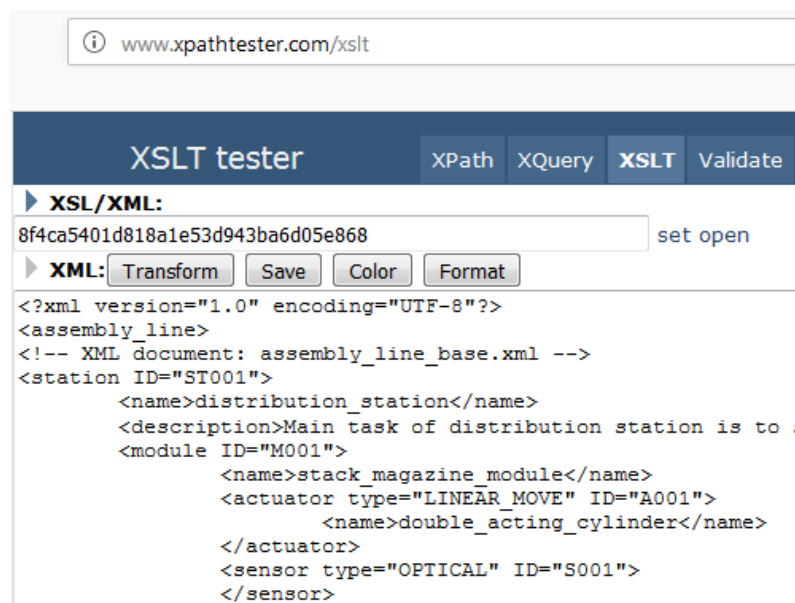
<http://www.xpathtester.com/xslt>

Paina XSL kentän perässä olevaa 'set'-painiketta ja kopioi *.xsl* tiedoston sisältö 'Content'-kenttään ja paina 'Save'. Tämän jälkeen XML lähtödokumentti kopioidaan XML-kenttään, jonka jälkeen voidaan ajaa XSLT muunnos painamalla 'Transform' painiketta. Muunnoksen tuloksena generoidu TEXT/XML kooditekstin tulisi aueta uuteen selaimen välilehteen. Kopioi lopullisen muunnoksen tuloskoodi ja tallenna se tekstitiedostoksi *aml\_result\_online.xml*.

On-line XSLT tester: <http://www.xpathtester.com/xslt>

Lähtötiedostot:

- Kohde XML: *Assembly\_line\_base.xml* (= tehtävässä 1 kirjoitettu perusmalli)
- XSL: *AL2AML\_transform.xml* (tässä tehtävässä kirjoitettu tiedosto)



Kuva 3. XSLT testerin selainnäkömä

