



VAASAN AMMATTIKORKEAKOULU  
VASA YRKESHÖGSKOLA  
UNIVERSITY OF APPLIED SCIENCES

Antti Helinko

# TYÖKALU MITTAKUVIEN AUTO- MAATTISEEN PÄIVITTÄMISEEN

Tekniikka ja liikenne  
2012

## **ALKUSANAT**

Tämä opinnäytetyö on osa Vaasan ammattikorkeakoulun tietotekniikan koulutusohjelmaa. Työn ajankohta sijoittuu vuoteen 2012. Työn ohjaajana Vaasan ammattikorkeakoulun puolelta oli yliopettaja Pirjo Prosi. Ohjaajana ABB Oy Motors and Generatorsin puolelta toimi suunnittelupäällikkö Sinikka Sauna-aho.

Haluan kiittää suuresti kumpaakin työn ohjaajaa. Lisäksi haluan kiittää Motors and Generatorsin markkinointiviestintäpäällikö Eila Nymania, jolta sain aiheen tähän työhön. Erityiskiitokset kuuluvat ABB Libraryn ylläpitäjille Marek Hacusille ja Marcin Pasteczkaalle. Ilman heidän apua sovelluksen toteuttaminen olisi ollut mahdotonta.

Vaasassa 24.12.2012

Antti Helinko





# SISÄLLYS

## TIIVISTELMÄ

## ABSTRACT

1	JOHDANTO .....	4
1.1	Työn tavoitteet .....	4
1.2	ABB Oy .....	5
2	TEAMCENTER.....	7
2.1	Teamcenterin ominaisuudet.....	7
2.2	Teamcenter ABB Motors & Generators Vaasan yksikössä.....	9
2.2.1	Nimike ja revisio.....	10
2.2.2	Moduulit ja moduulivariantit.....	14
2.2.3	Workflow ja nimikkeen revision hyväksyntä.....	15
3	ABB LIBRARY .....	17
3.1	ABB Libraryn käyttäminen .....	17
3.2	Dokumenttien lähettäminen.....	18
4	TYÖHÖN LIITTYVÄT TEKNIIKAT.....	20
4.1	JAVA.....	20
4.1.1	Historia.....	20
4.1.2	Java ohjelmointikielenä.....	21
4.2	XML.....	21
4.2.1	Historia.....	22
4.2.2	Käyttökohteet .....	23
4.2.3	SAX.....	24
4.3	FTP .....	24
4.3.1	Historia.....	24
4.3.2	FTP käytännössä.....	25
5	SOVELLUKSEN MÄÄRITTELY JA SUUNNITELU.....	26
5.1	Vaatimusmäärittelyt .....	26
5.1.1	Sovellukselta vaaditut ominaisuudet .....	27
5.2	Laitteisto- ja ympäristövaatimukset .....	27
5.3	Luokat ja metodit .....	28

5.3.1	Luokkakaavio .....	28
5.3.2	Main .....	29
5.3.3	EPoutputFolderHandler .....	29
5.3.4	LibraryData .....	29
5.3.5	Logger .....	30
5.3.6	ReadEPXml .....	30
5.3.7	WriteLibraryXML .....	30
5.3.8	FTPConnect.....	31
6	SOVELLUKSEN TOTEUTUS .....	32
6.1	Yleistä toteutuksesta.....	32
6.1.1	Sovelluksen toiminnot.....	33
6.2	Eri osioiden toteutus.....	34
6.2.1	Sovelluksen käynnistys .....	36
6.2.2	Exportpollerin kirjoittaman hakemiston lukeminen .....	37
6.2.3	Excel-tiedoston lukeminen .....	38
6.2.4	Uusien nimikkeiden etsintä .....	39
6.2.5	Xml-tiedoston luominen ja tiedostojen lähetys .....	40
7	JOHTOPÄÄTÖKSET .....	41
7.1	Tavoitteiden saavuttaminen .....	41
7.2	Jatkotoimet.....	42
	LÄHTEET .....	43

**KUVIO- JA TAULUKKOLUETTELO**

<b>Kuvio 1.</b>	Nimike ja sen revisio <i>Teamcenterin</i> käyttöliittymässä	s. 11
<b>Kuvio 2.</b>	<i>Main</i> -välilehden yläosa.	s. 13
<b>Kuvio 3.</b>	<i>Main</i> -välilehden alaosa.	s. 14
<b>Kuvio 4.</b>	Moduulivariantin osaluettelo PSE:ssä.	s. 15
<b>Kuvio 5.</b>	Julkisen ABB Libraryn etusivu.	s. 17
<b>Kuvio 6.</b>	ABB Libraryn laajennettu haku.	s. 18
<b>Kuvio 7.</b>	Yleiskatsaus dokumenttien siirrosta ABB Libraryyn.	s. 19
<b>Kuvio 8.</b>	<i>Java</i> -koodin käännösprosessi.	s. 21
<b>Kuvio 9.</b>	Kolmikerrosmallin prosessikaavio.	s. 24
<b>Kuvio 10.</b>	Luokkakaavio	s. 28
<b>Kuvio 11.</b>	Toimintatapakaavio	s. 35
<b>Kuvio 12.</b>	<i>Exportpoller</i> in output-hakemiston rakenne.	s. 37
<b>Kuvio 13.</b>	<i>Excel</i> -tiedoston lukeminen <i>HashMap</i> -muuttujaan.	s. 39
<b>Taulukko 1.</b>	Sovellukselta vaadittavat ominaisuudet.	s. 27
<b>Taulukko 2.</b>	Sovelluksen suorittamat toiminnot.	s. 34

# 1 JOHDANTO

Tämä opinnäytetyö on tehty *ABB Oy Motors & Generatorsin* Vaasan liiketoimintayksikön järjestelmäkehitysosastolle. Työn aihe syntyi eräässä palaverissa, jossa markkinointiosaston toimihenkilö näytti ohimennen tavan jolla he julkaisevat moottoreiden mittapiirustuksia ja muita dokumentteja *ABB Library*-nimiseen tiedostojen jakopalveluun. Toimintatapa vaikutti kankealta ja paljon aikaa vievältä, joten opinnäytetyöksi päätettiin tehdä työkalu, joka automatisoi mittapiirustusten julkaisuprosessin.

## 1.1 Työn tavoitteet

Vaasan yksikössä moottoreiden mittapiirustukset tallennetaan *Teamcenter*-nimiseen tuotetiedon hallintajärjestelmään sen jälkeen, kun suunnittelija on piirtänyt ne valmiiksi *CAD*-mallinnusohjelmalla. *Teamcenter*issä mittapiirustukset ovat liitetiedostoina nimikkeen revision alla olevassa *published*-kansiossa. Jokaisella revisioilla on oma *published*-kansionsa, josta ne pitää siirtää *ABB Libraryyn* aina kun mittapiirustuksen sisältävä nimike revisioidaan.

Alun perin työn tavoitteena oli kehittää työkalu, jonka avulla voitaisiin julkaista myös täysin uusia dokumentteja *ABB Libraryyn*. Mutta uusien dokumenttien julkaisun automatisointi kariutui metadatan puutteeseen *Teamcenter*issä. Päädyttiin ratkaisuun, missä kehitetään työkalu, jonka avulla voidaan päivittää jo valmiiksi *Libraryssä* olevia dokumentteja. Dokumenttien päivitys *Libraryyn* tulee tehdä silloin kun mittapiirustuksista julkaistaan uusi revisio *Teamcenter*issä.

Suunnitelmissa oli myös että työkalun avulla voitaisiin päivittää *CAD*-ohjelmalla luodut tiedostot *Libraryyn*. Näiden tiedostojen kohdalla ongelmaksi muodostuivat suunnittelijoiden erilaiset toimintatavat. Toiset suunnittelijat tallentavat tiedostot *Zip*-formaattissa ja toiset jättävät ne *dxf*-formaattiin. *Zip*-tiedostoihin pakataan myös muita tiedostoja kuin *CAD*-ohjelmalla luotuja mittapiirustuksia, joten ei voida varmistaa, että *Libraryyn* päätyisi vain sinne tarkoitetut tiedostot.



*Motors & Generatorsin* Vaasan yksikössä mittapiirustuksia julkaistaan myös muihin järjestelmiin kuin vain *Teamcenteriin*, muun muassa *Lotus Notes-* pohjaisiin tietokantoihin. *Teamcenter*-ohjelmiston toimittaja *IDEAL PLM* on kehittänyt *Exportpoller*-nimisen liitännäissovelluksen *Teamcenteriin*, jonka avulla liitetiedostot saadaan ulos järjestelmästä. *Exportpoller* kirjoittaa verkkolevyllä sijaitsevaan hakemistoon tiedot uusista julkaistuista revisioista *XML*-tiedostossa, sekä kopioi revisioon liittyvät liitetiedostot tähän hakemistoon. Tässä työssä käytetään uusien revisioiden julkaisun seuraamiseen *Exportpollerin* kirjoittamaa hakemistoa.

## 1.2 ABB Oy

*ABB* ([www.abb.com](http://www.abb.com)) on johtava sähkövoima- ja automaatioteknologiayhtymä, jonka tuotteet, järjestelmät ja palvelut parantavat teollisuus- ja energiayhtiöasiakkaiden kilpailukykyä. Vuonna 2012 *ABB:n* palveluksessa on yli 130 000 henkilöä yli 100 maassa./1/.

*ABB:n* ydinliiketoiminta rakentuu viidestä divisioonasta, jotka ovat:

- sähkövoimatuotteet.
- sähkövoimajärjestelmät.
- sähkökäytöt ja kappaletavara-automaatio.
- pienjännitetuotteet.
- prosessiautomaatio.

Sähkövoimatuotteet on jakautunut kolmeen liiketoimintayksikköön, jotka valmistavat muuntajia, suur- ja keskijännitekojeistoja, katkaisijoita, releitä, kaapeleita ja komponentteja. Ne myös tarjoavat palveluja, jotka liittyvät tuotteiden tehokkuuteen ja elinkaareen./1/.

Sähkövoimajärjestelmät-yksikkö tarjoaa voimansiirtoon ja sähkönjakeluun liittyviä järjestelmiä ja palveluja. Divisioonan neljä liiketoimintayksikköä tarjoavat sähköasemien ja – automaatiojärjestelmien lisäksi *FACTS-* ja *HVDC-*järjestelmiä, voimalaitos- ja verkostoautomaatiota sekä voiman tuotannon instrumentointi-, valvonta- ja sähköistysratkaisuja./1/.

Sähkökäytöt ja kappaletavara-automaatio-divisioona tarjoaa energia- ja tuotantotehokkuutta lisääviä tuotteita, järjestelmiä ja palveluja, kuten moottoreita, generaattoreita, tajuusmuuttajia, ohjelmoitavia logiikkoja, tehoelektroniikkaa sekä robotteja. Divisioona tarjoaa kattavan valikoiman ratkaisuja myös aurinkoenergiamarkkinoille./1/.

Pienjännitetuotteet-divisioona valmistaa pienjännitteisiä katkaisijoita, kytkimiä, ohjaus- ja valvontakojeita, asennustarvikkeita sekä kotelo- ja kaapelijärjestelmiä, jotka suojaavat ihmisiä, laitteistoja ja sähkölaitteita ylikuormitukselta. *KNX*-järjestelmät yhdistävät rakennuksen kaikki sähköiset toiminnot, kuten valaistuksen, ilmastoinnin ja valvonnan yhtenäiseksi älykkääksi verkoksi./1/.

Prosessiautomaatio-divisioona tarjoaa tuotantoprosessien energiatehokkuutta ja tuottavuutta parantavia tuotteita ja järjestelmiä niin öljy- ja kaasu, kemian- ja lääke-, metsä-, metalli- kuin meriteollisuudenkin tarpeisiin./1/

## 2 TEAMCENTER

*Teamcenter* on Siemens PLM Softwaren kehittämä Java-pohjainen ohjelmisto, jonka päätehtävä on hallinnoida tuote – ja suunnittelutietoa yrityksissä. Se on maailman laajimmin käytetty PLM-järjestelmä./2/

### 2.1 Teamcenterin ominaisuudet

*Teamcenter* on modulaarinen sekä skaalautuva järjestelmä, joten sen avulla voidaan luoda erilaisia ohjelmistokokonaisuuksia erikokoisten yritysten tarpeisiin/2/. Alla on esitelty *Teamcenterin* ominaisuuksia ja moduuleja.

***Bill of Materials Management*** on tuote- ja prosessitietojen hallintajärjestelmä. Sen avulla käyttäjät saavat oikeat ja ajantasalla olevat tuoterakennetiedot tuotteiden valmistusta varten./3/

***Community Collaboration*** on yhteistyöympäristö, jonka avulla voidaan laajentaa PLM-ominaisuuksien käyttömahdollisuuksia yrityksen kaikilla osa-alueilla, mukaanlukien yhteistyökumppanit ja tavaran toimittajat. Näin voidaan parantaa yleistä tuottavuutta./3/

***Compliance Management*** on säännösten ja direktiivien noudattamisen hallintatyökalu. Se auttaa seuraamaan erilaisten direktiivien ja säännösten noudattamista tuotteen koko valmistusprosessin ajan./3/

***Content and Document Management*** on sisällön ja dokumenttien hallintaan tarkoitettu ominaisuus. Sen avulla voidaan muodostaa rakenteellinen sisällön luonti- ja julkaisu-ympäristö, joka on integroitu PLM-ympäristöön./3/

***Engineering Process Management*** eli suunnitteluprosessin hallintaominaisuus, sisältää työkalut datan käsittelyyn ja suunnittelun työnkulun hallintaan. Näitä ominaisuuksia tarvitaan useiden CAD-järjestelmissä luotujen kokoonpano- ja suunnittelutietojen hallintaan./3/

***Enterprise Knowledge Foundation*** eli yrityksen tietämyksen keskittäminen. Keski- kittää yrityksen eri osastojen, kuten suunnittelun, tuotannon ja palveluiden tietä- myksen samaan ympäristöön./3/

***Formula, Package and Brand Management*** on tuotemerkkien, pakkausten ja kaavojen hallintaan suunniteltu ominaisuus. Sen avulla voidaan tuoda yhdenmu- kaiset tuotetiedot ja toimintamallit koko yrityksen käyttöön./3/

***Lifecycle Visualization*** eli tuotteen elinkaari visuaalisena. Antaa kuvan tuotteen koko elinkaaren ajalta ja sisältää tuotteen eri komponenttien 2D- ja 3D- visualisointiratkaisuja. Kuvat ovat katseltavissa vaikka ne olisi luotu eri sovelluk- silla./3/

***Maintenance, Repair and Overhaul.*** Ominaisuuden avulla yhdistetään yrityksen logistiikka, huolto, valmistus ja suunnittelu samaan ympäristöön. Tällä tavoin voidaan edistää tuottojen kasvua, sujuvampia palvelutoimintoja ja voimavarojen keskittämistä oikeisiin kohteisiin./3/

***Manufacturing Process Management*** eli valmistusprosessin hallinta. Ominai- suuden avulla yrityksen suunnittelu- ja valmistusosastot voivat tehdä yhteistyötä digitaalisessa ympäristössä./3/

***Mechatronics Process Management*** on mekaniikka-, sähkösuunnittelu- ja elekt- roniikkatekniikoiden kehitysympäristö. Ominaisuuden avulla voidaan keskittää kaikki tuote- ja prosessitiedot yhteen järjestelmään. /3/

***Platform Extensibility Services*** eli alustan laajennuspalvelut. Laajennuspalvelut auttavat yritystä saamaan tuottoa *Teamcenteriin* liittyvistä investoinneista nope- asti ja kustannustehokkaasti./3/

***Portfolio, Program and Project Management*** on ominaisuus, jonka avulla yrityk- sen strateginen tuotetarjonnan suunnittelu voidaan yhdistää ohjelmien ja projekti- en toiminnalliseen hallintaan./3/

**Reporting and Analytics** eli raportointi- ja analysointipalvelut. Ominaisuuden avulla saadaan hyvä näkyvyys liiketoiminnan eri prosesseihin, sekä *PLM*-ympäristössä tallennettuihin ja luotuihin tietoihin./3/

**Simulation Process Management** on simulaatioprosessien hallintaan suunniteltu ominaisuus. Sen avulla voidaan ylläpitää simulointidataa, jota tarvitaan tuotteiden suorituskyvyn ja laadun seuraamiseen./3/

**Supplier Relationship Management** on hankintaosaston yhteyksien hallintaan tarkoitettu ominaisuus. Se mahdollistaa yrityksen tuotekehityksen ja hankinnan toimimisen samassa ympäristössä./3/

**Systems Engineering and Requirements Management** on suunnittelun ja vaatimusten hallintaan kehitetty ominaisuus. Sen avulla sovitetaan yhteen tuotteiden käyttötarkoitus ja asianmukaiset vaatimukset. Tällä tavoin taataan se, että tuotteet vastaavat markkinoiden, asiakkaiden ja viranomaisten vaatimuksia./3/

## **2.2 Teamcenter ABB Motors & Generators Vaasan yksikössä**

*ABB Motors & Generators* Vaasan-yksikössä on käytetty *Teamcenteriä* tuotetiedon hallintajärjestelmänä vuodesta 2008 alkaen. Ennen *Teamcenterin* käyttöönottoa Vaasan yksikössä tuotetiedon hallintaan käytettiin *EMIS*-nimistä tuotannonohjausjärjestelmää sekä *Lotus Notes*-ohjelmiston avulla luotuja tietokantoja. Näissä tietokannoissa säilytettiin ja säilytetään edelleen erilaisia dokumentteja, kuten mittakuvia, materiaaleihin liittyviä standardeja ja sertifiointitodistuksia. *Lotus Notes*-tietokantoja käytetään, koska *EMIS*-ohjelmisto ei tukenut dokumenttien tallentamista. Yksikön tavoitteena on tulevaisuudessa saada kaikki suunnittelun dokumentit, ja niiden jakelu *Teamcenterin* alaisuuteen, silloin *Lotus Notes*-tietokannoista dokumenttien jakeluna voitaisiin luopua.

Vaasan yksikössä on tällä hetkellä käytössä *Teamcenter Engineering 2005 SR1/2007* versio. Käytössä olevassa versiossa on saatavilla kaikki tärkeimmät ominaisuudet, kuten liitántärajapinnat *CAD*- ja *ERP*-ohjelmistoja varten, visualisointimoduuli dokumenttien avaamiseen käyttöliittymässä ja työkalut osaluette-

laiden rakentamiseen. Uuden *Teamcenter*-version käyttöönottoprojekti on alkanut ja se otetaan käyttöön vuoden 2013 aikana.

Aktiivisia *Teamcenter*-käyttäjiä Vaasan yksikössä on noin 50 henkilöä. Käyttäjien ryhmä koostuu pääasiassa mekaniikka- ja sähkösuunnitelijoista sekä ylläpitäjistä. Tuotannossa käyttäjiä on vain muutamia, koska he saavat yleensä kaikki tarvitsemansa tiedot *ERP*-järjestelmästä. *Teamcenter*iä käytetään päivittäin muun muassa uusien nimikkeiden ja osaluetteloiden luomiseen, olemassa olevien nimikkeiden ja osaluetteloiden revisioimiseen, komponenttien metadatan päivittämiseen ja *CAD*-ohjelmistoilla mallinnettujen komponenttien tallentamiseen.

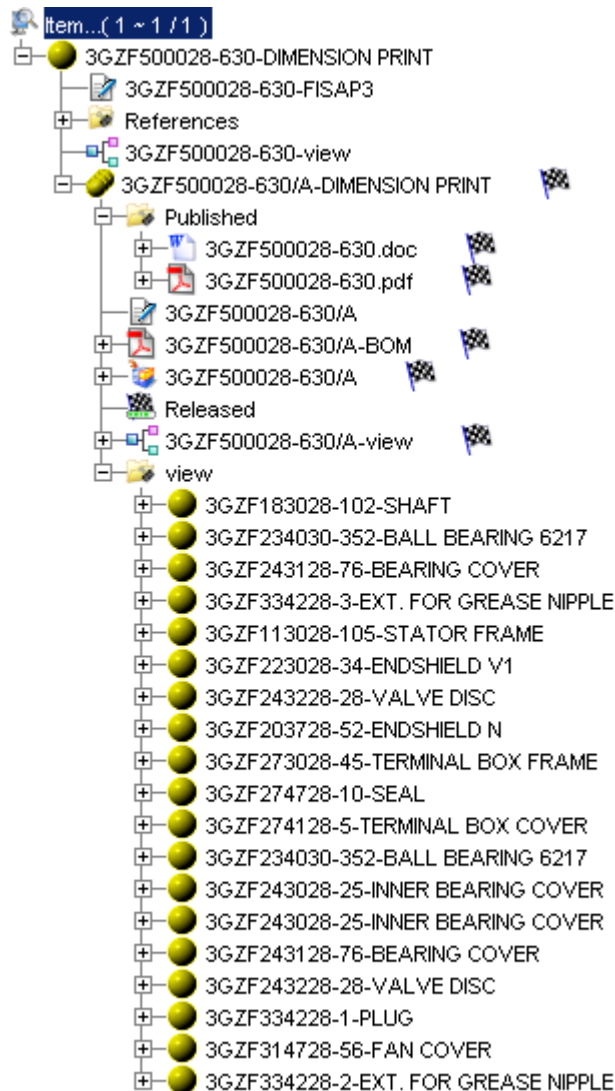
### 2.2.1 Nimike ja revisio

Nimike on *Teamcenterin* keskeisin objekti. Koko tuoterakenne voidaan esittää hierarkkisena ja monitasoisena rakenteena pelkästään toisilleen alisteisten nimikkeiden avulla. Nimike voi olla joko yksittäinen komponentti, jota ei voi jakaa pienempiin osiin (esim. ruuvi) tai se voi olla toisista nimikkeistä koostuva rakenne tai moduuli./4 s. 7/

Jokaisella nimikkeellä on sen yksilöivä koodi, *ItemID*, joka näkyy muun muassa nimikkeen perustietolomakkeella./4 s. 7/. Vaasan yksikössä on käytössä *3GZF*-koodisto nimikkeiden yksilöivään nimeämiseen. *3GZF*-koodin perusteella voidaan sanoa millaisesta nimikkeestä on kyse. Esimerkiksi tässä työssä käsiteltävät mittakuvat ovat *3GZF500*-alkuisten nimikkeiden alla.

Kunkin nimikkeen alla on vähintään yksi *item revisio*. *ABB*:llä käytetään ajallisesti peräkkäisistä versioista nimitystä revisio. Vain viimeisin hyväksytty revisio on voimassa ja ainoastaan tätä revisiota voidaan käyttää muiden nimikkeiden alla olevissa osaluetteloissa. Revisioita merkitään aakkosilla alkaen A-kirjaimesta. Eli jokaisen nimikkeen alla on aina vähintään A-revisio. Nimikkeestä tehdään uusi revisio, eli nimike revisioidaan, silloin kun nimikkeeseen tehdään rakenteellisia muutoksia. Esimerkiksi mittakuvien kohdalla muutetaan osien mittoja, tai osaluetteloissa vaihdetaan osia nimikkeen rakenteelle. Nimikkeestä voidaan tehdä myös alirevisio. Alirevisio muodostuu päärevision kirjaimesta, alaviivasta ja 3-

merkkisestä numerosarjasta. A-revision ensimmäinen alirevisio merkitään A\_001. Alirevisiointia käytetään silloin, kun nimikkeeseen tehdään muutos, joka ei vaikuta sen toiminnallisuuteen, esimerkiksi korjataan kirjoitusvirheitä päätietolomakkeella.



**Kuvio 1.** Nimike ja sen revisio *Teamcenterin* käyttöliittymässä.

Kuviossa 1 on kuvattu nimikkeen ja sen alla olevan revision näkymä *Teamcenterin* käyttöliittymässä. Ylimpänä hierarkiassa on nimike, jonka *itemID* on *3GZF500028-630*. Seuraavalla tasolla on *item revisio*, sen yksilöivä tunnus on

3GZF500028-630/A. Tällä nimikkeellä on vain yksi revisio. Sitä ei ole siis revisioitu. Revision sisältö lueteltuna ylimmästä objektista alkaen:

- *Published*-kansio, joka sisältää mittakuvat *pdf*- ja *doc*-formaattissa.
- Perustietolomake, jolle on tallennettu tarkempaa tietoa nimikkeestä.
- *Pdf*-tuloste, joka sisältää revision osaluettelon.
- *UGMaster*-objekti, jota käytetään kun mittakuvaa halutaan muokata *CAD*-ohjelmalla.
- *Released*-objekti, merkinä nimikkeen hyväksymisestä.
- *PSE*-objekti, *PSE* on *Teamcenterin* ominaisuus, jonka avulla nimikkeiden osaluettelot luodaan.
- *View*-kansio, joka sisältää nimikkeet, jotka kuuluvat tämän revision osaluetteloon.

Jokaisella revisiolla on oma perustietolomakkeensa. Lomakkeelle tallennetaan tietoja nimikkeistä, joiden avulla voidaan päätellä millaisilla rakenteilla nimikettä voidaan käyttää. Lomakkeella on kolme välilehteä:

1. *Main*-välilehti, joka sisältää kaikki tärkeimmät nimikkeen tiedot, kuten yksilöivän *ItemID*:n, nimikkeen luoja, nimikkeen hyväksyjän ja tiedot nimikkeen revisiohistoriasta.
2. *Drawing*-välilehti, joka sisältää tietoja, kuten dokumentissa käytetty kieli. Tätä välilehteä käytetään vain jos kyseessä on nimike, jonka alta löytyy dokumentti.
3. *Classification*-välilehti, joka sisältää tietoja, joiden avulla nähdään millaisten moottoreiden rakenteilla kyseistä nimikettä voidaan käyttää. Välilehdelle on tallennettu esimerkiksi moottorin runkopituus, sukupolvikoodi ja asennusasento.



Main	Drawing	Classification
Item ID:	3GZF500028-630 *	Revision: A *
Description 1:	DIMENSION PRINT *	
Description 2:		
System Classification		
Drawing Type:	13	
Classification		
Type:	N *	
Type Designation:	M2BA_280S_2_V15 *	
Material Type:		
MDF Code:	6CI *	
Distribution:	<input type="checkbox"/>	
Functional Code:	A	
	<input checked="" type="checkbox"/> Spare Part	
Responsible Department:	MLD200	
Based On:		
Work Number:		
Properties		Definitions
Material:		Material Standard:
Dimensions:		Dimension Standard:
Design Unit of Measure:	KPL *	P-Standard No:
Weight:	0,1 kg / Design Unit of Measure	Inspection Instruction:
Additional Info		
Comment:	M2BA. 280 S_2 V15, PROTECTIVE ROOF (005)	
Engineering Text:		
Production Text:		

**Kuvio 2.** *Main*-välilehden yläosa.

Kuviossa 2 nähdään *Main*-välilehden yläosa, jossa on tärkeimmät tiedot nimikkeestä. Punaisella tähdellä merkityt kentät ovat pakollisia. Jos näitä kenttiä ei ole täytetty, nimikkeen revisiota ei voida tallentaa.

Accessibility	
<input type="checkbox"/>	Obsolete
<input type="checkbox"/>	Phantom
<input type="checkbox"/>	Substitute with BOM Children
Revisoning	
Revision: C_002	
Revision Change:	<input type="text" value="DXF changed to correct one."/>
Creator:	K.Edsvik Created: 9/10/01 12:00 AM
Inspector:	<input type="text"/> Inspected:
Approver:	M.Sevola Approved: 9/10/01 12:00 PM
Revision Creator:	RiKa Revision Date Created: 7/24/12 12:14 PM
Revision Inspector:	<input type="text"/>
Revision Approver:	<input type="text" value="GuHa"/>

### Kuvio 3. Main-välilehden alaosa.

Kuviossa 3 on kuvattu *Main*-välilehden alaosa, jolta löytyy *Revisoning*-lohko. Tästä lohkosta löytyy seuraavat tiedot:

- *Revision*, kyseisen revision tunnus
- *Revision Change*, kuvaus revisioon tehdystä muutoksesta verrattuna edelliseen revisioon.
- *Creator*, A-revision luoja ja päivämäärä, jolloin A-revisio on luotu.
- *Inspector*, nimikkeen valvoja (ei käytössä Vaasan yksikössä).
- *Approver*, A-revision hyväksyjä.
- *Revision Creator*, kyseisen revision luoja ja luontipäivämäärä.
- *Revision inspector*, revision valvoja (ei käytössä Vaasan yksikössä).
- *Revision Approver*, kyseisen revision hyväksyjä.

### 2.2.2 Moduulit ja moduulivariantit

Vaasan yksikössä moottorin tuoterakenne perustuu erilaisiin moduuleihin. Moduulivariantiksi kutsutaan nimikettä, jonka osaluetteloon kuuluvat siihen moduuliin liittyvät komponentit. Yhteen moduuliin kuuluvia moduulivariantteja on sato-

ja tai tuhansia moduulista riippuen. Jokainen moduulivariantti on yksilöllinen moduulin sisällä, eli kahta samanlaista varianttia ei löydy samasta moduulista.

Ylimmällä tasolla moottorirakenteen hierarkiassa esiintyy pääasiassa vain eri moduulivariantteja, tähän ainakin pyritään uusimpien rakenteiden kohdalla. Käytetyt variantit määräytyvät sen mukaan, millaisia ominaisuuksia asiakas haluaa kyseiseen moottoriin.

3GZF101010-10/B-STATOR FRAME MODULE (view) - Latest Working - Date - "Now"					
BOM Line	Sequence No.	Revision	Quantity	Weight	Design UoM
3GZF101010-10/B-STATOR FRAME MODULE (view)		B		0.0100...	KPL
3GZF113031-3/H_001-STATOR FRAME (view) x 1	11	H_001	1		KPL
3GZF334330-4/A_002-THREADED STEEL RIVET (view) x 8	77	A_002	8	0.0100...	KPL
3GZF334330-4/A_002-THREADED STEEL RIVET (view) x 1	78	A_002	1	0.0100...	KPL
3GZF284731-3/B_009-COLLAR x 1	102	B_009	1		KPL
3GZF334030-193/A_003-WASHER x 1	103	A_003	1		KPL
9ADA337-6/A-LIFTING LUG SCREW x 1	104	A	1		KPL
3GZF194730-1/C_001-EARTHING PLATE x 1	128	C_001	1	0.0010...	KPL

**Kuvio 4.** Moduulivariantin osaluettelo PSE:ssä

*Teamcenterissä* moduulivarianttien osaluetteloiden hallintaan käytetään *PSE:tä* (*Product Structure Editor*) eli tuoterakenne-editoria. Kuviossa 4 on kuvattu staatoritunkomoduulin osaluettelo *Teamcenterin* käyttöliittymässä. *PSE:n* avulla luodaan, muokataan ja tarkastellaan osaluetteloita. Osaluettelot ovat hierarkkisia rakenteita ja ne ovat aina kytköksissä vain siihen nimikkeeseen tai moduulivarianttiin, jonka alle osaluettelo on luotu. Lisäksi osaluettelot ovat revisiokohtaisia, eli saman nimikkeen eri revisioilla voi olla erilaiset osaluettelot /4 s.53/.

### 2.2.3 Workflow ja nimikkeen revision hyväksyntä

*Teamcenterissä* käytetään *workflowta* nimikkeiden revisioiden hyväksyntään ja niiden automaattiseen julkaisuun eri järjestelmissä /4 s.9/. Vaasan yksikössä nimikkeiden revisioita julkaistaan, *Teamcenterin* lisäksi, toiminnanohjausjärjestelmään sekä erilaisiin *Lotus Notes*-tietokantoihin.

Kun suunnittelija saa valmiiksi uuden nimikkeen tai revision, se lähetetään *workflowta* käyttäen hyväksyttäväksi ennakkoon nimetylle hyväksyjälle. Jos revisio on

virheetön, hän käynnistää *workflow*-prosessin toisen vaiheen, jolloin revisio julkaistaan. Prosessin toisen vaiheen aikana käynnistyy *Exportpoller*-niminen Java-sovellus. Tämä sovellus on *PDM*-järjestelmän toimittajan luoma, se ei siis kuulu *Teamcenterin* vakiokokoonpanoon. *Exportpollerin* tehtävä on julkaista hyväksytyt revisiot *Lotus Notes*-tietokantoihin. Yksi näistä tietokannoista on nimeltään ”*Dokumenttien revisiohallinta*”. Tässä tietokannassa säilytetään dokumentteja sisältävien nimikkeiden viimeisimpiä hyväksytyjä revisioita ja itse dokumentteja, kuten mittapiirustuksia. *Exportpoller* käynnistyy vain silloin kun kyseessä on nimike, joka sisältää *Lotus Notes*-tietokantoihin julkaistavia dokumentteja. Aina *Exportpollerin* viedessä dokumentteja *Lotus Notesiin*, se kirjoittaa revision metadatasta *XML*-tiedoston ja tallentaa tiedoston verkkolevyllä sijaitsevaan hakemistoon. Täähän hakemistoon tallennetaan myös revisioon liittyvät dokumentit ja liitetiedostot.

### 3 ABB LIBRARY

*Library* on koko yhtiön laajuinen dokumenttien julkaisujärjestelmä. Se sisältää teknisiä dokumentteja *ABB*:n tuotteista, sekä dokumentteja tuotteiden markkinointia varten. *Libraryyn* kuuluu *WWW*-pohjainen työkalu, jota *ABB*:n työntekijät käyttävät päivittäin dokumenttien julkaisuun. *WWW*-sivun avulla myös asiakkaat voivat katsoa *Libraryyn* julkaistuja dokumentteja. Dokumentit voidaan julkaista myös niin, että ne näkyvät ainoastaan yhtiön sisäisen verkon käyttäjille. Erilaisia julkaisuja *Libraryssä* oli vuonna 2012 noin 290 000 kappaletta, ja määrä kasvaa koko ajan. Vuosittain sieltä ladataan tiedostoja yli 5 miljoonaa kertaa./5 s.2/

#### 3.1 ABB Libraryn käyttäminen

Julkisen suomenkielisen *Libraryn* verkko-osoite on [www.abb.fi/abblibrary/DownloadCenter/](http://www.abb.fi/abblibrary/DownloadCenter/). Julkinen verkkosivu on käytettävissä myös yhtiön verkon ulkopuolelta.

The screenshot shows the ABB Library website interface. At the top left is the ABB logo with the tagline "Power and productivity for a better world™". To the right, it says "ABB Suomessa". Below this is a navigation menu with items: Etusivu, ABB lyhyesti, Tuotteet & palvelut (highlighted), Media, ABB-uralle, Investor center, and ABB Group. Underneath, there are sub-links: Tuotteet A-Ö, Tuotteet ja järjestelmät (highlighted), Toimialaportaali, Palvelut, Yhteystiedot, and Materiaali- ja palvelutoimittajat. The main heading is "Download Center kategorialle Kaikki kategoriat (Suomi)". Below the heading are several tabs: "Pikahaku" (selected), "Laajennettu haku", "Tulokset", "Lataamasi dokumentit 0", and "Tilauksesi". To the right of these tabs are links: "→ Asetukset → Help → Anna palautetta". Below the tabs is a search bar with the text "Etsi:" and a search button labeled "Hae". To the right of the search button is a dropdown menu labeled "Vain dokumentin numero". Below the search bar is a small text: "(esim. tuotenimi, dokumenttinumero...)"

**Kuvio 5.** Julkisen ABB Libraryn etusivu.

Kuviossa 5 on kuvattuna julkisen *Library download centerin* etusivu. Etusivulla on oletuksena avoinna Pikahaku-välilehti. Välilehdellä on vain yksi vapaatekstikenttä. Pikahakua käytettäessä etsitään käyttäjän syöttämää merkkijonoa kaikkien kategorioiden dokumenttien otsikoista. Käyttämällä ”Laajennettu haku”-välilehteä

voidaan valita kategoriat, joista tuotteita haetaan, ja millaisia dokumentteja etsitään. Laajennettu haku –välilehti on kuvattu kuviossa 6.

## Download Center kategorialle Tuotteet (Suomi)

Pikahaku **Laajennettu haku** Tulokset Lataamasi dokumentit 0 Tilauksesi

**Etsi:**  
**Pääkategoria**  **Alakategoria**

**Etsi:**

esim. tuotenimi, dokumenttinumero...  
**Älä huomioi näitä sanoja**

**Hakutoiminnot:**  
**Saatavana**  **Kielivalinnat**   Näytä myös englanninkieliset dokumentit

**Dokumenttityypit**

**Tiedostotyytit**  
 Kaikki tiedostotyytit  
 PDF  EXE  
 Word  ZIP  
 PowerPoint  DXF  
 Excel  AFW

**Viimeksi päivitetty**  
 Viimeksi  päivän/-ien aikana

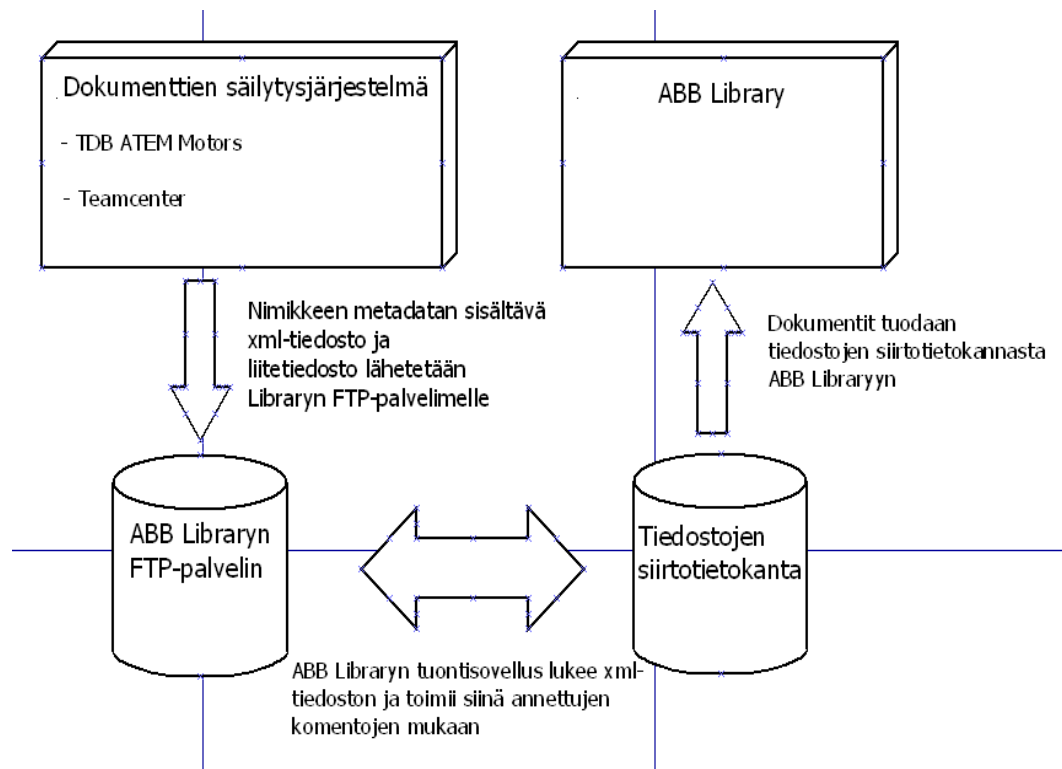
**Revisio**  
 Näytä kaikki revisiot

**Tuloksia/sivu:**

**Kuvio 6.** ABB Libraryn laajennettu haku.

### 3.2 Dokumenttien lähettäminen

Tässä työssä käytetään dokumenttien lähettämiseen *Library* tarjoamaa *XML/FTP*-rajapintaa. Rajapinnan avulla voidaan joko päivittää valmiiksi *Library*ssä olevia dokumentteja tai luoda sinne kokonaan uusi dokumentti.



**Kuvio 7.** Yleiskatsaus dokumenttien siirrosta ABB Libraryyn./5/

Kuviossa 7 on kuvattu dokumenttien siirtymisprosessi *Libraryyn*. Dokumenttien säilytysjärjestelmänä toimivat Vaasan yksikössä tällä hetkellä *TDB ATEM Motors*-niminen *Lotus Notes* -pohjainen tietokanta sekä *Teamcenter*. Tähän tietokantaan käyttäjät lisäävät jatkossakin kokonaan uudet *Libraryyn* lähetettävät dokumentit, mutta jo valmiiksi *Libraryssa* olevien dokumenttien päivitys tehdään tulevaisuudessa tässä työssä kehitetyn sovelluksen avulla. Sovellus tulee toimimaan *Teamcenterin* ja *Libraryn FTP*-palvelimen välissä. Kaikki muut kuviossa 7 esitetyt vaiheet tapahtuvat *Libraryn* palvelimilla.

## 4 TYÖHÖN LIITTYVÄT TEKNIIKAT

Tässä kappaleessa esitellään pääpiirteisesti työssä käytetyt tekniikat. Sovellus ohjelmoitiin käyttäen *Java*-ohjelmointikieltä ja tiedostot lähetetään käyttäen *Librarin* tarjoamaa *FTP/XML*-rajapintaa.

### 4.1 JAVA

*Java* on *Sun Microsystemsin* kehittämä ohjelmointialusta ja kokoelma erilaisia teknologioita. Maailmanlaajuisesti *Javaa* käytetään yli 850 miljoonassa tietokoneessa ja miljardeissa laitteissa. Sen olio-pohjainen ohjelmistoalusta on laitteistoriippumaton. Tulkattavana ohjelmointikielenä *Javaa* voidaan käyttää missä tahansa laitteessa, kunhan laitteeseen on asennettu *Java*-tulkki./6/

#### 4.1.1 Historia

*Javan* kehitys alkoi vuonna 1991, kun *Sun Microsystemsillä* työskennellyt James Gosling aloitti *Oak*-projektin yhdessä kollegoidensa kanssa. Vuonna 1995 *Javasta* julkaistiin ensimmäinen versio *Java 1.0*/6/. Samaan aikaan Internetin käytön yleistyessä 1990-luvun puolivälissä, *Sun Microsystemsin* insinöörit Patrick Naughton ja Jonathan Payne kehittivät *Oak*-ohjelmointikieltä käyttäen ensimmäisen Internet-selaimen, joka mahdollisti liikkuvien objektien ja dynaamisen koodin suorittamisen Internetsivuilla/7/. *Javan* suosio kasvoi 1990-luvun loppupuolella, kun suurimmat Internet-selaimia kehittävät yritykset ottivat sen osaksi selaintensa vakiokokoonpanoa/6/.

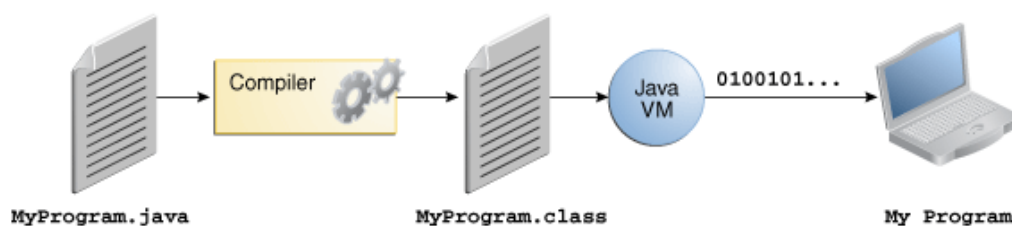
Vuosituhanen vaihteessa *Javaa* kehitettiin enemmän ohjelmistokehittäjien tarpeita ajatellen. Vuonna 1999 julkaistiin uusi *Java*-arkkitehtuuri, *Java 2 Platform*. Siihen kuuluu seuraavat ohjelmointialustat: *Standard Edition (J2SE)*, normaaleja kotikäyttöisiä tietokoneita varten, *Enterprise Edition (J2EE)* raskaskäyttöisiä servereitä varten ja *Micro Edition (J2ME)*, joka on suunniteltu erilaisten pienempien laitteiden käyttöön. Uuden arkkitehtuurin myötä *Java* alkoi levitä vauhdilla ympäri maailmaa ja sitä ryhdyttiin käyttämään myös erilaisissa sulautetuissa järjestelmissä./7/



Vuonna 2009 suuri ohjelmistoyritys *Oracle Corporation* osti *Sun Microsystemsin*. Kauppaan kuului *Sunin* lippulaivatuotteet *Java* ja *Solaris*. *Oracle* oli jo aiemmin käyttänyt *Javaa* tuotteissaan, koska piti sitä helppokäyttöisenä ja laajennuskykyisenä. Yrityskaupan myötä *Oracle* pystyy olemaan vahvemmin mukana *Javan* kehittämisessä./7/

#### 4.1.2 Java ohjelmointikielenä

*Java* eroaa muista yleisimmistä ohjelmointikielistä sen tulkattavuuden takia. Tämä tarkoittaa, ettei ohjelmakoodia käännetä käyttöjärjestelmäkohtaisesti, vaan se tulkataan alustan ymmärtämään muotoon käyttäen *Java*-asennuksen mukana tulevaa virtuaalikonetta. Tulkattavuus mahdollistaa saman ohjelmakoodin ajamisen käyttöjärjestelmästä riippumatta./8/



**Kuvio 8.** *Java*-koodin käännösprosessi./8/

Kuviossa 8 kuvataan *Java*-lähdekoodin käännösprosessi. Ensin lähdekoodi kirjoitetaan tekstitiedostoihin (*.java*). Lähdekoodit käännetään *javac*-kääntäjällä virtuaalikonetta varten (*.class*). Tämän jälkeen virtuaalikone tulkaa *class*-tiedostot prosessorin ymmärtämään muotoon./8/

*Java* on ohjelmointikielenä luokka- ja oliopohjainen. Se on sukua *C*- ja *C++*-kielille, ja muistuttaa niitä läheisesti syntaksiltaan. *Javan* kielioppi ja standardikirjasto ovat suunniteltu helposti omaksuttaviksi. Tämä edesauttaa sovelluskehittäjiä oppimaan *Java*-kielen sujuvan kirjoittamisen nopeammin./9/

## 4.2 XML

*Extensible Markup Language*, eli *XML* on yksinkertainen ja joustava tiedostomuoto. Alun perin se on suunniteltu vastaamaan haasteisiin, jotka liittyvät laaja-

mittaisten sähköisten julkaisujen käsittelyyn. Sillä on jatkuvasti kasvava ja tärkeä rooli tiedonvaihdossa Internetissä ja muissa tietotekniikan sovelluksissa./10/

#### 4.2.1 Historia

1980-luvun lopussa, ennen Internetin nousua, *Standard Generalized Markup Language*-tekniikan edut ymmärrettiin dynaamisen informaation esittämisessä digitaalisen median julkaisijoiden keskuudessa. Kokeneet *SGML*- ja *WWW*-käyttäjät ymmärsivät, että Internetin kasvuun liittyy ongelmia. Ratkaisun näihin ongelmiin tarjoaisi *SGML*. Niinpä *SGML* lisättiin *World Wide Web Consortiumin* toimintalistalle vuonna 1995. Työ tämän tekniikan parissa alkoi vuonna 1996, ja sitä johti Jon Bosak./11/

*XML*-merkintäkielten kehittämisessä oli mukana 11 hengen työryhmä sekä 150 aiheesta kiinnostunutta ihmistä. Väittelyt teknisistä asioista käytiin sähköpostin välityksellä intressiryhmän kesken. Jos syntyi erimielisyyksiä, ne ratkaistiin joko yhteisymmärryksessä tai enemmistöäänestyksellä työryhmän kesken. Michael Sperberg-McQueen kokosi vuonna 1997 suunnittelun linjaukset ja teorian niiden takana. Työryhmän tekninen johtaja James Clark nimesi merkintäkielen *XML*:ksi.

Työryhmällä oli seuraavat tavoitteet uudelle merkintäkielelle:

- käytettävyys Internetissä.
- yhteensopivuus *SGML*-tekniikan kanssa.
- yleinen vakaus.
- kaavamaisuus.
- tiiviys.
- luettavuus.
- kirjoittamisen helppous.
- valinnaisten toimintojen minimointi.

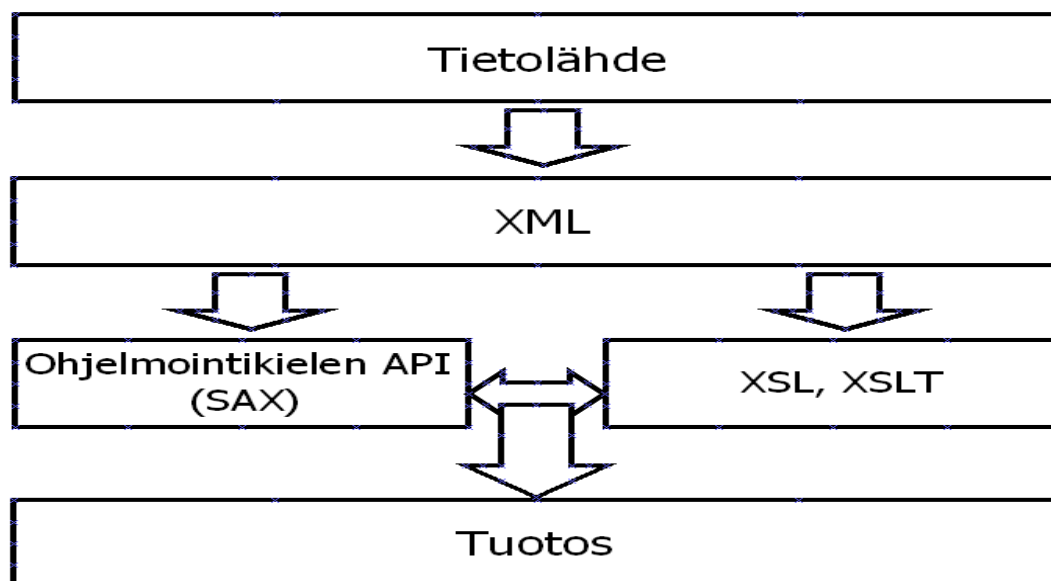
Asetetut tavoitteet saavutettiin *XML 1.0*:ssa ja *W3C* suositteli sitä ensimmäisen kerran 10.2.1998./11/

## 4.2.2 Käyttökohteet

*”XML on tarkoitettu rakenteisen tiedon esittämiseen. Rakenteinen tieto käsittää mm. taulukkolaskennan tiedostot, osoitekirjat, konfiguraatiotiedostot, sähköisen kaupankäynnin viestit sekä tekniset piirustukset. XML on joukko sääntöjä rakenteisen tiedon esittämiseen tarkoitettujen tekstiformaattien suunnitteluun. (XML:n voi nähdä myös ohjeena tai yhteisenä käytäntönä.) XML ei ole ohjelmointikieli eikä soveltajan tarvitse osata ohjelmoida pystyäkseen käyttämään sitä. XML:n avulla tietokoneiden on helppoa tuottaa ja lukea tietoa täsmällisessä muodossa. XML välttää suunnittelun tyypilliset sudenkuopat: se on laajennettavissa, järjestelmäriippumaton ja se tukee kansainvälistämistä ja lokalisointia. XML on myös täysin Unicode-yhteensopiva.”/12/*

Koska XML on laitteistoriippumaton, sitä voidaan käyttää missä tahansa sovelluksessa tai järjestelmässä/13/. Kaikki informaatio tallennetaan ja esitetään tekstimuodossa, joten tiedon siirtäminen järjestelmien ja sovellusten välillä on helppoa/13/. Tekstimuotoisen tallentamisen etuna on myös se, että informaatio on ihmisen luettavissa. Tämä helpottaa sovelluskehittäjiä suunnittelemaan ohjelmiaan ja löytämään niistä virheet helpommin kehitysvaiheessa.

Työskentely XML-sovellusten kanssa eroaa perinteisestä sovelluskehityksestä, koska tavoitteena on toteuttaa ”aitoa kolmikerrosmallia”. Kolmikerrosmalli on kuvattu kuviossa 9. Tässä mallissa informaatio, merkintä- ja esitystapa ovat erotettuja toisistaan. Informaatiokerros on ylimpänä mallissa. Se koostuu esimerkiksi tietokannasta tai hakemistosta, jonne informaatio on säilötty. Merkintätapakerros on XML-dokumentti. Esitystapakerros sisältää toiminnot, joiden avulla XML-dokumenttia voidaan muokata tai lukea. Käytännössä tämä tehdään jonkin ohjelmointikielen avulla, joka sisältää XML-dokumentin käsittelyyn tarkoitetun rajapinnan, kuten SAX./13/



**Kuvio 9.** Kolmikerrosmallin prosessikaavio./13/

### 4.2.3 SAX

*Simple API for XML (SAX)* on alun perin vain Javalle suunniteltu *XML*-muotoisen tiedon käsittelyrajapinta. Uusimmassa versiossa *SAX 2.0.1* on mukana rajapinnat myös useille muille ohjelmointikielille. *SAX* kehitettiin samaan aikaan *XML*-tekniikan kanssa ja saman asiantuntijajoukon toimesta./17/

*SAX*-rajapintaa käytettäessä *XML*-dokumentti käsitellään elementti elementiltä, eikä jo käsiteltyjä elementtejä tallenneta tietokoneen muistiin. Tällä tavoin voidaan pienentää sovelluksen tarvitsemää muistikapasiteettia, mutta navigointi dokumentin hierarkkisessa rakenteessa on kömpelöä.

## 4.3 FTP

Lyhenne *FTP* tulee sanoista *File Transfer Protocol*. Se on tiedonsiirtomenetelmä, joka on suunniteltu käytettäväksi kahden tietokoneen välillä. *FTP* käyttää tiedonsiirtoon *TCP*-protokollaa ja se toimii asiakas-palvelin-periaatteella./14/

### 4.3.1 Historia

*FTP*-protokollaa on kehitetty vuosien varrella aina 1970-luvulta lähtien. Ensimmäiset *FTP*:n kaltaiset tiedonsiirtomenetelmät kehitettiin *Massachusetts Institute*

*of Technology* -yliopistossa vuonna 1971. Vuotta myöhemmin *FTP*:aa ryhdyttiin käyttämään Yhdysvaltain puolustusministeriön verkossa tiedostojen siirtoon. Myöhemmin protokollaa on kehitetty tietoturvallisemmaksi ja luotettavammaksi, lisäämällä siihen muun muassa virheenkorjaus ja yhteyden salausmenetelmiä./14/

#### **4.3.2 FTP käytännössä**

*FTP*-protokolla tarvitsee toimiakseen *FTP*-palvelimen ja *FTP*-asiakkaan. Palvelinohjelmisto voi toimia joko passiivisessa tai aktiivisessa tilassa. Palvelinohjelmiston ollessa passiivinen, tiedonsiirron voi aloittaa vain, jos asiakas muodostaa yhteyden palvelimeen. Aktiivisessa tilassa oleva palvelinohjelmisto voi aloittaa tiedonsiirron ilman asiakkaan lähettämää pyyntöä./14/

Tietoa voidaan lähettää binääri- tai *ASCII*-muodossa. *ASCII*-muodossa lähetettävä data on yleensä merkkijono, jonka palvelin lähettää asiakkaalle merkiksi onnistuneesta komennosta. Tiedostot lähetetään binäärimuodossa./5/

## 5 SOVELLUKSEN MÄÄRITTELY JA SUUNNITELU

Sovelluksen tehtävänä on päivittää *Libraryssä* valmiiksi olevia dokumentteja. Päivittäminen tapahtuu silloin, kun näistä dokumenteista julkaistaan uusi revisio *Teamcenterissä*. Sovellus seuraa uusia hyväksytyjä revisioita käymällä läpi *Exportpollerin* kirjoittamaa hakemistoa. Ohjelma suoritetaan päivittäin, ja täysin automaattisesti ilman erikseen annettavia komentoja. Graafista käyttöliittymää ei tarvita.

### 5.1 Vaatimusmäärittelyt

Sovelluksen vaatimusmäärittelyt koottiin palavereissa yhdessä markkinointiviestintäpäällikkö Eila Nymanin kanssa vuoden 2012 aikana. Pelkästään *Libraryyn* liittyvistä asioista käytiin sähköpostikeskusteluja sen ylläpitäjien Marek Hacusin ja Marcin Pasteczkan kanssa.

### 5.1.1 Sovellukselta vaaditut ominaisuudet

Taulukossa 1 on lueteltu ominaisuudet, joita sovellukselta vaaditaan.

**Taulukko 1.** Sovellukselta vaadittavat ominaisuudet

Ominaisuus	Kuvaus
O1	Täysin automaattinen.
O2	Suoritetaan päivittäin samaan kellon aikaan.
O3	Seuraa uusien revisioiden ja niiden dokumenttien julkaisua <i>Teamcenterissä</i> .
O4	Päivittää <i>Libaryssä</i> olevien dokumenttien liitetiedosto ja revisio-kirjain. Muut dokumentin tiedot pysyvät ennallaan.
O5	Päivitettävien dokumenttien listaan on mahdollista lisätä nimikkeitä.
O6	Kirjoittaa lokitiedoston aina kun ohjelma suoritetaan. Lokista on luettavissa <i>Libraryyn</i> lähetetyt revisiot ja niiden dokumentit.

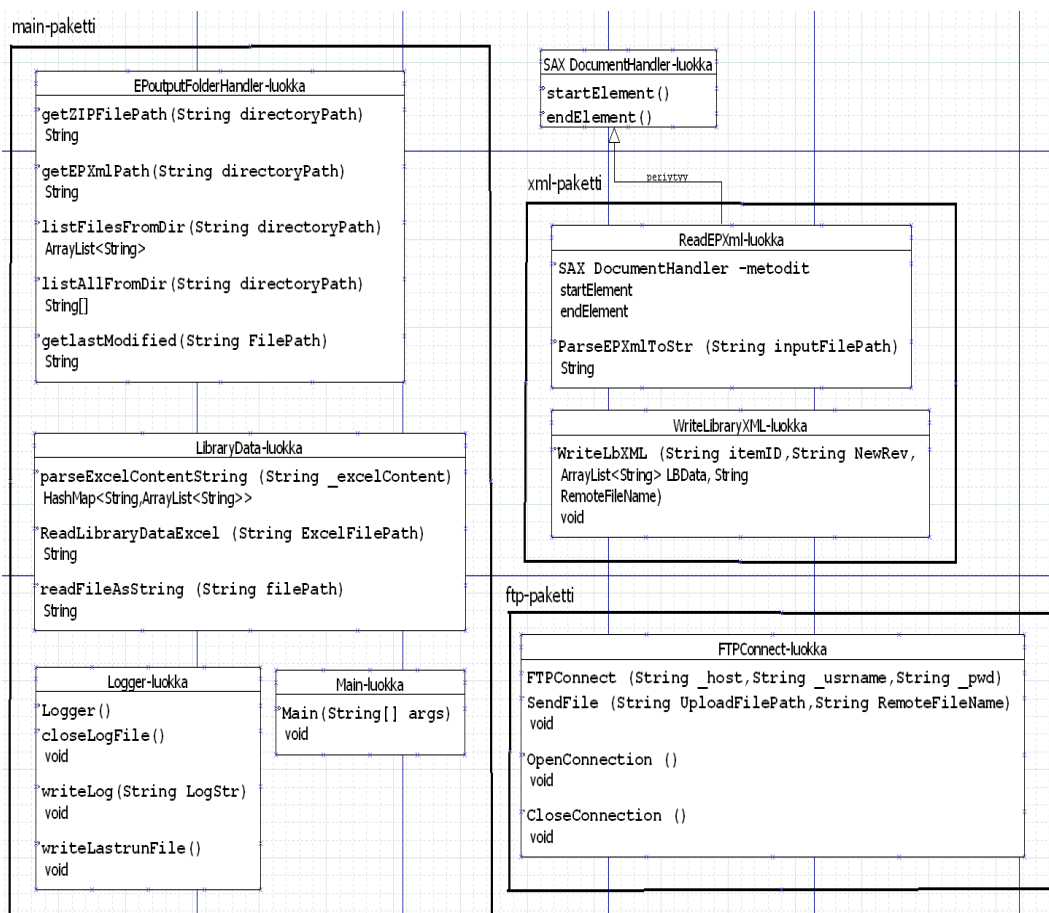
### 5.2 Laitteisto- ja ympäristövaatimukset

Sovellusta voidaan ajaa vain *ABB:n* verkossa olevassa tietokoneessa, jossa on *Windows*-käyttöjärjestelmä (*XP* tai *7*). Tietokoneessa tulee olla asennettuna *Java*-ohjelmisto ja sillä pitää olla pääsy yksikön *W*-verkkolevylle. *Libraryn* palvelin toimii myös yhtiön verkossa, joten *FTP*-portit ovat avoinna.

### 5.3 Luokat ja metodit

Tässä luvussa esitellään sovelluksen luokkien ja metodien toiminnot. Ohjelmisto koostuu kolmesta paketista: *Main*, *ftp* ja *xml*. Jokaisessa paketissa on samaan aihealueeseen kuuluvat luokat. Luokkia on yhteensä seitsemän kappaletta.

#### 5.3.1 Luokkakaavio



**Kuvio 10.** Luokkakaavio

Kuviossa 10 on kuvattuna sovelluksen luokat ja niiden metodit. Sulkujen sisällä metodin nimen vieressä on sen saamat parametrit. Metodin nimen alapuolella on sen muuttujan tyyppi jonka metodi palauttaa.



### 5.3.2 Main

Luokka sisältyy *main*-pakettiin ja se sisältää ainoastaan *Main*-metodin, se on siis sovelluksen pääluokka. Pääluokasta käsin kutsutaan sovelluksen muiden luokkien olioita ja suoritetaan eri toiminnot niiden avulla. Lokitiedoston kirjoitus suoritetaan tästä luokasta käsin käyttäen *Logger*-luokan olioita.

### 5.3.3 EPoutputFolderHandler

Luokka kuuluu *main*-pakettiin, ja sen avulla käydään läpi *Exportpollerin* luoma hakemisto. Luokka sisältää seuraavat metodit:

- *listAllFromDir()*-metodi palauttaa parametrinaan saamansa hakemistopolun sisällön *String*-taulukossa.
- *listFilesFromDir()*-metodi palauttaa parametrinaan saamansa hakemistopolun tiedostot *ArrayList*-muuttujassa.
- *getEPXmlPath()*-metodi etsii parametrinaan saamasta hakemistosta *xml*-tiedoston, jonka nimi alkaa merkkijonolla ”3GZF”. Palauttaa *String*-muuttujassa hakemistopolun jossa *xml*-tiedosto sijaitsee.
- *getLastModified()*-metodi palauttaa *String*-muuttujassa parametrinaan saamansa hakemistopolun päässä olevan hakemiston aikaleiman sen luontihetkestä. Aikaleima on muodossa: *dd-mm-yyyy hh:mm:ss*.

### 5.3.4 LibraryData

Luokka sisältyy *main*-pakettiin ja sitä käytetään *Excel*-tiedoston lukemiseen, sekä varmuuskopiotiedostojen luomiseen. Varmuuskopiot *Excel*-tiedoston sisällöstä luodaan jokaisella ohjelman ajokerralla. Luokka sisältää seuraavat metodit:

- *readFileAsString()*-metodi lukee parametrina annetun tiedoston *String*-muuttujaan ja palauttaa sen.
- *ReadLibraryDataExcel()*-metodi lukee parametrina saamansa *Excel*-tiedoston. Kirjoittaa projektin juureen varmuuskopiot *Excel*-tiedoston sisällöstä *txt*-formaattissa. Palauttaa tiedoston tekstisisällön *String*-muuttujassa.

- *parseExcelContentString()*-metodi lukee *ReadLibraryDataExcel*-metodin palauttaman *Stringin*. Luo siitä *HashMapin*, jossa on avaimena *ItemID* (esim. *3GZF500116-163*), ja arvona *ArrayList*-muuttuja joka sisältää kyseisen nimikkeen klassifiointidatat aina samassa järjestyksessä. Metodi palauttaa täytetyn *HashMapin*.

### 5.3.5 Logger

Luokka sisältyy *main*-pakettiin. Sen metodien avulla kirjoitetaan lokitiedosto, sekä tiedosto joka sisältää aikaleiman ohjelman viimeisimmästä ajokerrasta. Luokan konstruktorissa luodaan lokitiedosto, jonka nimi on muodossa: *log\_[aikaleima].log*. Tiedostoon kirjoitetaan käyttäen *writeLog()*-metodia.

### 5.3.6 ReadEPXml

Luokka sisältyy *xml*-pakettiin ja se periytyy *DefaultHandler*-luokasta. Luokan metodien avulla jäsennetään ja luetaan *Exportpollerin* luoma *xml*-tiedosto. Tiedostosta poimitaan *item*-elementtien attribuuttien sisältö. *Xml*-tiedosto jäsennetään käyttäen *SAX*-ohjelmointirajapintaa.

Luokka sisältää *DefaultHandler*-luokan metodien lisäksi *ParseEPXmlToStr()*-metodin. Tämä metodi käy läpi parametrinaan saamansa *XML*-tiedoston elementti kerrallaan, käyttäen *DefaultHandler*-luokan metodeja *StartElement()* ja *EndElement()*. Jos elementin nimi on *Item* ja sillä on attribuutteja, tallennetaan *itemIdentifier*- ja *revisionIdentifier*-attribuuttien arvot *String*-muuttujaan. Metodi palauttaa *String*-muuttujan, jossa on tabulaattorilla erotettuna nimikkeen *3GZF*-koodi ja revisiokirjain.

### 5.3.7 WriteLibraryXML

Luokka sisältyy *xml*-pakettiin. Sen metodien avulla kirjoitetaan *xml*-tiedosto, joka lähetetään *Libraryn* palvelimelle. Projektin juuressa on *XMLTemplate.xml*-tiedosto, jota käytetään pohjana *Libraryyn* lähetettävän *xml*-tiedoston luomisessa.

Luokan metodi *WriteLbXml()* saa parametrina: *itemID:n* (3GZF-koodi), sen liitetiedotiedoston nimen joka lähetetään palvelimelle ja revisiokirjaimen *String*-muuttujissa, sekä kyseisen nimikkeen muut tiedot *ArrayList*-muuttujassa. Näiden muuttujien sisältämän datan pohjalta kirjoitetaan *xml*-tiedosto *Libraryyn* lähetettäväksi. Metodi tallentaa luomansa *xml*-tiedoston projektin juureen *logs/transfers*-hakemistoon.

### 5.3.8 FTPConnect

Luokka sisältyy *ftp*-pakettiin. Sen konstruktorin ja metodien avulla luodaan yhteys *Libraryn* *FTP*-palvelimeen sekä lähetetään tiedostoja palvelimelle. Konstruktorissa luodaan *com.enterprisedt.net.ftp.FileTransferClient*-luokan olio, jonka avulla *ftp*-yhteys palvelimeen luodaan. Yhteys avataan *OpenConnection()*-metodin avulla ja suljetaan käyttäen *CloseConnection()*-metodia. Tiedostot palvelimelle lähetetään käyttäen *SendFile()*-metodia. Metodi saa parametrina *String*-muuttujassa olevan hakemistopolun, missä lähetettävä tiedosto sijaitsee sekä nimen, jonka mukaan tiedosto nimetään palvelimen päässä.

## 6 SOVELLUKSEN TOTEUTUS

Tässä luvussa esitellään sovelluksen toimintaa ja sen toteuttamistapaa yksityiskohtaisesti.

### 6.1 Yleistä toteutuksesta

Sovellus kehitettiin käyttäen *Java*-ohjelmointikieltä ja sen kehitysympäristön versiota *JDK 7*, joka oli uusin versio kehityshetkellä. Ohjelmoinnissa käytettiin pääosin *JDK 7* mukana tulevia luokkia ja kirjastoja. Ulkoisia kirjastoja tarvittiin vain *Excel*-tiedoston lukemiseen ja *ftp*-yhteyden hallintaan. Alla on luettelo sovelluksessa käytetyistä *JDK 7*:ään kuuluvista luokista:

- `java.io.BufferedWriter`
- `java.io.File`
- `java.io.FileInputStream`
- `java.io.FileWriter`
- `java.io.InputStream`
- `java.io.IOException`
- `java.text.DateFormat`
- `java.text.ParseException`
- `java.text.SimpleDateFormat`
- `java.util.ArrayList`
- `java.util.Date`
- `java.util.HashMap`
- `java.util.Iterator`
- `javax.xml.parsers.SAXParser`
- `javax.xml.parsers.SAXParserFactory`
- `org.xml.sax.Attributes`
- `org.xml.sax.helpers.DefaultHandler`
- `org.xml.sax.SAXException`
- `org.xml.sax.SAXParseException`

*Excel*-tiedoston lukemiseen käytettiin Apachen *POI 3.8 API*:a. *POI*:n avulla voidaan käsitellä seuraavien *Microsoft Office*-tuoteperheen sovellusten käyttämiä tiedostoformaatteja: *MS Excel*, *MS PowerPoint* ja *MS Word* /15/.

*Ftp*-yhteyden hallintaan käytettiin *Enterprise Distributed Technologies* -yrityksen tarjoamaa ilmaista *Javalle* kehitettyä *ftp*-kirjastoa. Kirjastosta on saatavilla myös maksullinen versio, joka tarjoaa muun muassa tekniikoita *ftp*-yhteyden salaamiseen. Tässä työssä yhteyden salaamiseen ei ollut tarvetta. Ilmaisversio on avoimen lähdekoodin kirjasto ja sitä käytetään *LGPL*-lisenssin mukaisin käyttöehdoin./16/

### **6.1.1 Sovelluksen toiminnot**

Taulukossa 2 on kuvattu toiminnot jotka sovellus suorittaa. Toiminnot suoritetaan taulukon mukaisessa järjestyksessä. Osa toiminnoista suoritetaan vain silloin kun sovellus löytää uusia revisioita *Exportpollerin* hakemistosta joiden liitetiedostona on *Libraryssä* oleva dokumentti. Kolmanteen sarakkeeseen on merkitty ”kyllä” tai ”ei”, riippuen siitä suoritetaanko toiminto aina kun sovellus käynnistetään.

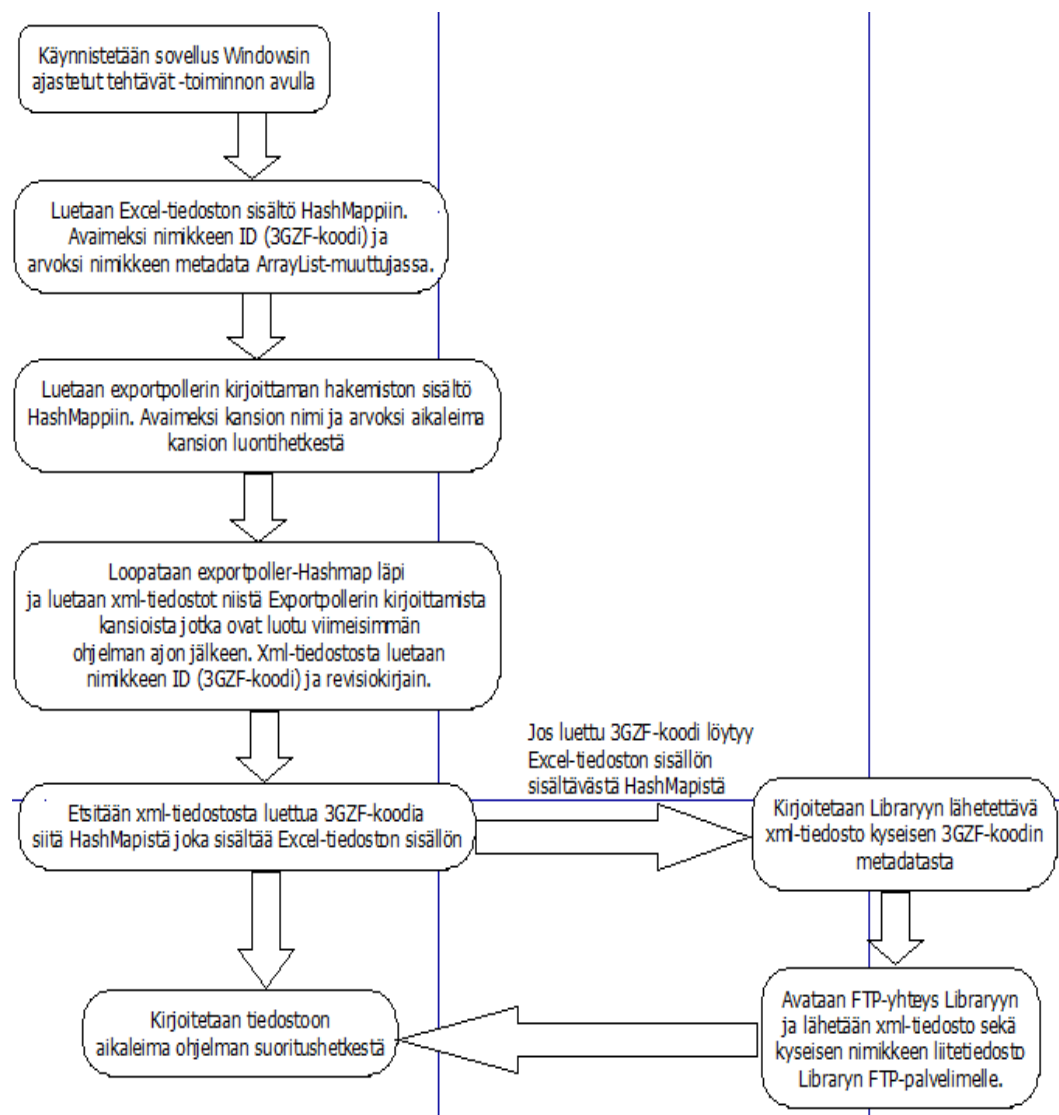
**Taulukko 2.** Sovelluksen suorittamat toiminnot

Toiminto	Kuvaus	Suoritetaan aina
T1	Luetaan Excel-tiedostot joissa on tiedot Libraryssä olevista dokumenteista.	Kyllä
T2	Luetaan tiedosto jossa on aikaleima ohjelman edellisestä ajokerrasta.	Kyllä
T3	Käydään läpi ne hakemistot jotka Exportpoller on luonut edellisen ohjelman ajon jälkeen.	Kyllä
T4	Luetaan Exportpollerin kirjoittamat xml-tiedostot uusista hakemistoista.	Kyllä
T5	Kirjoitetaan xml-tiedosto Libraryä varten.	Ei
T6	Avataan FTP-yhteys Libraryn palvelimeen.	Ei
T7	Lähetetään xml- ja liitetiedostot Libraryn palvelimelle.	Ei
T8	Kirjoitetaan tiedostoon aikaleima ohjelman ajohetkestä.	Kyllä

## 6.2 Eri osioiden toteutus

Sovellus on kehitetty koodaamalla jokainen toiminto erikseen sekä testattu ne toimiviksi yksi kerrallaan. Tämän jälkeen kirjoitettiin pääluokka, jossa suoritetaan sovelluksen eri toiminnot kutsumalla muiden luokkien metodeja. Pääluokassa käsitellään myös kaikkien metodien eteenpäin heittämät poikkeukset, jos niitä syn-

tyy ohjelman ajon aikana. Ainoa sovelluksen osio, joka suoritetaan pääluokassa, on *Excel*-tiedostossa olevien nimikkeiden ja *Exportpollerin* output-hakemistossa olevien nimikkeiden vertailu. Jos vertailussa löytyy samoja nimikkeitä molemmista lähteistä, suoritetaan kaikki taulukossa 2 ja kuviossa 11 kuvatut toiminnot. Jokaisen toiminnon jälkeen kirjoitetaan lokitiedostoon tiedot sen suorittamisesta.



**Kuvio 11.** Toimintatapakaavio

Sovellus voidaan jakaa kuuteen osioon:

1. *Exportpollerin* kirjoittaman *output*-hakemiston läpi käynti
2. *Exportpollerin* kirjoittaman *xml*-tiedoston lukeminen
3. *Excel*-tiedoston lukeminen
4. *Excel*-tiedostossa olevien nimikkeiden ja *output*-hakemistossa olevien nimikkeiden vertailu
5. *Xml*-tiedoston kirjoittaminen
6. Tiedostojen lähetys

### 6.2.1 Sovelluksen käynnistys

Sovellus käynnistetään päivittäin samaan kellon aikaan käyttäen *Windows*-käyttöjärjestelmän ajoitetut tehtävät -toimintoa. Tämä tehdään lisäämällä *Library\_export.jar*-tiedosto ajoitettujen tehtävien listaan. Tehtävän käynnistyskansiksi laitetaan se hakemisto, jossa *jar*-tiedosto sijaitsee. Samassa hakemistossa pitää olla kansio *logs* ja sen alihakemistona *transfers*, jotta sovellus saadaan suoritettua.

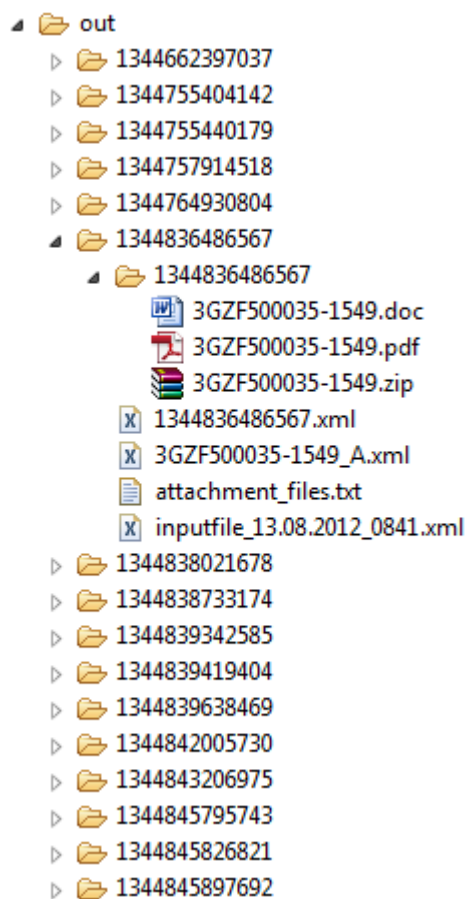
Sovellus voidaan käynnistää myös komentoriviltä, jolloin sille voidaan antaa parametreina *FTP*-palvelimen tiedot, *Excel*-tiedoston sijainti ja *Exportpollerin output*-hakemiston sijainti. Jos sovellus käynnistetään ilman parametreja, se käyttää lähdekoodiin kovakoodattuja osoitteita ja hakemistopolkuja. Kovakoodatut tiedot ovat:

- *Excel*-tiedoston sijainti: Projektin juurihakemisto, eli sama hakemisto missä *jar*-tiedosto sijaitsee.
- *Exportpollerin output* hakemiston sijainti: `W:\Export_Poller\OUT\`
- *FTP*- palvelimen osoite: `abblibraryftp.abb.com`
- *FTP*-palvelimen käyttäjätunnus: `*****`
- *FTP*-palvelimen salasana: `*****`



## 6.2.2 Exportpollerin kirjoittaman hakemiston lukeminen

*Exportpoller* kirjoittaa output-hakemistoon *xml*-tiedostoja uusista *Teamcenter*issä hyväksytyistä nimikkeen revisioista, sekä kopioi saman hakemiston alihakemistoihin kyseiseen revisioon liittyvät liitetiedostot. *Output*-hakemiston juuressa on hakemistoja, jotka on nimetty 13-merkkisellä juoksevalla numerosarjalla. Jokainen näistä hakemistoista sisältää vain yhden nimikkeen revision tiedot ja hakemistojen nimet ovat yksilöllisiä. Juoksevalla numerosarjalla nimetyn hakemiston alihakemistona on samalla numerosarjalla nimetty hakemisto, joka sisältää kyseisen revision liitetiedostot.



**Kuvio 12.** *Exportpoller*in output-hakemiston rakenne.

Kuviossa 12 on kuvattu *Exportpoller*in kirjoittaman hakemiston rakenne. Sovellus lukee tästä hakemistosta sen juuressa olevien hakemistojen nimet ja niiden *Last*

*Modified* – arvon, *xml*-tiedoston, joka alkaa merkkijonolla ”3GZF” sekä *pdf*-tiedoston absoluuttisen hakemistopolun.

*Output*-hakemiston sisältö luetaan *HashMap*-muuttujaan, jonka avaimiksi tulevat juoksevalla numerosarjalla nimettyjen hakemistojen nimet *String*-tyyppisinä ja arvoiksi näiden hakemistojen *Last modified* -arvo *Java.util.Date*-luokan oliona. *Output*-hakemiston sisältö luetaan *Java.io.File*-luokan *list()*-metodilla, joka lukee parametrina saamansa hakemistopolun sisältämien hakemistojen ja tiedostojen nimet *String*-tyyppiseen taulukkoon. Aikaleima hakemiston luomisesta saadaan saman luokan *lastModified()*-metodilla, joka palauttaa aikaleiman *long*-tyyppisenä. *Long*-muuttujassa oleva aikaleima jäsennetään *Date*-olioksi, joka sijoitetaan *HashMap*-muuttujan arvoksi.

*Exportpollerin* kirjoittamaa hakemistoa ei tarvitse käyttäjien lukea tai muokata, joten voidaan luottaa siihen, että hakemistojen *Last modified* -arvo on suunnilleen sama kuin aikaleima siitä hetkestä, kun *Teamcenterissä* käynnistetään revision hyväksyntäkierron toinen vaihe.

### 6.2.3 Excel-tiedoston lukeminen

*Librarydata.xlsx*-tiedostossa tulee olemaan tiedot kaikista niistä *pdf*-formaattissa olevista mittapiirustuksista jotka ovat *Libraryssä*. Tiedosto luodaan käyttäen, *TDB ATEM Motors* –tietokannan, *Get Doc Characteristics into excel extended version* –toimintoa. Toiminto on kehitetty tätä työtä varten *Libraryn* ylläpitäjien toimesta. Tiedostoon on tallennettu kaikki metadata *Libraryssä* olevista revisioista, jota tarvitaan sinne lähetettävän *xml*-tiedoston luomiseen. Tiedoston A-sarakkeessa on *3GZF*-koodi, joka sijoitetaan *HashMap*-muuttujan avaimeksi. *HashMap*-muuttujan arvoksi sijoitetaan *ArrayList*-muuttuja, jonka sisältö on *Excel*-tiedoston yhdeltä riviltä sarakkeet B-L. *Excel*-tiedoston tekstisisältö luetaan *String excel-Content*-muuttujaan *org.apache.poi.xssf*-luokan metodeilla, jos kyseessä on *xlsx*-päätteinen tiedosto. *Xls*-päätteisten *Excel*-tiedostojen lukemiseen käytetään *org.apache.poi.hssf*-luokan metodeita.

*String excelContent*-muuttujan sisältö kirjoitetaan aina *ExcelContentBackup.txt*-tiedostoon siltä varalta, ettei *Excel*-tiedoston lukeminen onnistu. Näin voi tapahtua esimerkiksi silloin, jos tiedosto on avoinna käyttäjällä juuri sillä hetkellä, kun sovellus yrittää sitä lukea. Silloin luetaan *Library*n metadatat *backup*-tiedostosta. Kun *Excel*-tiedoston sisältö on *String*-tyyppisessä muuttujassa, voidaan käyttää *Java.lang.String*-luokan metodeja *HashMap*in täyttämisen apuna. *Excel*-tiedoston rivit on eroteltu toisistaan ”\n”-erottimella ja solut ”\t”-erottimella. Kuviossa 13 on kuvattu *excelContent*-muuttujan sisällön sijoittaminen *HashMap*iin.

```
HashMap<String,ArrayList<String>> lbrDataInMap = new HashMap<String,ArrayList<String>>();

//splitataan excelContent rivinvaihdon kohdalta
String[] line = excelContent.split("\n");

//käsitellään jokainen solu line-muuttujassa(rivi excelContent-muuttujassa) erikseen.
for(int i = 0; i < line.length; i++)
{
    //splitataan rivi \t:n (TAB) kohdalta, eli jokaisen solun kohdalta excel-tiedostossa.
    String[] lineSplit = line[i].split("\t");

    //jos excel-tiedoston ensimmäinen solu sisältää "3GZF"-merkkijonon,
    //luodaan rivistä avain-arvo pari joka sijoitetaan metodin palauttamaan HashMappiin.
    if(lineSplit[0].contains("3GZF"))
    {
        //Luodaan apu List joka sijoitetaan arvoksi HashMappiin
        ArrayList<String> apu = new ArrayList<String>();

        //Puhdistetaan solujen sisältö mahdollisista ylimääräisistä merkeistä
        //ja täytetään ArrayList
        for (int a = 1; a < lineSplit.length; a++ )
        {
            lineSplit[a] = lineSplit[a].replaceAll("\r\n", "");
            lineSplit[a] = lineSplit[a].replaceAll("\t", "");
            lineSplit[a] = lineSplit[a].trim();

            apu.add(lineSplit[a]);
        }
        lineSplit[0] = lineSplit[0].trim();
        lbrDataInMap.put(lineSplit[0], apu);
    }
}

return lbrDataInMap;
```

**Kuvio 13.** *Excel*-tiedoston lukeminen *HashMap*-muuttuunaan.

#### 6.2.4 Uusien nimikkeiden etsintä

Uusilla nimikkeillä tarkoitetaan niitä nimikkeen revisioita, jotka on julkaistu *Teamcenter*issä *lastrun*-tiedostossa olevan aikaleiman jälkeen. *Lastrun*-tiedostoon kirjoitetaan aikaleima aina siitä hetkestä kun sovelluksen suoritus alkaa. Uudet

nimikkeen revisiot löydetään iteroimalla läpi *HashMap*, jossa on *Exportpollerin* kirjoittaman hakemiston sisältö. *HashMap*issa olevia *Last modified* -arvoja verrataan *Lastrun*-tiedostossa olevaan aikaleimaan. Jos vertailussa löydetään tämän aikaleiman jäkeen luotuja hakemistoja, etsitään näistä *xml*-tiedostot, joista luetaan *SAX*-rajapintaa käyttäen nimikkeiden *3GZF*-koodit ja revisiokirjaimet. Luettua *3GZF*-koodia etsitään sen *HashMap*-muuttujan avainten joukosta, joka sisältää tiedot *Library*ssä olevista nimikkeistä. Tällä tavoin löydetään uusien nimikkeen revisioiden joukosta revisiot, jotka *Library*yn halutaan lähettää.

### 6.2.5 Xml-tiedoston luominen ja tiedostojen lähetys

Mikäli *Exportpollerin* kirjoittamasta *xml*-tiedostosta luettu *3GZF*-koodi löydetään *Library*ssä olevien nimikkeiden tiedot sisältävästä *HashMap*istä, kirjoitetaan *xml*-tiedosto *Library*yn lähetettäväksi. Tarvittavat tiedot tiedoston luomiseen ovat *ArrayList*-muuttujassa, joka on sijoitettu *HashMap*in arvoksi. *WriteLbXml()*-metodi saa parametrina tämän *ArrayList*in. Tiedoston pohjana käytetään *xmlTemplate.xml*-tiedostoa, johon korvataan tarvittavien elementtien sisällöt. Uusi *xml*-tiedosto tallennetaan projektin juuressa olevaan *transfers*-hakemistoon.

*Xml*-tiedoston kirjoittamisen jälkeen avataan *FTP*-yhteys *Library*yn palvelimeen. Ensin palvelimelle lähetetään *Library*yn päivitettävä dokumentti ja sen jälkeen ohjausparametrit sisältävä *xml*-tiedosto. Molemmat tiedostot lähetetään käyttäen *FTP*-luokan *SendFile()*-metodia. Metodille annetaan parametrina kaksi *String*-muuttujaa. Ensimmäiseen muuttujaan sijoitetaan lähetettävän tiedoston absoluuttinen hakemistopolku ja toiseen muuttujaan se hakemistopolku palvelimella, mihin tiedosto halutaan lähettää.

## 7 JOHTOPÄÄTÖKSET

Sovellus toteutettiin *Java*-ohjelmointikielellä, koska *Teamcenter* on myös *Java*-pohjainen. Ohjelmaa tullaan tulevaisuudessa kehittämään enemmän *Teamcenterin* yhteydessä toimivaksi. Sovelluskehitys on helpompaa, kun yhdistettävät sovellukset on ohjelmoitu käyttäen samaa ohjelmointikieltä. *Javan* valikoitumiseen ohjelmointikieleksi vaikutti myös oma osaamiseni.

### 7.1 Tavoitteiden saavuttaminen

Työlle asetetut tavoitteet saavutettiin pääosin. Suunnitelma uusien dokumenttien luomisesta *Libraryyn* hylättiin jo ennen kuin työn sisältö suunniteltiin. Hylkääminen johtui siitä, että *Teamcenterin* päätietolomakkeella ei ole tällä hetkellä tarpeeksi tietoa mittapiirustuksista, jotta ne voitaisiin kategorioida oikein *Libraryssä*. Metadata jota lomakkeilta olisi voitu käyttää, on tallennettu pääosin vapaatekstikenttiin, joten tietoa on kirjoitettu monella eri tavalla. Tällaisiin kenttiin tallennettua tietoa on vaikea käsitellä ohjelmallisesti.

*CAD*-mallinnusohjelmalla luotuja mittapiirustuksia, jotka ovat *dxf* – tai *zip* – formaatissa, ei voida tällä hetkellä päivittää *Libraryyn* tämän sovelluksen avulla. Päivittäminen olisi mahdollista, jos *CAD*-tiedostojen nimeämistapa olisi yhtenäistä kaikkien nimikkeiden kohdalla. *Zip*-paketteihin on pakattu usein muitakin tiedostoja, kuin *CAD*-mallinnusohjelmalla luotuja mittapiirustuksia. Ei siis voida tällä hetkellä varmistaa että *Libraryyn* päätyisi vain sinne tarkoitetut tiedostot.

Sovellus on testattu toimivaksi yhdessä *Libraryyn* ylläpitäjien kanssa. Sovellusta ei kuitenkaan voida ottaa käyttöön ennen kuin on luotu *Excel*-tiedosto, joka sisältää tiedot *Libraryssä* olevista mittapiirustuksista. Tiedosto luodaan valitsemalla *TDB ATEM Motors* – tietokannasta kaikki ne dokumentit, joiden päivittäminen halutaan mahdollistaa sovelluksen avulla. Valinnan jälkeen käytetään *Get Doc Characteristics into excel extended version* – toimintoa, joka tuo tiedot tietokannasta *Excel*-tiedostoon.

## 7.2 Jatkotoimet

*CAD*-mittapiirustusten saaminen päivitettävien dokumenttien listaan edellyttää *dxf*-tiedostojen tallennustavan yhtenäistämistä *Teamcenter*issä, sekä ohjeistamalla käyttäjät käyttämään yhtenäistä tapaa tiedostojen nimeämisessä. Tämän jälkeen sovellusta kehitetään siten, että se päivittää myös *CAD*-mallinnusohjelmalla luodut mittapiirustukset *Library*yn.

Uusien dokumenttien luominen *Library*yn voidaan mahdollistaa rakentamalla *Teamcenter*iin klassifiointi mittapiirustuksia varten. Klassifiointi on yksi *Teamcenter*in ominaisuuksista, jonka avulla voidaan varmistaa, että tallennettava tieto on luettavissa ohjelmallisesti. Tämä voidaan tehdä esimerkiksi korvaamalla vapaatekstikentät valintalistoilla. Sen jälkeen kun metadata on kunnossa *Teamcenter*issä mittapiirustusten osalta, voidaan yhdistää tämän sovelluksen *xml*-tiedoston kirjoitus- ja *ftp*-yhteysluokka suoraan *Teamcenter*iin käyttäen sen tarjoamaa ohjelmointirajapintaa. Tämän jälkeen tietoja ei tarvitse enää lukea ulkoisesta *Excel*-tiedostosta.

## LÄHTEET

/13/ 2Kmediat. Miksi XML-teknologiaa tulisi hyödyntää? Saatavilla verkossa. Viitattu 29.11.2012. <http://www.2kmediat.com/xml/johdanto2.asp>

/1/ ABB Oy, ABB Lyhyesti. Saatavilla verkossa. Viitattu 10.3.2012. <http://www.abb.fi/cawp/fiabb251/657dfdcf6e344cc7c1256b20003149ae.aspx>

/17/ About SAX. Saatavilla verkossa. Viitattu 6.12.2012. <http://www.webopedia.com/TERM/S/SAX.html>

/8/ About the Java Technology. Saatavilla verkossa. Viitattu 23.11.2012 <http://docs.oracle.com/javase/tutorial/getStarted/intro/definition.html>

/15/ Apache POI - the Java API for Microsoft Documents. Saatavilla verkossa. Viitattu 6.12.2012. <http://poi.apache.org/>

/16/ Enterprise Distributed Technologies, edtFTPj/free. Saatavilla verkossa. Viitattu 6.12.2012. <http://www.enterprisedt.com/products/edtftp>

/2/ IDEAL PLM, Teamcenter. Saatavilla verkossa. Viitattu 23.10.2012 <http://fi.ideal-plm.com/static/teamcenter/>

/6/ JAVA. What is JAVA? Saatavilla verkossa. Viitattu 22.11.2012. [http://www.java.com/en/download/faq/whatis\\_java.xml](http://www.java.com/en/download/faq/whatis_java.xml)

/9/ Java Language Specification, Chapter 1, Introduction. Saatavilla verkossa. Viitattu 23.11.2012. <http://docs.oracle.com/javase/specs/jls/se7/html/jls-1.html>

/14/ RFC 959 Standard. FILE TRANSFER PROTOCOL. J.Postel, 1985. Saatavilla verkossa. Viitattu 29.11.2012. <http://tools.ietf.org/html/rfc959>

/3/ SIEMENS PLM Software. Explore Teamcenter. Saatavilla verkossa. Viitattu 23.10.2012. [http://www.plm.automation.siemens.com/en\\_us/products/teamcenter/index.shtml](http://www.plm.automation.siemens.com/en_us/products/teamcenter/index.shtml)

/4/ Teamcenter käyttöohje 2012. ABB Oy, Jonathan Heyno ja Mikko Hämäläinen. ABB:n sisäinen ohje.

/11/ The History of XML, Total XML, Delta Consulting. Saatavilla verkossa. Viitattu 28.11.2012. <http://www.totalxml.net/history-xml.php>

/7/ Timeline of Java. Saatavilla verkossa. Viitattu 22.11.2012. <http://oracle.com.edgesuite.net/timeline/java/>

/5/ Using ABB Library, 2012, ABB Oy, Bartosz B Zajac. ABB:n sisäinen ohje.

/10/ W3C, Extensible Markup Language, Introduction. Saatavilla verkossa. Viitattu 27.11.2012. <http://www.w3.org/XML/>

/12/ W3C:n Suomen toimisto. XML10 kohdan tiivistelmä. Saatavavilla verkossa. Viitattu 28.11.2012. <http://www.w3c.tut.fi/translations/xml/xmlin10pts/>