



Valmet DNA Single Window -sovellus OPC UA -linkillä

Olli Höyssä

OPINNÄYTETYÖ
Marraskuu 2019

Sähkö- ja automaatiotekniikka
Automaatiotekniikka

TIIVISTELMÄ

Tampereen ammattikorkeakoulu
Sähkö- ja automaatiotekniikka
Automaatiotekniikka

HÖYSSÄ, OLLI:

Valmet DNA Single Window -sovellus OPC UA -linkillä

Opinnäytetyö 47 sivua, joista liitteitä 0 sivua
Marraskuu 2019

Lähtökohtana tälle opinnäytetyölle oli Valmet Automation Oy:n asiakasprojekti, jonka päämääränä oli toimittaa paperikoneen sähkökäyttöjen ohjausjärjestelmälle käyttöliittymäsovellus. Linkki käyttöliittymän ja ohjausjärjestelmän välille tuli toteuttaa uudella tiedonsiirtomenetelmällä, jota kyseiseen käyttötarkoitukseen ei ollut vielä käytetty. Opinnäytetyössä kehitettiin vanhan käyttöliittymäsovelluksen pohjalta uusi tuote, jonka avulla voidaan generoida OPC UA -tiedonsiirtomäärittelyä tukeva käyttöliittymäsovellus myös jatkossa vastaaviin projekteihin.

Työn alussa hankittiin tietoa eri tiedonsiirtomenetelmistä, paperikoneiden sähkökäyttöjärjestelmistä sekä sähkökäyttöjen ohjausjärjestelmistä. Selvitettävää oli myös vastaavanlaisten käyttöliittymäsovelluksien toteuttamisessa Valmet DNA -automaatiojärjestelmällä. Näiden tietojen perusteella kehitettiin generointialusta uudelle käyttöliittymäsovellustuotteelle ja generoitiin lopullinen käyttöliittymäsovellus asiakasprojektiin. Sovellus testattiin OPC UA -tiedonsiirtolinkin ja toiminnallisuuksien osalta osana projektin FAT-testiä.

Testausten tuloksena voitiin OPC UA -määrittelyä käyttävä käyttöliittymäsovellus todeta tiedonsiirtolinkin osalta muutamaa poikkeusta lukuun ottamatta toimivaksi. Tietotyyppien ristiriitoihin liittyvät toimimattomat linkin osat olivat kuitenkin korjattavissa helposti, ja kriittiset virheet korjattiin jo testausten aikana. Sovelluksen toiminnallisuuksissa oli vielä FAT-testin jälkeen joitakin vikoja, mutta myös ne saatiin korjattua pian testien jälkeen. Erylistä huomiota vaativiksi seikoiksi testausten aikana nousi linkin konfigurointi tietoturva-asetusten ja linkissä kulkevien signaalien muuttujamäärittelyjen osalta.

Valmet Automation Oy:llä on työn tulosten perusteella hyvät valmiudet toimittaa vastaavanlaisia OPC UA -liityntätekniiikkaa hyödyntäviä käyttöliittymäsovelluksia uusiin projekteihin. Toimitusvalmiutta voidaan vielä parantaa korjaamalla generointialustaa testauksessa havaittujen virheiden osalta ja kehittämällä työkalu linkkiä käyttävien muuttujien konfigurointia varten. Tämä vähentäisi työkuormaa ja parantaisi tuotteen laatua.

ABSTRACT

Tampereen ammattikorkeakoulu
Tampere University of Applied Sciences
Degree Programme in Electrical and Automation Engineering
Automation Engineering

HÖYSSÄ, OLLI:
Valmet DNA Single Window Application with OPC UA Link

Bachelor's thesis 47 pages, appendices 0 pages
November 2019

This thesis was based on Valmet Automation's customer project where a user interface application had to be provided for the electrical drive control system of a paper machine. The link between the user interface and the control system had to be implemented with a new industrial telecommunication protocol that had not yet been applied for the purpose in question. The objective of this thesis was to develop a new application which supports the OPC UA specification, and which can be utilized in corresponding projects also in future. For the development, an old user interface application was used as a basis.

At first, data were acquired on various data transmission methods, electrical drive systems for paper machines, control systems for electrical drives and Valmet DNA user interface applications. Based on collected information, a generation platform for user interface application was implemented and the application for the customer project was generated using the new generation platform. The OPC UA communication link and functionalities of the user interface application were tested as a part of factory acceptance testing (FAT) of the project.

The results of the testing showed that the OPC UA interface of the application was working correctly despite a few exceptions. However, these exceptions were related to data type conflicts and could be fixed with a relatively little effort, and the errors were corrected already during the testing. Functionalities of the application still had some flaws after the FAT, but they were fixed soon afterwards as well. During the testing, particular attention was paid to configuration of the OPC UA link in terms of security settings and variable definitions.

The results of this work made Valmet Automation capable to supply the user interface applications utilizing the OPC UA specification for new similar projects. The delivery capability can be further improved by correcting the generation platform for errors detected during testing and by developing a tool for configuration of variables to reduce the workload and improve the quality of the product.

Key words: industrial data transmission, OPC UA, Valmet DNA

SISÄLLYS

1	JOHDANTO	6
2	PAPERIKONEEN LINJAKÄYTÖN OHJAUS.....	7
2.1	Linjakäyttöjärjestelmän rakenne.....	7
2.2	Ohjausjärjestelmät	8
2.3	Käyttöliittymät.....	9
3	VALMET DNA SINGLE WINDOW -SOVELLUS	10
3.1	Operointinäytöt.....	11
3.2	Liityntärajapinta	14
3.2.1	Siemens 3964R	15
3.2.2	Modbus TCP.....	16
3.2.3	Profinet ja Profibus DP	17
3.2.4	OPC.....	18
4	OPC-TIEDONSIIRTOSTANDARDI.....	19
4.1	OPC yleisesti.....	19
4.2	OPC Classic.....	20
4.3	OPC Unified Architecture	22
4.3.1	Tietoturva ja kommunikointi	22
4.3.2	Alustariippumattomuus	23
4.3.3	Tietomallintaminen	24
4.3.4	Yhteensopivuus	25
5	VALMET DNA OPC UA -ASIAKASOHJELMA.....	26
5.1	Muuttujien konfigurointi (Variable Configuration File).....	27
5.2	DNA OPC UA Client -sovelluksen konfigurointi (CPU File).....	28
6	OPC UA SINGLE WINDOW -TOTEUTUS.....	30
6.1	Generoitavan Single Window -sovelluksen tuottaminen	30
6.1.1	Generoinnin yleinen periaate.....	30
6.1.2	Sovelluspohjien tekeminen.....	33
6.1.3	Muokkaukset generointikoodissa.....	35
6.2	OPC UA Single Window -linkin järjestelmärakenne	35
6.3	Tuotteistuksen soveltaminen asiakasprojektissa.....	36
7	TESTAUS JA TULOKSET	39
7.1	Sisäinen testaus.....	39
7.2	Varsinainen FAT-testaus.....	41
8	JOHTOPÄÄTÖKSET JA POHDINTA.....	44
	LÄHTEET.....	46

LYHENTEET JA TERMIT

ACN	Application and Control Node
BU	Backup
COM	Component Object Model
DCOM	Distributed Component Object Model
DCS	Distributed Control System
DNA	Dynamic Network of Applications
DP	Profibus Decentralized Peripherals
EAS	Engineering Activity Server
ERP	Enterprise Resource Planning
FAT	Factory Acceptance Testing
HMI	Human Machine Interface
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
I/O	Input/output
LIS	Valmet DNA Logic Interface Server
LWC	Light Weight Coated
MES	Manufacturing Execution System
OLE	Open Linking and Embedding
OPC	OLE for Process Control
OPC DA	OPC Data Access
OPC UA	OPC Unified Architecture
OPS	Operation Server
PCS	Process Control Server
PLC	Programmable Logic Controller
PN	Profinet
PYTHON	tulkattava ohjelmointikieli
SCADA	Supervisory Control and Data Acquisition
TCP/IP	Transmission Control Protocol/Internet Protocol
TLS	Transport Layer Security
VBA	Visual Basic for Applications

1 JOHDANTO

Tämä opinnäytetyö on tehty Valmet Automation Oy:n toimeksiantona. Aihe työlle syntyi tarpeesta toimittaa käyttöliittymä- ja liitäntäsovellus paperikoneen linjakäyttöjen ohjausjärjestelmälle soveltaen OPC UA -tiedonsiirtoprotokollaa. Käyttöliittymäsovelluksella mahdollistetaan toisen automaatiojärjestelmän ope- roiminen Valmet DNA -valvomonäytöistä, mikä eliminoi tarpeen käyttää kahta eri valvomojärjestelmää. Tästä juontuu nimitys Single Window.

Työn tarkoituksena on kehittää generoitava käyttöliittymäsovellustuote, jonka lii- täntäsovellusosuus pohjautuu OPC UA -kommunikointiprotokollaan. Liitäntäso- velluksen avulla käyttöliittymä liitetään kolmannen osapuolen toimittamaan linja- käyttöjen ohjausjärjestelmään. Tuotteistuksen pohjalta generoidaan käyttöliit- tymä- ja liitäntäsovellus (Single Window -sovellus) asiakasprojektiin, jossa pa- peritehtaan prosessi- ja koneohjaukset ovat toteutettu Valmet DNA -automa- tiojärjestelmällä ja käyttöjenohjausjärjestelmä on kolmannen osapuolen toimit- tama. Tarkoituksena on myös testata generoidun sovelluksen toiminta OPC UA -linkin ja toiminnallisuuksien osalta.

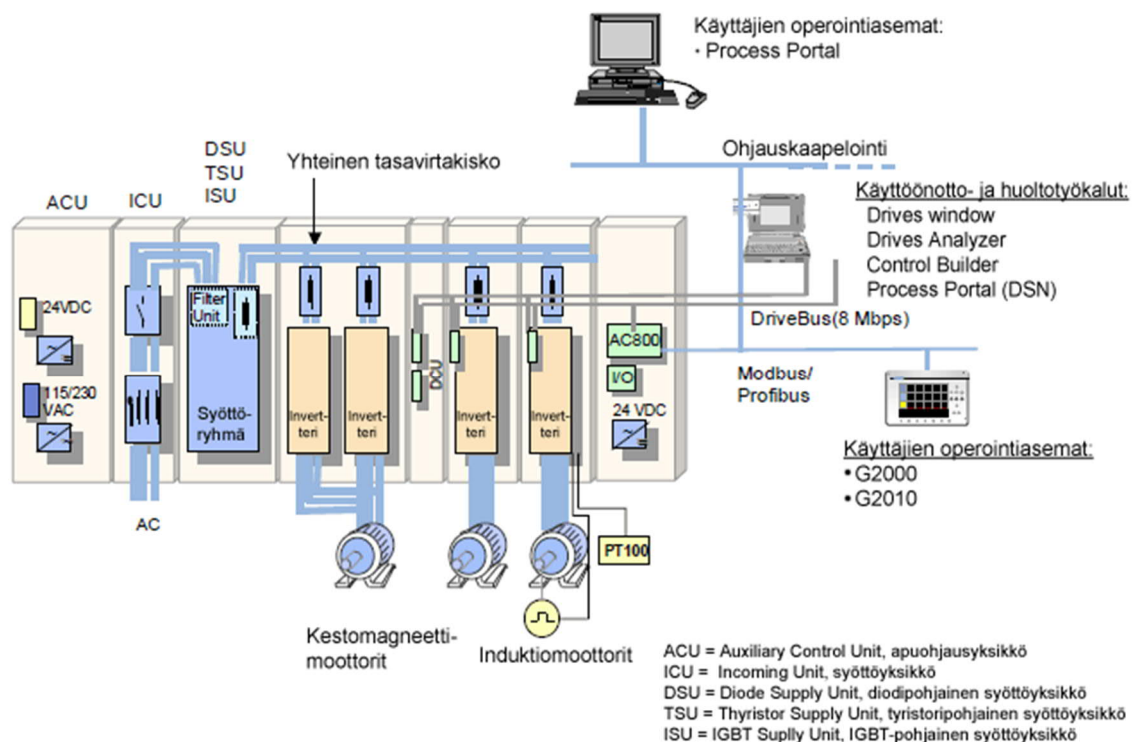
Opinnäytetyön tavoitteena on mahdollistaa paperikoneen linjakäyttöjen käyttö- liittymäsovelluksen toimittaminen uusimmalla OPC-liityntäteknikalla. Työn tulok- sena syntyvää generoitavaa sovellustuotetta voidaan käyttää kaikissa vastaa- vissa projekteissa tulevaisuudessa, mikä vähentää projektikohtaista työmäärää ja pienentää inhimillisten virheiden mahdollisuutta sovelluksen suunnitteluvai- heessa. Tämän ansiosta loppuasiakkaalle toimitettava sovellus on laaduk- kaampi, ja tarvittava testausaika tilaajan tehtaalla jää lyhyemmäksi. Tavoitteena on myös hankkia tietoa OPC UA -tiedonsiirtoprotokollasta ja sen käyttämisestä Valmet DNA -automaatiojärjestelmän kanssa.

2 PAPERIKONEEN LINJAKÄYTÖN OHJAUS

Paperikoneen linjakäyttö koostuu nopeussäädetyistä moottoreista, jotka pyörittävät paperirataa kuljettavia teloja. Kun paperirata kulkee koneen läpi, sen ominaisuudet muuttuvat. Tämä aiheuttaa rainaan venymää, minkä johdosta rainan nopeus vaihtelee koneenosasta riippuen. Muun muassa tästä syystä koneen kaikki telat eivät voi pyöriä samalla kehänopeudella, vaan niitä on ohjattava erikseen. (Romppainen 2002, 39, 43.)

2.1 Linjakäyttöjärjestelmän rakenne

Linjakäyttöjärjestelmää operoidaan ja sen toimintaa seurataan valvomon operointinäytöiltä sekä koneen vierellä sijaitsevista ohjauspulpeteista. Valvomojärjestelmä on ohjelmisto operointiasemalla, joka on liitetty sähkökäyttöjä ohjaavaan ohjausjärjestelmään. Ohjausjärjestelmä on käytännössä tietokone (prosessiasema), jossa käyttöjen ohjaussovellus sijaitsee. (Valmet DNA Drive Controls 2016.)



KUVA 1. PMC800-linjakäyttöjärjestelmä (KnowPap 2017)

Paperikoneen sähkökäyttö koostuu taajuusmuuttajasta ja telaa pyörittävästä sähkömoottorista (ABB 2001). Kuvan 1 PMC800-linjakäyttöjärjestelmässä AC-syöttöinen syöttöryhmä, tasavirtakiskosto ja invertteri yhdessä muodostavat taajuusmuuttajan. Syöttöryhmä tasasuuntaa AC-syöttövirran yhteiseen tasavirtakiskostoon, josta invertterit vaihtosuuntaa sen halutulla taajuudella AC-moottoreille. (KnowPap 2017.)

Taajuusmuuttaja liikennöi ohjausjärjestelmän kanssa, ja säätää ohjaussignaalien perusteella mm. sähkömoottorin pyörimisnopeutta. Samalla ohjausjärjestelmä lukee tietoa käytön tilasta. Linkki prosessiaseman ja käyttöjen välillä toteutetaan nykyään lähes poikkeuksetta kenttäväyläratkaisuilla (esim. Profibus ja Profinet) (Valmet DNA Drive Controls 2016).

2.2 Ohjausjärjestelmät

Ohjausjärjestelmä huolehtii mm. sähkökäyttöjen ajotoimintatiloista ja antaa käytöille pysäytys- ja käynnistyskomennot. Se pitää sisällään nopeusohjearvoketjun, joka laskee oikeat asetusarvot nopeudelle ja momentille linjakäytön kaikille yksittäisille käytöille (KnowPap 2017). Eri koneenosien välistä nopeuseroa ja yhden koneenosan eri käyttöjen välistä kuormituksen jakoa voidaan ohjata ope-
rointinäytöiltä (Romppainen 2002, 43).

Ajotoimintatiloista tyypillisimpiä ovat ajo, ryömintä ja nykäysajo. Ajotilassa käyttöjä ohjataan koneen tuotantonopeudella. Ryömintätilaa käytetään, kun konetta valmistellaan tuotantoajoon. Nykäysajotilassa käyttöjä voidaan ajaa jaksottaisesti esimerkiksi huopien ja viirojen vaihdon aikana. (KnowPap 2017.)

Paperikoneen sähkökäyttöjen tavallisimmat säätötavat ovat nopeus-, momentti- ja kireyssäätö. Nopeussäädetyssä käytössä telan pyörimisnopeus pyritään pitämään asetusarvossa, joka tulee ohjausjärjestelmästä. Pyörimisnopeuden takaisinkytkentä saadaan usein kierroslukumittarilta eli pulssianturilta. Momenttisäädön periaatteena on, että sähkökäytön pyörittämän telan momentti pyritään pitämään asetusarvossa, joka saadaan ohjausjärjestelmästä tai toiselta käytöltä

(KnowPap 2017). Esimerkiksi keskikäyttöisessä kiinnirullaimessa ei voida käyttää suoraan nopeussäätöistä käyttöä, sillä sylinterin halkaisija kasvaa paperin rullautuessa sen ympärille. Tämä tarkoittaisi kehänopeuden jatkuvaa kasvua, vaikka kierrosnopeus pysyisi vakiona. Momenttisäädöllä keskiövetoinen kiinnirullaussylinteri pitää paperin rullauskireyden vakiona pyörimisnopeutta säätämällä (Center Driven Reeling 2014). Kireyssäätö toimii kaskadisäätönä, jossa paperiradan kireyttä mitataan kireysmittarilla, ja kireys pyritään pitämään asetusarvossa. Kireyssäädin antaa asetusarvon joko nopeus- tai momenttisäätimelle (Kaijankoski 2012, 15).

2.3 Käyttöliittymät

Käyttöjenohjausjärjestelmä voi toimia osana hajautettua prosessiohjausjärjestelmää, tai se voi olla oma erillinen järjestelmänsä. Tavallisesti, kummassakin tapauksessa käyttöjen operointi tapahtuu prosessinohjausten kanssa samasta valvomojärjestelmästä. Joissain tapauksissa voidaan kuitenkin asentaa erillinen valvomojärjestelmä sähkökäyttöjen ohjausta varten. Näissä molemmissa ratkaisutavoissa operaattorilla on samat operointimahdollisuudet ja sama informaatio saatavilla, mutta valvomonäyttöjen ulkoasut ja käyttöliittymät voivat erota toisistaan. (KnowPap 2017.)

Sähkökäyttöille on monen tyyppisiä ohjausnäyttöjä ja niitä voidaan räätälöidä asiakkaan tarpeiden mukaan. Tyypillisimpiä ohjausnäyttöjä ovat ryhmänäytöt, joissa on esitetty yhden koneenosan käytöt ja niiden tärkeimmät operoinnit. Operaattori voi seurata ryhmänäytöistä samanaikaisesti yhden koneenosan käyttöjen tilaa ja oloarvoja, sekä mahdollisia häiriöitä. Koko koneen ajonseurantaa varten on yleiskatsausnäyttö, jossa voidaan esittää paperikoneen sivukuva ja kaikkien käyttöjen tila mahdollisuuksien mukaan. Yhteenvetonäyttö sisältää tietoa koneen nopeudesta, varoituksista ja hälytyksistä sekä lukituksista. Muita näyttöjä ovat mm. listanäytöt, joissa useampia käyttöjä on listattuna tärkeimpine tietoineen, tai profiilinäytöt, joissa esitetään pylväsdiagrammeihin tietoa esimerkiksi moottorien lämpötiloista. (Drives Operator Training 2008.)

3 VALMET DNA SINGLE WINDOW -SOVELLUS

Single Window -sovelluksella tarkoitetaan Valmet Automation Oy:n tuotetta, jolla mahdollistetaan paperikoneen kolmannen osapuolen toimittaman linjakäyttöjen ohjausjärjestelmän operoiminen Valmet DNA -operointinäyttöistä. Tällaiselle sovellukselle on tarvetta sellaisissa tilanteissa, joissa paperitehtaan prosessiohjausjärjestelmä on toteutettu Valmet DNA:lla ja linjakäyttöjen ohjaus toisella järjestelmällä, ja asiakas haluaa sekä prosessin että sähkökäyttöjen operoimisen hoituvan samoista valvomonäyttöistä. (Single Window Interface... 1998.)

Käyttöjenohjausjärjestelmän operoinnin integroiminen prosessinohjausnäyttöihin helpottaa operaattorin työtä, sillä siten kaikki valvomonäytöt ovat samalla periaatteella toteutettuja. Tämä poissulkee tarpeen opetella käyttämään useita eri valvomojärjestelmiä, mikä taas pienentää käyttökoulutukseen kuluvaan aikaa ja vähentää operointivirheiden riskiä. (Kaijankoski 2012, 45.)

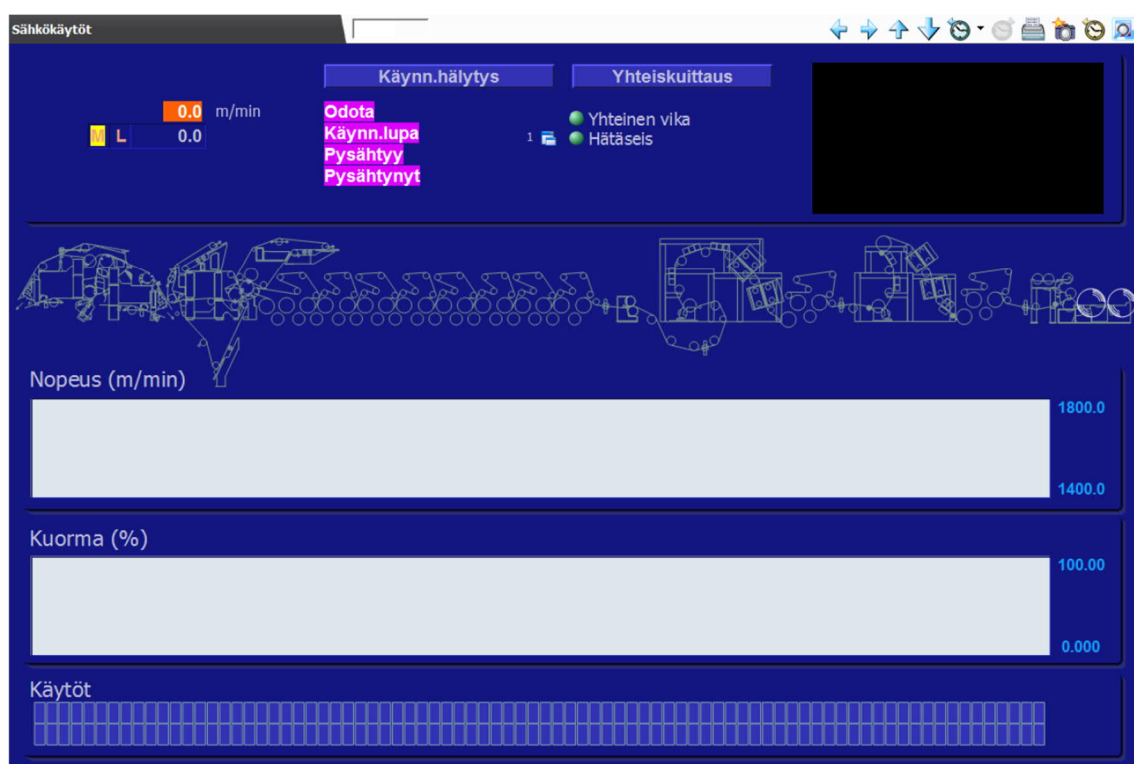
Taulukossa 1 on esitetty, minkä tyyppisiä signaaleita DNA Single Window -sovelluksen ja käyttöjenohjausjärjestelmän välillä kulkee. DNA-operointinäyttöiltä voidaan esimerkiksi ohjata paperikoneen nopeutta ja seurata teloja pyörittävien moottoreiden lämpötiloja. (Single Window Interface... 1998.)

TAULUKKO 1. Single Window -sovelluksen signaaleja

DNA → Käytöt	Käytöt → DNA
Koneen nopeuden ohjearvo	Koneen nopeuden mittausarvo
Nopeuseron ohjearvo	Nopeuseron mittausarvo
Kireyden ohjearvo	Kireyden mittausarvo
Kuormanjaon ohjearvo	Momentin mittausarvo
Momentin ohjearvo	Moottorin lämpötila
Käytön ohjaus (mm. pysäytys, ryömintä, ajo)	Käytön tila (mm. ryömintä, ajo, vika)
Ohjaustavan valinta (mm. nopeussäätö, kireyssäätö, momenttisäätö)	Käytön diagnostiikka (mm. vikasana, hälytyssana)

3.1 Operointinäytöt

Single Window -sovelluksen yleiskatsausnäyttö (kuva 2) on niin sanottu päänäyttö paperikoneen sähkökäyttöjen ohjauksille. Siitä on mahdollista antaa koneen käynnistyshälytys, kuitata viat kaikista käytöistä ja operoida koneen nopeutta. Yleiskatsausnäytössä on myös pylväsdiagrammit telojen pyörimisnopeudelle ja momentille, sekä indikaattorit käyttöjen lukituksille. Keskellä päänäyttöä esitetään yleensä sivukuva paperikoneesta, jotta alla olevista pylväsdiagrammeista tiettyä käyttöä tarkasteltaessa se on helpompi kohdistaa tietylle koneenosalle. Päänäytöstä käy myös ilmi, mikäli hätäpysäytys on aktiivisena. Hätäseis-indikaattorin vieressä sijaitsevasta painikkeesta saa avattua hätäseispainikelistan, josta selviää, mikä hätäseispainike on painettuna. (Drives Operator Training 2008.)



KUVA 2. Päänäyttö

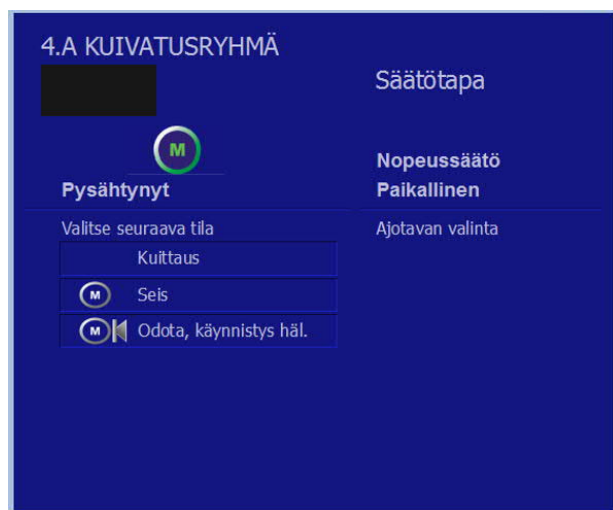
Single Window -sovelluksen ryhmänäytössä (kuva 3) esitetään sarakkeina tietyn koneenosan sähkökäytöt, joita mahtuu yhdelle ryhmänäytölle maksimissaan kahdeksan. Käytöille esitettäviä tietoja ovat telan kehänopeuden olo- ja ohjearvo, momentin oloarvo (kuorma), moottorin lämpötila, käytön tila sekä säätö- ja ajotapa. Säädetavasta riippuen, käytön nopeuserolle, kuormanjaolle tai kireydelle

voidaan asettaa ohjearvo ryhmänäytöltä. Koko paperikoneen ajonopeus ja sen asetusarvo esitetään jokaisen ryhmänäytön vasemmassa yläkulmassa. Ryhmänäytön alaosassa on symbolit käyttöryhmien sekä yhteiskäynnistysryhmien tilan seurannalle ja operoinnille.



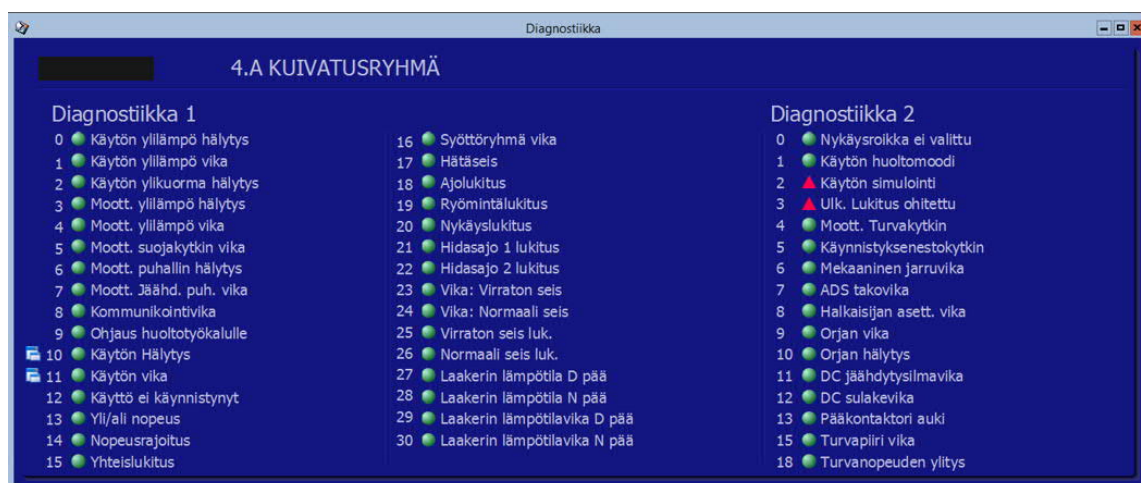
KUVA 3. Ryhmänäyttö

Käytön tilaa indikoivasta symbolista klikkaamalla aukeaa käytön operointi-ikkuna (kuva 4). Siitä voidaan tehdä käytölle määriteltäviä operointeja, kuten pysäyttää tai käynnistää käyttö. Käyttöryhmän ja yhteiskäynnistysryhmän operointia varten on omat vastaavat ikkunat.



KUVA 4. Operointi-ikkuna

Jokaisella käytöllä on ryhmänäytössä pikakuvake diagnostiikkaikkunaan (kuva 5), josta käy ilmi käytön mahdolliset viat, hälytykset ja käytön operoinnin estävät lukitukset. Mikäli käytöllä on vikoja, hälytyksiä tai lukituksia aktiivisena, ryhmänäytöllä diagnostiikkaikkunan pikakuvakkeen vieressä on punainen kolmio indikoimassa tätä. Käytön hälytys- ja vikasanoja pääsee tarkastelemaan diagnostiikkaikkunan vasemmassa laidassa sijaitsevista ikkunakuvakkeista.



KUVA 5. Diagnostiikkaikkuna

Datalista-näytössä (kuva 6) on maksimissaan 20 käyttöä listattuna allekkain. Kul-
lekin käytölle esitetään käytön tila, telan nopeuden ja momentin olo- ja ohjearvot
sekä moottorin lämpötila ja teho. Myös datalista-näytöstä pääsee käytön diag-
nostiikka ikkunaan rivin oikeassa laidassa sijaitsevasta painikkeesta.

Käyttöryhmäluettelo													
				Koneen nopeus		0.0	0.0	600.00	m/min				
				Nopeus		Momentas		Mootilämp.					
				Nopeus m/min	m/min	Nop.gro. %	as	Mom.en.ti %	°C	Teho kW			
PERÄNSYÖT.	PUMPPU M1	M	Valmis	0.0	0.0			0.0	25	0.0	▲	📄	
	PUMPPU M2	M	Valmis	0.0	0.0			0.0	25	0.0	▲	📄	
	SEKOITUSPUMPPU	M	Valmis	0.0	0.0			0.0	25	0.0	▲	📄	
VIIRAOSA													
SISÄVIIRAN	1.IMUTELA	M	Pysähtyy	478.3	0.0			-42.1	35	500	▲	📄	
	2.IMUTELA	M	Pysähtyy	483.9	0.0		5.1	-40.7	35	0.0	▲	📄	
	KIR.TELA	M	Pysähtyy	483.9	0.0		30.9	-40.7	35	0.0	▲	📄	
	ULKOVIRAN KIR.TELA	M	Pysähtyy	483.9	0.0		35.3	-40.7	35	0.0	▲	📄	
PURISTINOSA													
PIOK-UP	IMUTELA	M		0.0	0.0		0.0	0.0	0	0.0		📄	
	1. PURIST. SYM.TELA	M		0.0	0.0		0.0	0.0	0	0.0		📄	
	PURISTIMEN IMUTELA	M		0.0	0.0		0.0	0.0	0	0.0		📄	
	KESKITELA M1	M	Pysähtyy	478.3	0.0	0.00		-42.1	35	0.0	▲	📄	
	KESKITELA M2	M	Pysähtyy	483.9	0.0		0.0	-40.7	35	0.0	▲	📄	

KUVA 6. Datalista-näyttö

Sähkökäyttöjen dataa voidaan esittää myös profiilinäytöissä (kuva 7). Profiilinäytössä voi olla kaksi tai kolme profiilia kuvaamassa eri suureita pylväsdiagrammeina. Sähkökäytöt ovat vaakakselilla prosessijärjestyksessä. Tarkasteltavan sähkökäytön tunnus saadaan selville, kun kursori vieään kyseisen palkin päälle.



KUVA 7. Profiilinäyttö

3.2 Liityntärajapinta

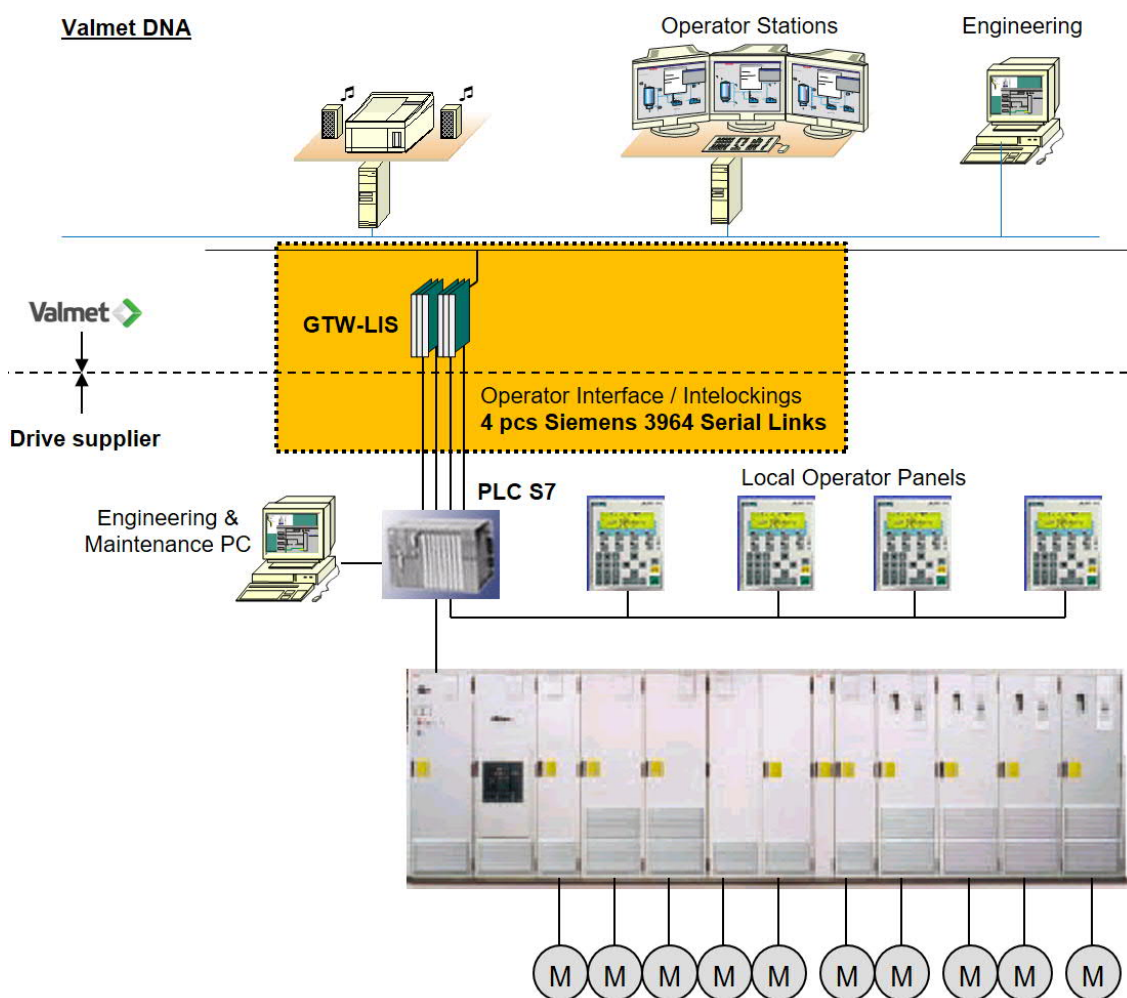
Liityntärajapintaa Single Window -sovelluksen ja kolmannen osapuolen käyttöjenohjausjärjestelmän välillä kutsutaan Single Window -linkiksi. Koska kyseessä on eri toimittajien kaksi eri järjestelmää, jotka eivät ole yhteensopivia keskenään, on niiden välisessä kommunikoinnissa käytettävä joko laiteajureita tai standardoitua tiedonsiirtomenetelmää. Käytettävä menetelmä riippuu mm. käyttöjenohjausjärjestelmän toimittajasta, käyttöjen lukumäärästä ja asiakkaan toiveista.

Mikäli paperikoneen koneautomaatio on toteutettu Valmet DNA:lla, tarvitaan Single Window -linkin lisäksi lukituslinkki käyttöjenohjausjärjestelmän ja koneohjausjärjestelmän välille. Lukituslinkissä kulkee lukitussignaaleja, jotka ovat esimerkiksi käynnistys- tai pysäytysehtoja ohjattaville koneille ja käytöille. Nämä lukitussignaalit ovat joissain tapauksissa samassa linkissä Single Window -linkin kanssa. Seuraavissa alaluvuissa esitellään erilaisia liityntäteknikoita, joita Single Window -linkin toteutuksessa on käytetty.

3.2.1 Siemens 3964R

Aikaisemmin 80- ja 90-luvulla Single Window -linkki toteutettiin sarjalinkeillä Siemensin 3964R-protokollalla, jolla pystyttiin liittymään myös useiden muiden toimittajien PLC-järjestelmiin. 3964R-protokollalla yksi sarjalinkki tarjoaa pääsyn maksimissaan 318 osoitteeseen. Koska Single Window -linkissä voi kulkea useita tuhansia signaaleita, sarjalinkkejä tarvittiin useita kerralla.

Kuvassa 8 on esitetty järjestelmäkaavio 3964R Single Window -linkin rakenteesta. Käyttöliittymäsovellus on ylätasolla kuvatulla operointiasemalla. Operointiasemalta liitytään LIS-liityntäaseman (Valmet DNA Logic Interface Server) kautta 3964R-sarjalinkkeihin, jotka ovat kytketty sähkökäyttöjä ohjaavaan PLC-järjestelmään. Sekä lukitussignaalit että Single Window -signaalit kulkevat 3964R-sarjalinkeissä. (Simatic 3964(R) RK512 Protocol... 2019, 1-4)

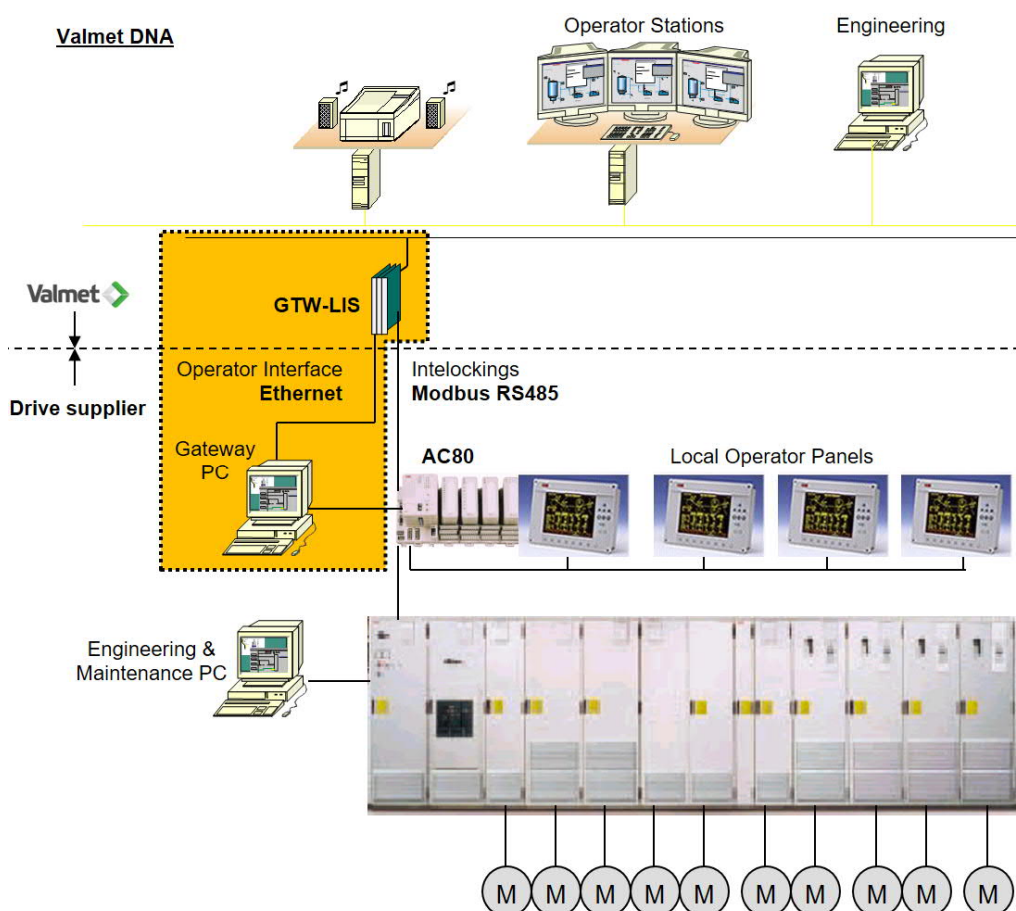


KUVA 8. Single Window -linkki 3964R-protokollalla, Siemensin käytöt (Single Window Interface... 1998, muokattu)

3.2.2 Modbus TCP

Myöhemmin Single Window -linkkiin alettiin käyttämään Modbus TCP -protokollaa. Modbus on vuonna 1979 julkaistu automaation tiedonsiirtoprotokolla, jossa voidaan käyttää isäntä/orja- (master/slave) tai asiakas/palvelin (client/server) -mallia. Single Window -linkissä käytetty Modbus TCP (v. 1999) on asiakas/palvelin -malliin perustuva tiedonsiirtomenetelmä, jossa liikennöinti tapahtuu TCP/IP Ethernet -yhteydellä. TCP/IP (Transmission Control Protocol/Internet Protocol) koostuu useista eri tiedonsiirtoprotokollista, joista itse nimessä esiintyvät TCP ja IP ovat niin kutsuttuja pääprotokollia.

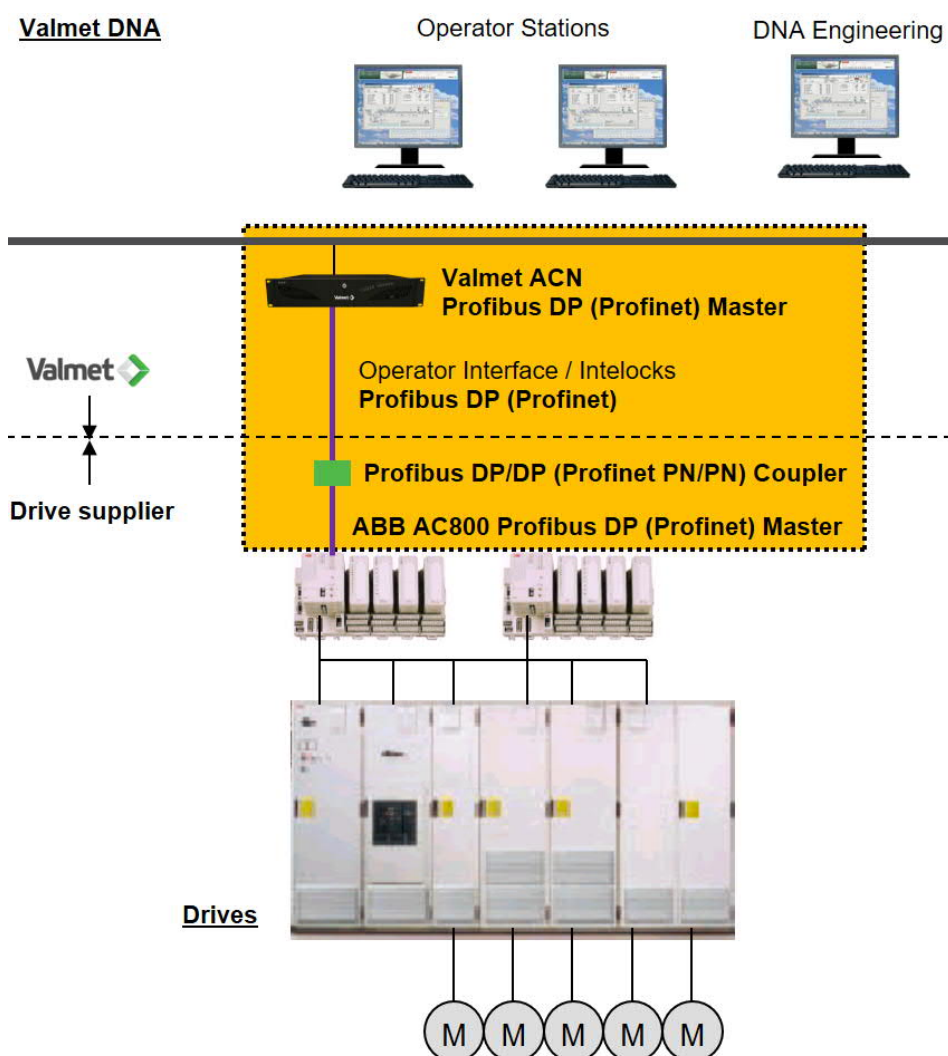
Modbus TCP Single Window -linkin rakenne on esitetty kuvassa 9. Modbus TCP -linkin kautta kulkevat vain Single Window -sovelluksen signaalit, ja lukituslinkki kone- ja käyttöjenohjauksilla on erikseen. LIS-liityntäasema voidaan konfiguroida master- tai slave-tilaan yksitellen kommunikointikanavakohtaisesti. (WAGO 2018; Modbus Protocol Interface Manual 2019.)



KUVA 9. Single Window -linkki Modbus TCP/IP -protokollalla, ABB:n käytöt (Single Window Interface... 1998, muokattu)

3.2.3 Profinet ja Profibus DP

Joissain tapauksissa, kun käyttöjen lukumäärä on tarpeeksi pieni, Single Window -linkki voidaan toteuttaa Profibus DP (Decentralized Peripherals) tai Profinet-väyläliitynnällä (kuva 10). Samassa väylässä voidaan siirtää myös kone- ja käyttöjenohjausten väliset lukitussignaalit. Profibus DP (DP) ja Profinet (PN) -linkkien kapasiteetti ei tavallisesti riitä käyttöliittymäsovelluksen signaaleille, sillä kytkettäessä kaksi DP- tai PN-isäntää yhteen, tarvitaan väliin DP/DP- tai PN/PN-väyläjakaja (coupler), jonka datamäärä on pieni. Datamäärää voidaan tosin kasvattaa lisäämällä väylään useampia väyläjakajia. (Valmet DNA Drive Controls 2016; Siemens 2014.)

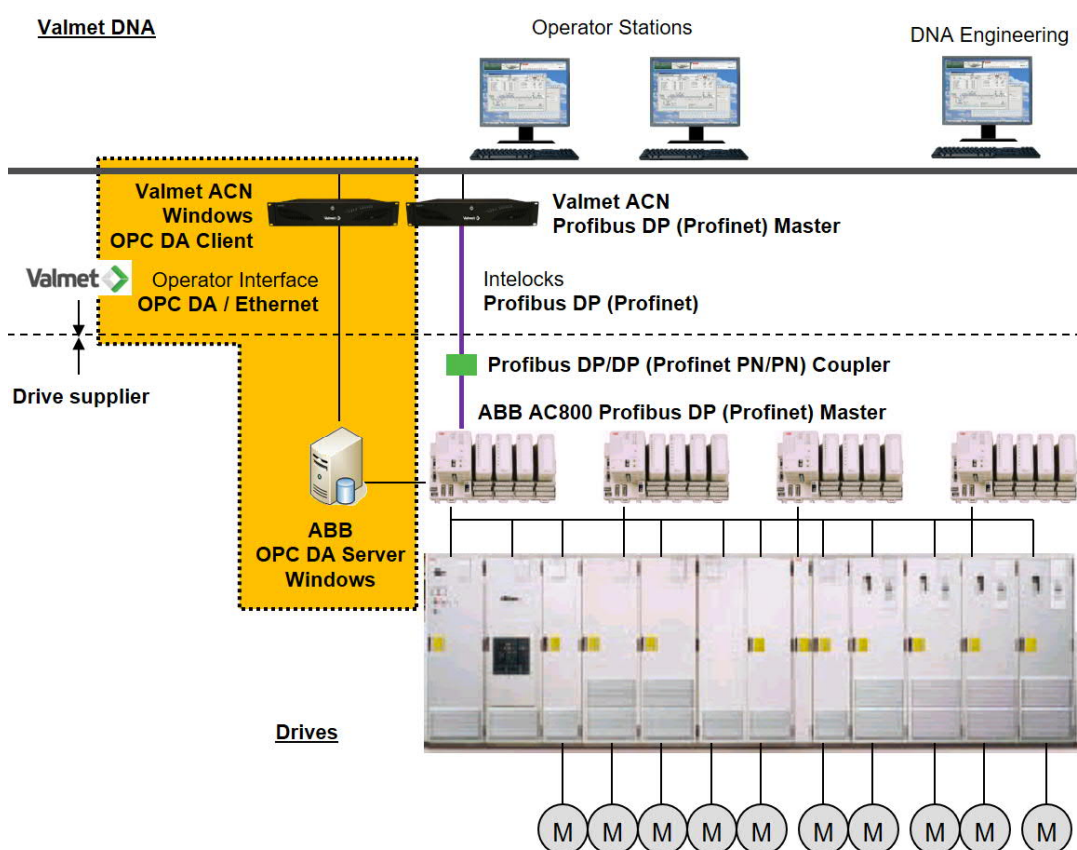


KUVA 10. Single Window -linkki ja lukituslinkki samassa DP- tai PN -linkissä, ABB:n käytöt (Valmet DNA Drive Controls 2016, muokattu)

3.2.4 OPC

Nykyisin yleisimmin käytetty Single Window -liityntäteknikka on OPC. Tässä osiossa esitellään OPC Single Window -linkin rakenne yleisellä tasolla, mutta OPC-määrittelyiden teoriaa käydään läpi seuraavassa osiossa tarkemmin.

Kun Single Window -linkki toteutetaan ensimmäisellä OPC-määrittelyllä, eli OPC DA:lla, järjestelmä on kuvan 11 mukainen. Linkki koostuu yleensä käyttöjenohjausjärjestelmän toimittajan OPC DA -palvelimesta ja Valmet DNA OPC DA -asiakasohjelmasta, jotka kummatkin ovat sovelluksia Windows-koneilla. Nämä sovellukset muuntavat sekä käyttöliittymä- että käyttöjenohjaussovelluksen datan yhtenäiseen muotoon. Tiedonsiirto palvelimen ja asiakasohjelman välillä tapahtuu Ethernet-liitynnällä. (Valmet DNA Drive Controls 2016)



KUVA 11. Single Window -linkki OPC DA -määrittelyllä, ABB:n käytöt (Valmet DNA Drive Controls 2016, muokattu)

Seuraavaksi tässä raportissa paneudutaan tarkemmin OPC-määrittelyihin ja erityisesti näistä uusimpaan OPC UA-määrittelyyn sekä sen käyttämiseen Single Window -linkin toteutuksessa.

4 OPC-TIEDONSIIRTOSTANDARDI

Teollisuuden automaatioissa ohjausjärjestelmät toteutetaan yhä useammin PC- ja sovelluspohjaisesti. Erityisesti Windows-tietokoneita käytetään alustoina käyttöliittymille ja ohjaussovelluksille. Automaatiojärjestelmien kehittyessä, ongelmaksi on muodostunut tiedonsiirto eri järjestelmien ja laitteiden välillä, kun tehtaassa voi olla käytössä lukuisia erilaisia väyläratkaisuja ja yhteyskäytäntöjä. (Damm, Leitner & Mahnke 2009, 1.)

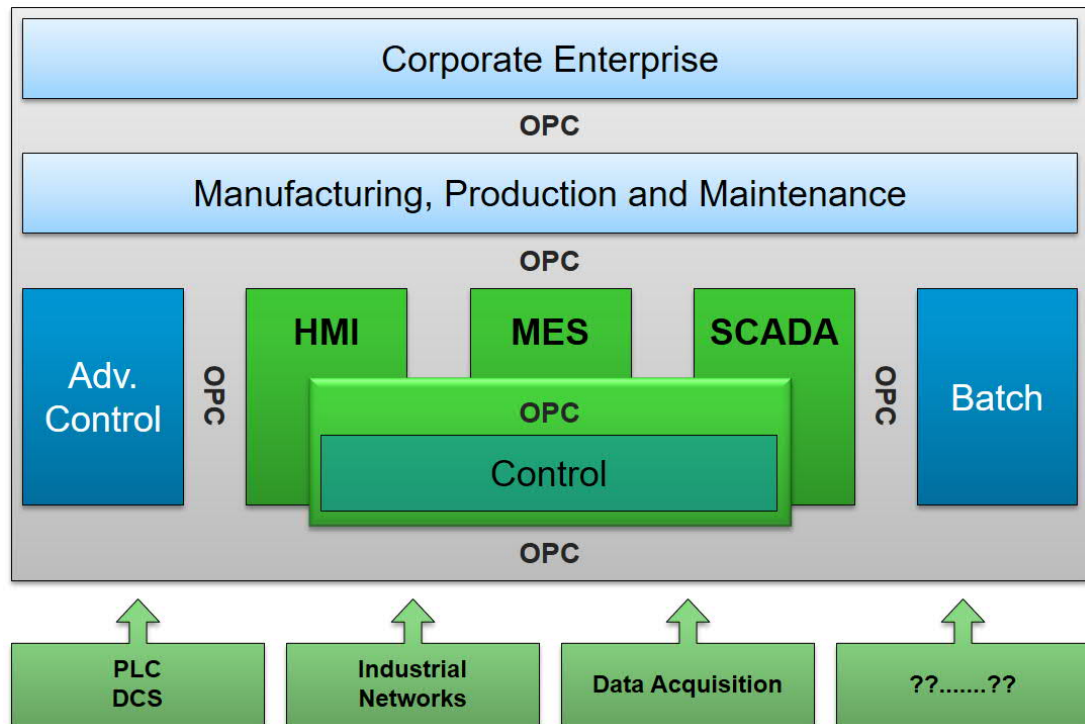
Useat käyttöliittymä- ja valvomosovelluksia toimittavat yritykset, kuten Fisher-Rosemount ja Rockwell Software, kohtasivat samoja ongelmia 90-luvun alkupuolella. Nämä yritykset perustivat työryhmän, jonka tavoitteena oli määrittää laitteajureille Plug and Play -standardi, joka tarjoaisi laitteille standardoidun pääsyn automaatiojärjestelmien dataan Windows-järjestelmissä. Tämän tuloksena, vuonna 1996, julkaistiin OPC Data Access (DA), ensimmäinen OPC-tiedonsiirtostandardi. (Damm, Leitner & Mahnke 2009, 1.)

OPC-tiedonsiirtostandardeja ylläpitää voittoa tavoittelematon OPC Foundation -säätiö (Damm, Leitner & Mahnke 2009, 1). Säätiöllä on yli 680 jäsenyritystä, joiden joukkoon kuuluu kaikki merkittävät automaatiojärjestelmien toimittajat ympäri maailman (OPC Foundation 2019).

4.1 OPC yleisesti

OPC-nimi on lyhenne sanoille *OLE for Process Control*, missä *OLE* tulee sanoista *Object Linking and Embedding* (OPC Foundation 2019). *OLE* viittaa Microsoftin kehittämään teknologiaan, jonka avulla dataa voidaan jakaa sovellusten välillä Windows-ympäristössä (Computer-Notes Inc 2014). OPC pohjautuu asiakas-palvelin-malliseen kommunikointiin (client/server). Yksi tai useampi OPC-palvelin kerää dataa prosessitietolähteiltä, kuten laitteilta, ja tuo datan saataville OPC-rajapintaan. OPC-asiakasohjelmat liittyvät OPC-palvelimeen ja saavat pääsyn sen sisältämään dataan. (Novotek 2015.)

OPC:n merkittävimpiä käyttötapauksia ovat SCADA- ja HMI-järjestelmät, sekä prosessidatan ja tapahtumien (events) saattaminen tehtaan johdon saataville, esimerkiksi ERP-ohjelmistoihin. Kuvalla 12 on havainnoitu OPC-dataintegraation monia käyttökohteita. (Damm, Leitner & Mahnke 2009, 3; Overview and Concepts 2017, 9-10.)



KUVA 12. OPC:n sovelluskohteita

Erilaisia OPC-määrittelyitä on useita, mutta tässä raportissa keskitytään vain työn kannalta olennaisiin määrittelyihin. Seuraavissa kappaleissa esitellään eri OPC-määrittelyitä ja sukupolvia, sekä tutkitaan niiden välisiä eroavaisuuksia.

4.2 OPC Classic

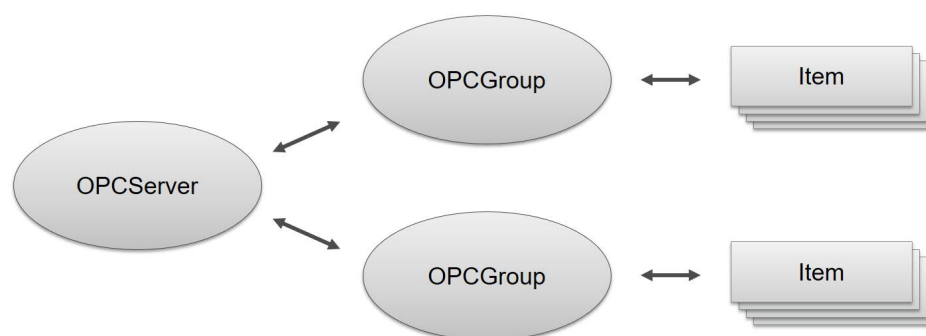
OPC Foundation on kehittänyt edellä mainittuihin tarpeisiin kolme pääasiallista OPC-määrittelyä: OPC DA, OPC Alarms & Events (A&E) ja OPC Historical Data Access (HDA). Pääsy reaaliaikaiseen prosessidataan on kuvailtu OPC DA -määrittelyssä, ja HDA-määrittely mahdollistaa pääsyn arkistoituihin historiadataan. A&E käsittää rajapinnan tapahtumapohjaiselle datalle, kuten hälytyksille. (Damm, Leitner & Mahnke 2009, 3.)

OPC Classic -määrittelyt käyttävät Microsoftin COM- ja DCOM-teknologiaa ohjelmakomponenttien väliseen tiedonsiirtoon (OPC Foundation 2019). Määrittelyjen kehitysvaiheessa tämän etuna oli se, että tarvittava teknologia oli valmiina kaikilla Windows-koneilla, mikä vähensi määrittelytarvetta ja sitä kautta kehitykseen kuluva aikaa. Tämä toisaalta aiheutti myös rajoituksia, joista suurimpana on OPC Classic -määrittelyiden Windows-alustariippuvuus. DCOM-teknologian käyttö rajoittaa myös etäkäyttömahdollisuuksia, ja sitä ei voida esimerkiksi käyttää Internet-kommunikointiin. Sen konfigurointi on haastavaa, mikä voi osaltaan johtaa myös tietoturvan laiminlyöntiin. (Damm, Leitner & Mahnke 2009, 4, 8; National Instruments 2019.)

OPC Data Access

OPC DA -rajapinta mahdollistaa prosessidataa sisältävien muuttujien lukemisen, kirjoittamisen ja monitoroinnin. Sen tyypillinen käyttötarkoitus on siirtää reaaliaikaista dataa PLC- ja DCS-järjestelmistä HMI- ja muille Client-näytöille esitettäväksi. (Damm, Leitner & Mahnke 2009, 4.)

Kun OPC Client haluaa käyttää OPC-palvelimen muuttujia (OPC item), se luo *OPCServer*-objektin (kuva 13). Sen avulla asiakasohjelma ottaa yhteyden palvelimeen ja saa metodit oikeiden muuttujien ja niiden ominaisuuksien löytämiseen osoiteavaruushierarkiasta. Muuttujan ominaisuuksia ovat mm. datatyyppi, käyttöoikeudet ja tiedon laatu. Asiakasohjelma ryhmittelee muuttujat niiden asetusten, kuten päivitysajan, perusteella *OPCGroup*-objekteihin. Tämän jälkeen muuttujat ovat asiakasohjelman luettavissa tai kirjoitettavissa. (Damm, Leitner & Mahnke 2009, 4–5.)



KUVA 13. OPC-asiakasohjelman luomat objektit tiedonsaantiin

OPC-asiakasohjelma voi lukea dataa palvelimelta joko sykleittäin, tai vain silloin, kun muuttujan arvo (value) muuttuu. Jälkimmäisessä tavassa asiakasohjelma monitoroi palvelimen arvoja tietyllä päivityssyklillä, ja palvelin lähettää muuttujien arvot takaisin vain, jos ne ovat muuttuneet edelliseen sykliin nähden. (Damm, Leitner & Mahnke 2009, 5.)

4.3 OPC Unified Architecture

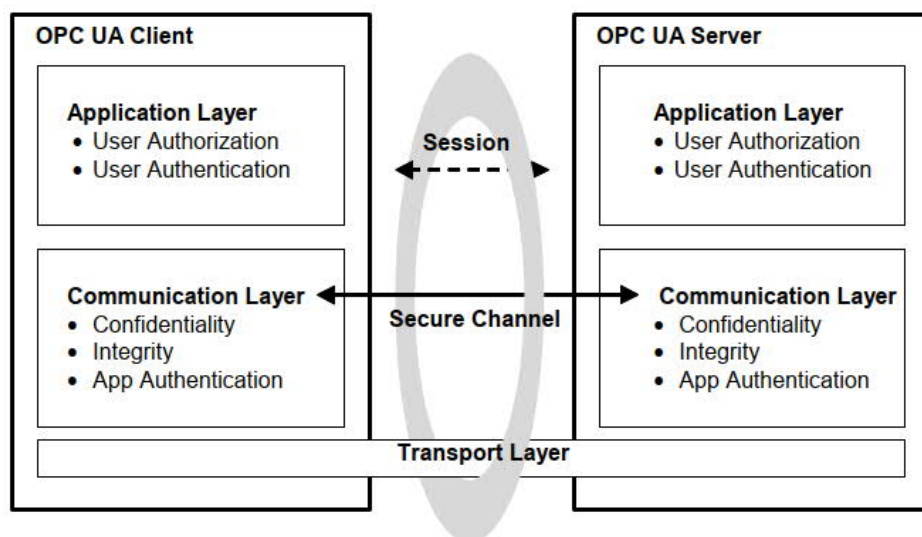
Ratkaisuna OPC Classic -määrittelyn rajoituksiin, OPC Foundation -jäsenyritysten vaatimusten pohjalta kehitettiin uusi OPC-määrittely, OPC Unified Architecture (UA). OPC UA julkaistiin vuonna 2008, noin 12 vuotta OPC Classic -määrittelyn jälkeen. Tärkeimpiä vaatimuksia uudelle määrittelylle olivat alustariippumattomuus, toiminnallinen vastaavuus OPC Classic -määrittelyjen kanssa, turvallisuus, laajennettavuus ja monipuolinen tietomallintaminen. (Damm, Leitner & Mahnke 2009, 1, 8-9; OPC Foundation 2019.)

4.3.1 Tietoturva ja kommunikointi

Industrial Internet -kehityksen johdosta, yhä suurempi osa tehtaan laitteista ja järjestelmistä on kytkettynä verkkoon, mikä korostaa automaatiojärjestelmien tietoturvan merkitystä. Tämä on näkynyt myös OPC UA -määrittelyn kehityksessä. Automaation tietoturvallisuuteen liittyviä seikkoja ovat mm. kahden eri järjestelmän välillä siirrettävän tiedon salaus ja allekirjoitus, sovellusten tunnistaminen (palvelin/asiakasohjelma), käyttäjien todentaminen ja valtuuttaminen, datan siirtäminen palomuurin läpi ja auditointi. (Hunkar 2014, 2; OPC Foundation 2018, 8-9.)

Kuvalla 14 on havainnoitu OPC UA:n tietoturva-arkkitehtuuria. Normaali kommunikointi palvelimen ja asiakasohjelman välillä, kuten tiedon ja asetusten välittäminen, tapahtuu sovellustasolla (Application Layer). Sovellustasolla huolehditaan myös käyttäjän todentamisesta ja valtuuttamisesta. Sovellustason data kulkee kommunikointitasolle (Communication Layer) käsiteltäväksi, ja kaikki sovellustason kommunikointi (Session) tapahtuu kommunikointitason kanavalla (Secure

Channel). Kommunikointitason kanava huolehtii datan salauksesta luottamuksellisuuden ylläpitämiseksi, allekirjoituksesta tiedon eheyden säilyttämiseksi ja digitaalisista sertifikaateista sovelluksen todentamiseksi. Kommunikointitason kanavan toimittaman datan lähetyks ja vastaanotto tapahtuu liikennöintitasolla (Transport Layer). (OPC Foundation 2018, 13-15.)



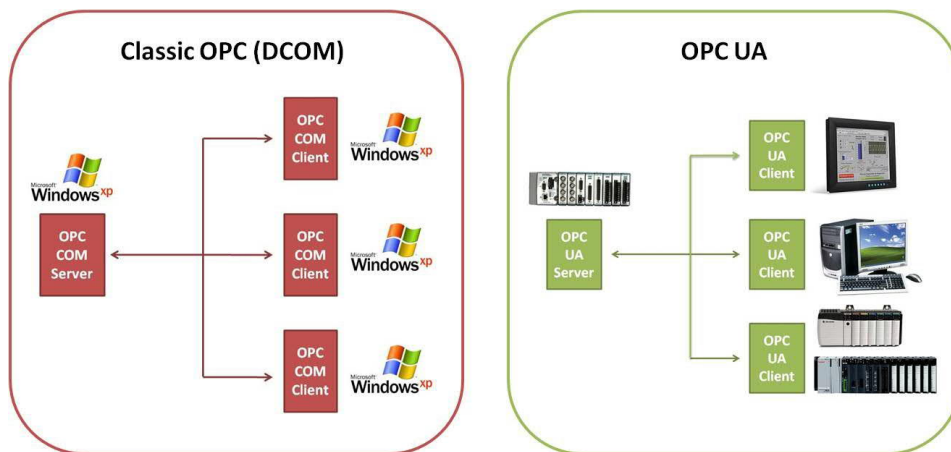
KUVA 14. OPC UA tietoturva-arkkitehtuuri (OPC Foundation 2018)

OPC UA:n liikennöintitasolle on määritelty kaksi siirtoyhteysprotokollaa: OPC UA TCP ja OPC UA HTTPS. OPC UA TCP on optimoitu (vähän resursseja vaativa) binääriprotokolla, jossa tiedonsiirto tapahtuu TCP/IP:llä. OPC UA HTTPS on www-sovelluspalveluprotokolla (web service), jossa HTTP-viestit salataan TLS-salausprotokollalla ennen lähettämistä. OPC UA -palvelimen URL-osoitteen etuliite on siirtoyhteysprotokollasta riippuen joko *opc.tcp://* (OPC UA TCP) tai *https://* (OPC UA HTTPS). (OPC Foundation 2017c, 60; National Instruments 2019.)

4.3.2 Alustariippumattomuus

OPC Unified Architecture avaa ovet pitkään toivotulle mahdollisuudelle käyttää ohjausjärjestelmissä Linux-käyttöjärjestelmää Windowsin sijaan, kun aiemmin OPC Classic -määrittelyjen vahva läsnäolo on sitä rajoittanut. Linuxia pidetään Windowsia parempana alustana teollisuuden ohjausjärjestelmiin erityisesti sen stabiilisuuden johdosta. Uuden päivityksen asentaminen Linux-tietokoneeseen

onnistuu usein ilman tarvetta uudelleenkäynnistykselle, toisin kuin Windows-koneeseen. Windowsin päivitykset ovat taakka monelle tuotantolinjalle, kun tuotanto joudutaan usein pysäyttämään esimerkiksi prosessiaseman käyttöjärjestelmäpäivityksen vuoksi. Linux-pohjaisella ohjausjärjestelmällä tuotantolinjan seisonta-aika on pienempi, eli tuotantotaso on parempi. DCOM-riippuvuudesta luopuminen mahdollistaa suoran kommunikoinnin mm. sulautettujen järjestelmien sekä ohjelmoitavien logiikoiden kanssa riippumatta käyttöjärjestelmästä (kuva 15).



KUVA 15. Alustariippumaton kommunikointi (National Instruments 2019)

Merkittävänä etuna Linuxin käytössä on myös pienemmät kustannukset. Linux on avoimen lähdekoodin käyttöjärjestelmä, joten se on saatavilla ilmaiseksi. Tämän lisäksi myös Linuxin ylläpito on edullisempaa, sillä tuotantoyhtiöiden ei tarvitse maksaa Windows-palvelimien lisensseistä ja ylläpito vie vähemmän aikaa. Windows-järjestelmän katsotaan myös olevan yleisesti Linuxia herkempi toimintahäiriöille. (Duty 2010.)

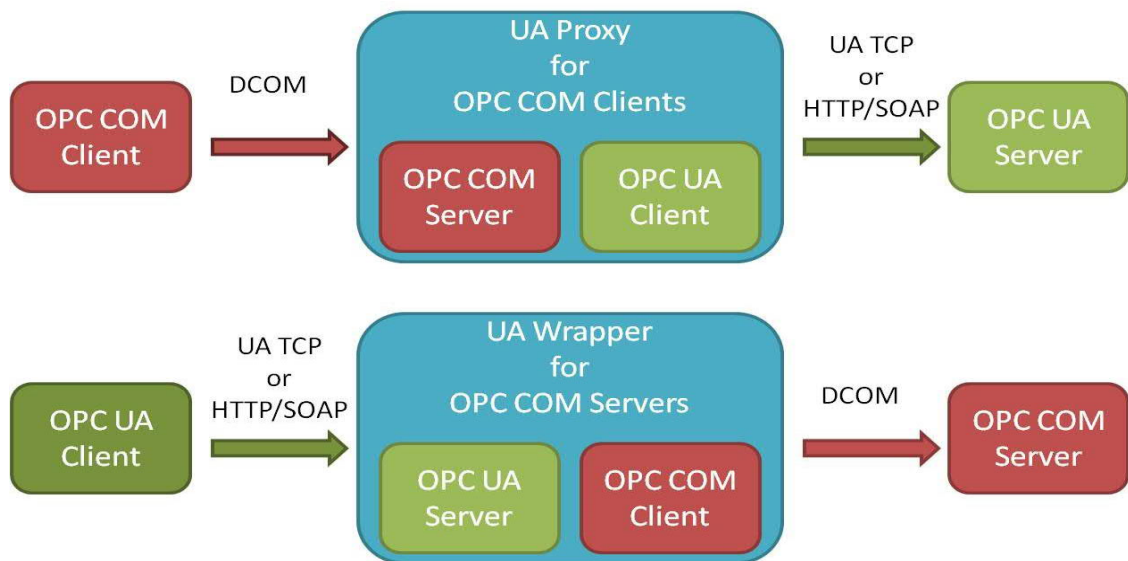
4.3.3 Tietomallintaminen

Tietomallilla tarkoitetaan organisatorista kehystä, jolla järjestelmän (tai useamman järjestelmän) tietoresurssit määritellään, karakterisoidaan ja identifioidaan (OPC Foundation 2017a, 3). Tietomallintaminen on yksi OPC UA -määrittelyn perusperiaatteista. OPC Classic -määrittely tarjoaa vain puhdasta dataa. Esimerkiksi lämpötilaa mitattaessa, saatavilla olevia tietoja ovat vain muuttujan nimi, mi-

tattu lämpötila ja jotkin perustiedot, kuten mitatun lämpötilan yksikkö. OPC Classic -määrittelyn tarjoamien tietojen lisäksi OPC UA voi esimerkiksi tarjota tietoa lämpötila-anturin tyypistä ja jakaa tyyppitietoa muille asiakasohjelmille. Asiakasohjelmat kykenevät näin suorittamaan kehittyneempiä tehtäviä tulkiten OPC-serverin tarjoaman datan semantiikkaa. (Damm, Leitner & Mahnke 2009, 19; OPC Foundation 2017b, 9; Novotek 2015.)

4.3.4 Yhteensopivuus

Tietoliikenneteknologian vaihtumisen myötä OPC UA ei ole suoraan yhteensopiva OPC Classic -määrittelyjen kanssa. OPC DA -palvelimet vaativat wrapper-ohjelman UA-asiakasohjelmien käyttämiseen, ja UA-palvelimiin pääsy edellyttää DA-asiakasohjelmalta UA-välityspalvelimen (proxy) käyttöä (kuva 16). (National Instruments 2019.)

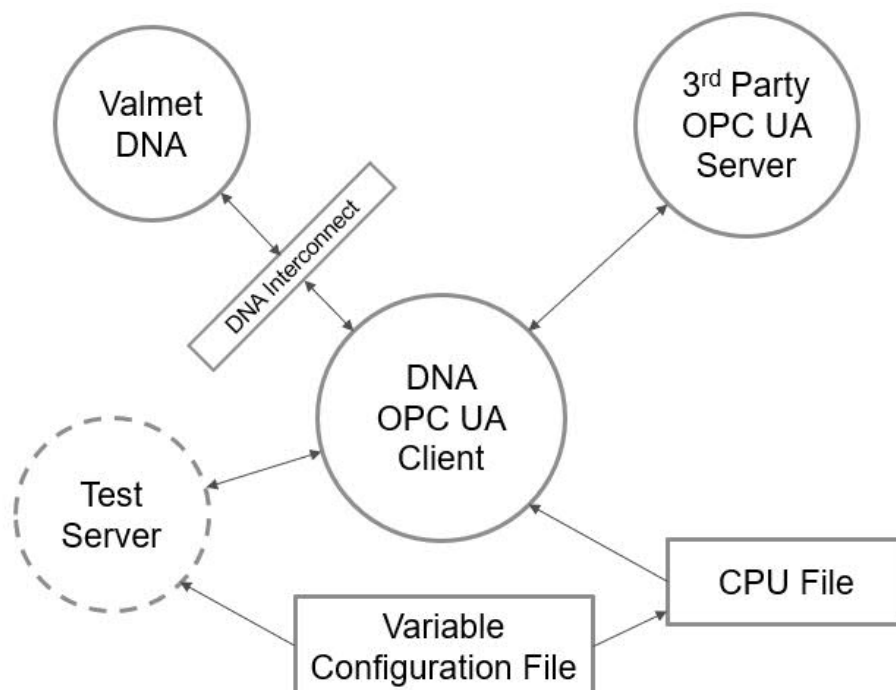


KUVA 16. OPC UA ja OPC Classic -määrittelyt eivät ole suoraan yhteensopivia (National Instruments 2019, muokattu)

Markkinoilla on useita yhdyskäytäväsovelluksia, joiden avulla COM/DCOM-pohjaiset OPC Classic -asiakasohjelmat ja -palvelimet voidaan muuntaa OPC UA -määrittelyn kanssa yhteensopiviksi.

5 VALMET DNA OPC UA -ASIAKASOHJELMA

Kun Valmet DNA toimii OPC UA -asiakasohjelmana, DNA OPC UA Client -sovel-
lus toimii yhdyskäytävänä DNA-automaatiojärjestelmän ja minkä tahansa kol-
mannen osapuolen OPC UA -palvelimen välillä. Järjestelmän rakenne on esitetty
kuvassa 17. (DNA OPC UA User's Guide 2018, 30.)

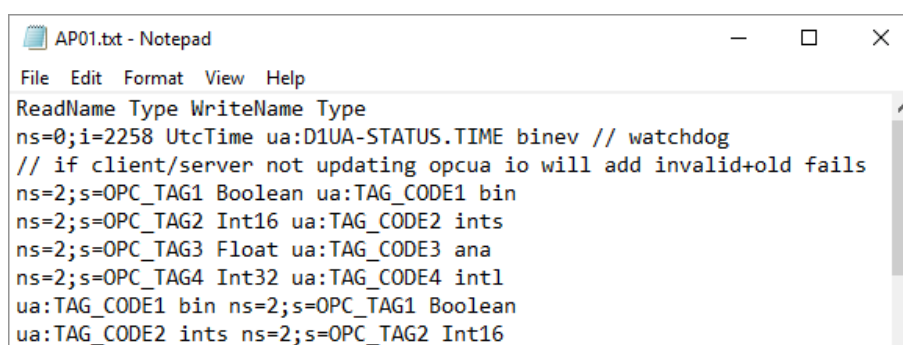


KUVA 17. Yleinen DNA OPC UA Client -järjestelmäkaavio

DNA OPC UA Client on sovellus Windows- tai Linux-tietokoneelle (Karaila 2018). Se keskustelee DNA-automaatiojärjestelmän kanssa DNA Interconnect -yhdys-
käytäväsovelluksen avulla. DNA OPC UA Client -sovelluksen testaukseen voi-
daan käyttää simulointipalvelinta (Test Server), johon voidaan määrittää erilaisia
muuttujia ja simuloida niiden arvoja. Asiakasohjelmalla näitä muuttujia voidaan
lukea ja kirjoittaa. Muuttujien konfigurointitiedosto (Variable Configuration File) ja
asiakasohjelman yleinen konfigurointitiedosto (CPU File) ovat tekstimuotoisia tie-
dostoja, jotka määrittävät asiakasohjelman ja sen käyttämien muuttujien konfigu-
raation. Konfiguroinnissa, eli edellä mainittujen tekstitiedostojen luomisessa voi-
daan hyödyntää erilaisia työkaluja. (DNA OPC UA User's Guide 2018, 30-33.)

5.1 Muuttujien konfigurointi (Variable Configuration File)

DNA OPC UA Client -sovelluksen ja OPC UA -palvelimen muuttujat kytketään toisiinsa tekstitiedostolla (Variable Configuration File), joka sisältää tarvittavat tiedot muuttujan määrittelystä ja siitä, mistä muuttujan data haetaan. Tekstitiedoston jokainen rivi muodostaa luku-kirjoitus-parin siten, että ensimmäisenä on muuttuja, joka luetaan, ja seuraavana muuttuja, johon kirjoitetaan (kuva 18). (DNA OPC UA User's Guide 2018, 31-32.)



KUVA 18. Variable Configuration File -esimerkki

Kuvan 18 esimerkkitiedostossa *ua:TAG_CODE1*-muotoiset nimet ovat DNA-tageja ja *OPC_TAG1*-muotoiset nimet OPC UA -palvelimen muuttujien nimiä. Nimiä lisäksi jokaiselle muuttujalle tulee määrittää sen datatyyppi kuvan 18 mukaisesti siinä muodossa, jossa se kyseisessä järjestelmässä esitetään (esim. to-
tuusarvomuuttujan datatyyppi on DNA:ssa *bin* ja OPC UA:ssa *Boolean*). Lisäksi OPC-muuttujan määrittely vaatii indeksinumeron nimiavaruudelle (Namespace), jossa kyseinen muuttuja sijaitsee palvelimella. Nimiavaruuden indeksinumero määritellään kuvan 18 mukaisesti, esimerkiksi *ns=0*, ja muuttujan datatyyppi (*i* eli *Integer* tai *s* eli *String*) samalla tavalla tämän perään puolipilkulla erotettuna, esimerkiksi *i=2258*. (DNA OPC UA User's Guide 2018, 31-32.)

Tekstitiedoston (kuva 18) kaksi ylintä riviä tulee löytyä muuttujien konfigurointitiedostosta aina. Toisella rivillä määritetään muuttuja diagnostiikkatiedolle, jota luetaan valvontapiirillä DNA-sovelluksessa. Kyseinen valvontapiiri tulee olla kaikissa DNA-automaatiojärjestelmän sovelluksissa, jotka liikennöivät DNA OPC UA Client -ohjelman kanssa. Diagnostiikkatieto on aikaleimattu binäärisignaali (*binev*), jonka avulla valvotaan kommunikointivirheitä. (DNA OPC UA User's Guide 2018, 31-32.)

5.2 DNA OPC UA Client -sovelluksen konfigurointi (CPU File)

Yleiset konfigurointiparametrit DNA OPC UA Client -ohjelmalle määritellään CPU-tiedostossa (kuva 19), joka on tiedostomuodoltaan JSON (esim. AU01_cpu.json). CPU-tiedostossa määritellään mm. osoite ja nimi palvelimelle, johon Client-ohjelma halutaan liittää, erilaisia tietoturva asetuksia, sekä muita Client-ohjelman ja palvelimen väliseen kommunikointiin liittyviä asetuksia.



```

1 {
2   "clientName": "AU01",
3   "endpoint": "opc.tcp://10.10.10.10:12345",
4   "securityMode": "SignAndEncrypt",
5   "securityPolicy": "Basic256Sha256",
6   "doAlarmsAndEvents": false,
7   "lawFile": "/bu/dna_law.cfg",
8   "userName": "userName",
9   "password": "password",
10  "configFile": "/bu/AU01.txt",
11  "SERVER": "UASRV",
12  "logLevel": "warn",
13  "logFileMaxSize": 1000000,
14  "logFileMaxDays": "5",
15  "dnaWriteParams": "{ writeInterval: 1000, writeTimeout: 10000 }",
16  "dnaReadParams": "{ minTime: 500, maxTime: 20000 }",
17  "samplingInterval": 1000,
18  "queueSize": 10,
19  "mode": "cyclic",
20  "timerInterval": 1000,
21  "opcuaReadItems": 2000,
22  "opcuaSingleWrite": true,
23  "opcuaDelay": 100,
24  "doAlarmsAndEvents": false,
25  "AutoAcknowledge": false,
26  "SingleAcknowledge": true,
27  "AcknowledgeAll": "",
28  "PackageID": "AO01"
29 }

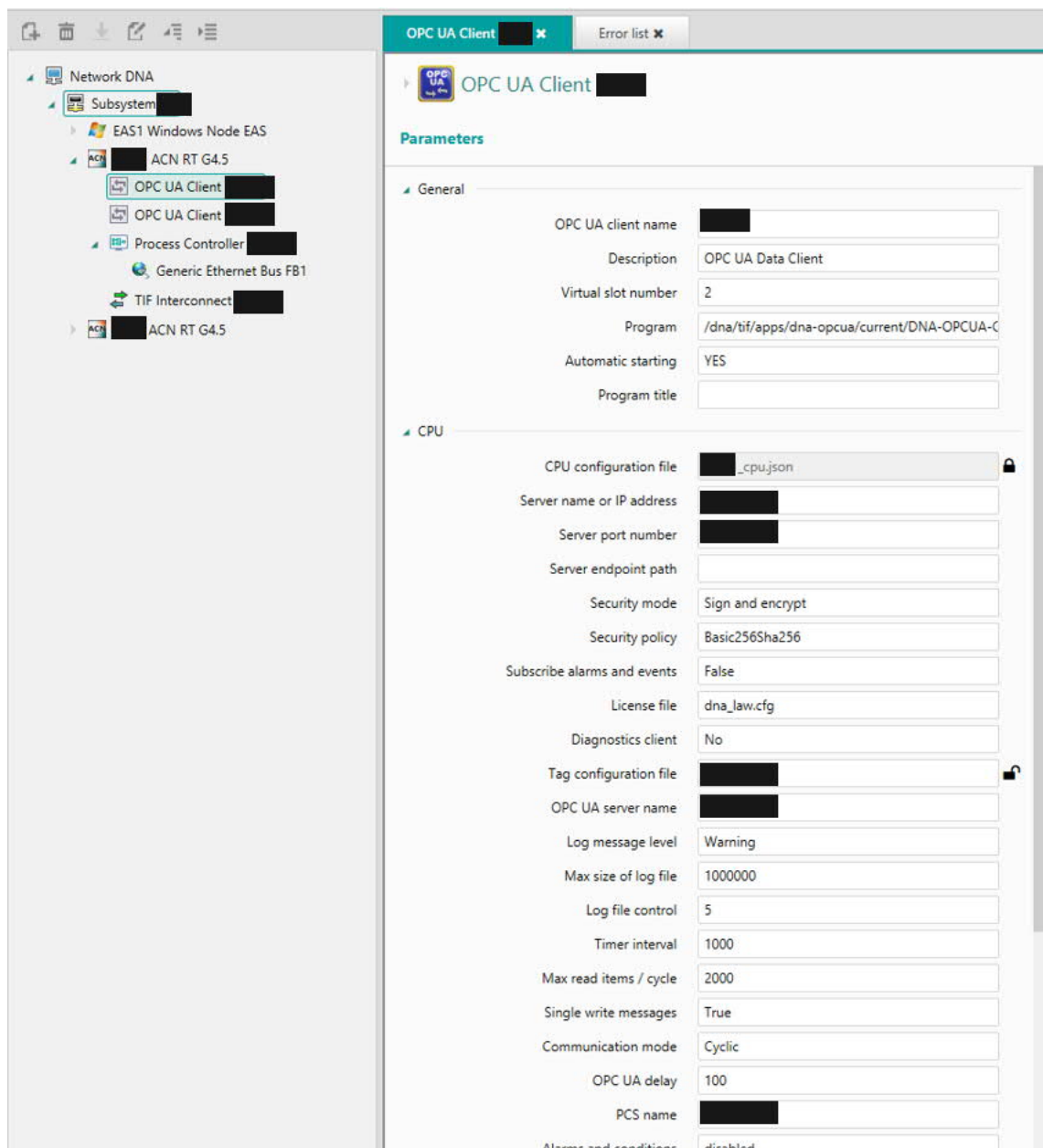
```

KUVA 19. DNA OPC UA Client -sovelluksen konfigurointitiedosto

Nämä tiedot määritellään DNA System Configurator -ohjelmassa (kuva 20), joka generoi JSON-tiedoston syötettyjen tietojen perusteella ja tallentaa sen varmennuspalvelimelle. Prosessiasema lukee konfiguraatitiedostot varmennuspalvelimelta käynnistyessään. Kuvassa 20 nähdään, millaisia asetuksia konfiguroinnissa voidaan määritellä. Luottamukselliset tiedot ovat peitetty kuvasta. Seuraavaksi esitellään olennaisimmat parametrit konfiguroinnin osalta.

OPC UA -palvelimen IP-osoite ja porttinumero syötetään CPU-osion toiselle ja kolmannelle riville. Muuttujien konfigurointitiedoston nimi määritellään kohdassa "Tag configuration file". Kommunikointitapa (Communication mode) voi olla joko syklinen (Cyclic) tai tilauspohjainen (Subscription). Syklisellä kommunikoinnilla tietoja luetaan palvelimelta sykleittäin, kun taas tilauspohjaisessa luetaan vain

muutoksesta. Kirjoitus palvelimelle tapahtuu kummassakin tapauksessa vain muutoksesta. Tilauspohjainen kommunikointi on prosessorille kevyempää ja sen avulla päästään teoreettisesti nopeampaan kommunikointiin. Tietoturva-asetuksissa (Security mode) vaihtoehtoja ovat "None", "Sign" ja "Sign and Encrypt". Ensimmäinen näistä tarkoittaa sitä, että tietoturva-asetukset ovat pois päältä. "Sign" puolestaan tarkoittaa, että Client-ohjelman tulee käyttää palvelimella hyväksyttyä käyttäjätunnusta yhteyden muodostamiseen. Tämän lisäksi "Sign and Encrypt" -tilassa linkissä kulkeva tieto kryptataan määriteltävissä olevalla salausalgoritmilla (Security policy). Tietoturva-asetukset tulee olla samalla tavalla määritettyjä sekä palvelimella että Client-ohjelmalla.



KUVA 20. DNA System Configurator, OPC UA Client -konfiguraatio

6 OPC UA SINGLE WINDOW -TOTEUTUS

Tämän työn tarkoituksena oli kehittää uusi generoitava Single Window -linjakäyttösovellustuote, jonka liittäminen kolmannen osapuolen käyttöjenohjausjärjestelmään voidaan toteuttaa OPC UA -protokollaa käyttäen. Tässä osiossa kerrotaan, mitä generoitavalla sovelluksella tarkoitetaan, esitellään sovelluksen tuotteistusprosessin kulku, kuvaillaan OPC UA Single Window -linkin rakenne, ja lopuksi käydään läpi, miten sovellustuotetta hyödynnettiin toimitusprojektin sovellussuunnittelussa.

6.1 Generoitavan Single Window -sovelluksen tuottaminen

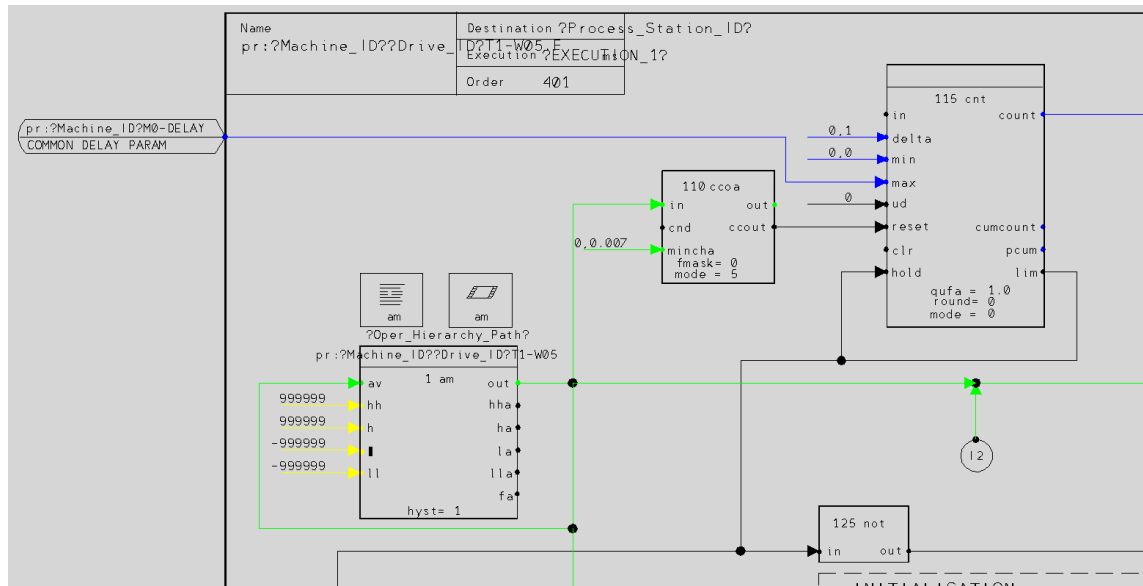
Sovelluksen generoinnilla tarkoitetaan sovelluksen tuottamista ennalta rakennetuista sovelluspohjista sovellussuunnittelijan määrittämiä sääntöjä soveltamalla. Kyseiset säännöt riippuvat kohdeprojektista ja ideana on, että säännöt ovat helposti määritettävissä generoinnin alustamiseen tarkoitetun työkalun avulla.

6.1.1 Generoinnin yleinen periaate

Kaikille generoitaville linjakäyttösovellustuotteille on luotu kantahakemisto, joka pitää sisällään sovellus- ja operointinäyttöpohjat, generointikoodin sekä Excel-määrittelytyökalun.

Sovelluspohjat (kuva 21) ovat DNA-suunnitteluympäristöön kuuluvalla Function Block CAD -ohjelmistolla suunniteltuja toimilohko-ohjelmia. Sovelluspohjien tekstit ovat joko staattisia tai dynaamisia. Staattiset tekstit pysyvät samana projektista riippumatta, mutta dynaamiset tekstit ovat koodattu erilaisilla muuttujilla, joiden perusteella niihin tuodaan projektikohtaista dataa sovellusta generoidessa. Nämä dynaamiset tekstiosat tunnistavat siitä, että ne alkavat ja päättyvät kysymysmerkkiin. Esimerkiksi kuvan 21 sovelluspohjassa teksti *?Machine_ID?* on dynaaminen muuttuja, joka kertoo paperikoneen konetunnuksen. Operointinäyttöpohjat (kuva 22) ovat puolestaan DNA Picture Designer -työkalulla piirrettyjä kuvia, jotka

määrittävät ohjausnäyttöjen käyttöliittymän ja ulkoasun. Näyttöpohjissa on myös sekä staattisia että dynaamisia tekstejä ja erilaisilla funktioilla varusteltavissa olevia symboleita. DNA-automaatiojärjestelmä on nimipohjainen, ja näyttöpohjat ovat linkitetty sovellukseen signaalien TAG-nimillä.



KUVA 21. Sovelluspohja

Operointinäyttöpohjissa lähes kaikki tekstit ovat dynaamisia, eli lähes kaikki tekstit korvautuvat Excel-työkaluun syötetyillä tiedoilla. Tämä mahdollistaa käyttöliittymän kielen valitsemisen sovelluksen generointivaiheessa. Yleisimmille kielille löytyy Excel-työkalusta ohjausnäyttöjen tekstien kielikäännökset, minkä avulla myös eri kieliset tekstit näyttöihin voidaan generoida automaattisesti.

The screenshot displays a template for a control display interface, organized into a grid of columns and rows. Each cell contains a label followed by a text input field. The labels are:

- Row 1:** ?Run_Speed_SP(14)?, ?Dr1_Name_L1(14)?, ?Dr2_Name_L1(14)?, ?Dr3_Name_L1(14)?, ?Dr4_Name_L1(14)?
- Row 2:** ?Speed_ME(14)?, ?Speed_ME_Unit?, ?Speed_SP(14)?, ?Speed_ME_Unit?, ?Speed_Diff_SP(14)?, ?Speed_Diff_SP_Unit?, ?Load_Share_SP(14)?, ?Torque_SP_Unit?
- Row 3:** ?Torque_ME(14)?, ?Torque_ME_Unit?, ?Temp_ME(14)?, ?Temp_ME_Unit?
- Row 4:** ?StTxt_Status(14)?, ?Opr_ReadyToSt, ?Opr_ReadyToSt, ?Opr_ReadyToSt, ?Opr_ReadyToSt
- Row 5:** ?StTxt_Control_Mode(14)?, ?Speed_On(14)?, ?Speed_On(14)?, ?Speed_On(14)?, ?Speed_On(14)?
- Row 6:** ?StTxt_Mode(14)?, ?Local(14)?, ?Local(14)?, ?Local(14)?, ?Local(14)?
- Row 7:** ?StTxt_Diagword(14)?

 Each input field contains the text '000000'. The columns are labeled at the top as ?Dr1_Name_L1(14)?, ?Dr2_Name_L1(14)?, ?Dr3_Name_L1(14)?, and ?Dr4_Name_L1(14)?, with additional labels like ?Dr1_Name_L2(14)?, ?Dr2_Name_L2(14)?, etc., below them.

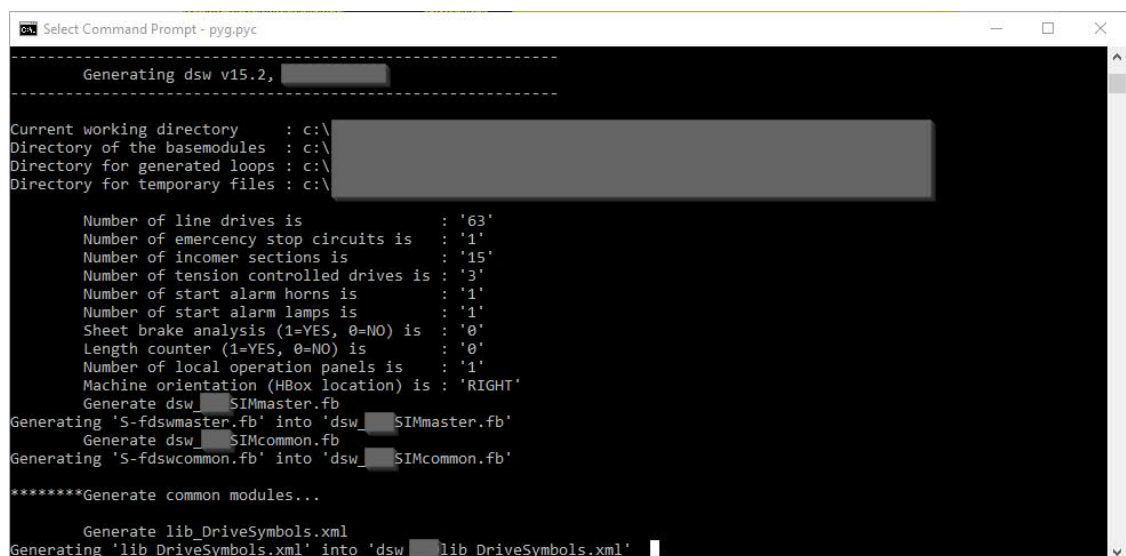
KUVA 22. Operointinäyttöpohja

Sovelluksen generointi alustetaan täyttämällä projektin ja siihen liittyvien käyttöjen tiedot Excel-määrittelytyökaluun. Tämän jälkeen luodaan VBA-makrojen avulla viisi korvaustiedostoa, joissa määrittelytyökaluun syötetty data esitetään yksinkertaistetussa muodossa. Generointi käynnistetään sille luodussa graafisessa käyttöliittymässä (kuva 23), jossa valitaan projekti, sovellustuote ja määrittelytyökalulla luodut korvaustiedostot. Pythonilla koodattu generointiohjelma siirtää dataa korvaustiedostoista asianmukaisesti sovellus- ja näyttöpohjiin.



KUVA 23. Sovellusgeneraattorin käyttöliittymä

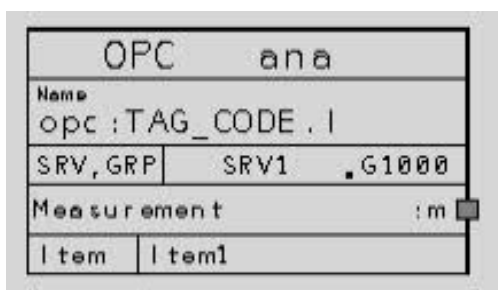
Generoinnin käynnistyttyä, sen edistymistä voidaan seurata komentoriviltä (kuva 24). Generointilokissa esitetään aluksi yhteenveto generoitavasta sovelluksesta, minkä jälkeen siitä näkee reaaliajassa, mitä moduulia generaattori kullakin hetkellä generoi. Komentorivillä esitetään myös generoinnin mahdolliset virheet.



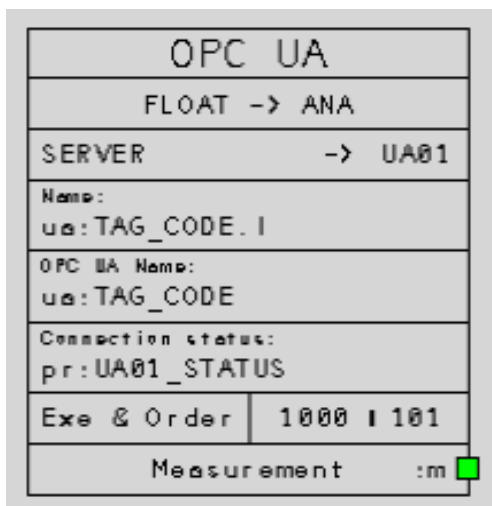
KUVA 24. Generointiloki komentorivillä

6.1.2 Sovelluspohjien tekeminen

OPC UA Single Window -sovelluspohjien lähtökohtana oli vanhat OPC DA -sovelluspohjat. Kaikki OPC I/O -lohkoja sisältävät piirit nimettiin uudelleen *_UA-*jälkiliitteellä ja vanhat DA I/O -lohkot (kuva 25) vaihdettiin vanhoista pohjista uusiin UA I/O -lohkoihin (kuva 26). Suurin osa UA I/O -lohkoista oli jo olemassa DNA-suunnitteluympäristössä. Puuttuvat lohkot määriteltiin, ja tuotekehitys-osasto valmisti ja lisäsi ne DNA-suunnitteluympäristön lohkokirjastoon. Single Window -sovelluksessa tarvittavat datatyytit ovat *UtcTime* (binev), *Boolean* (bin), *Float* (ana), 16-bittinen *Integer* (ints) ja 32-bittinen *Integer* (intl). DNA-esitustapa datatyypeille on esitetty sulkeissa.



KUVA 25. Vanha OPC DA analog input -lohko

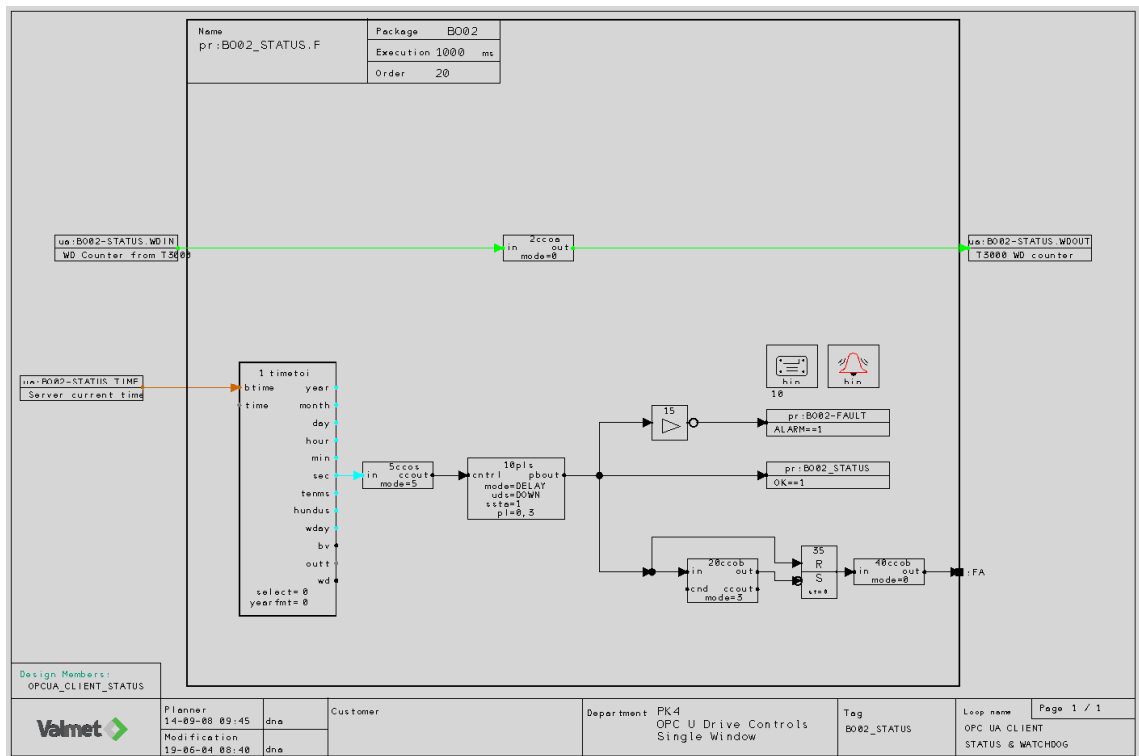


KUVA 26. Uusi OPC UA analog input -lohko

OPC UA I/O -lohkoihin määritettäviä tietoja ovat mm. I/O-lohkon nimi, signaalin OPC UA -nimi, valvontapiirin tunnus, OPC UA -palvelin sekä suoritusnumero. Signaalin OPC UA -nimeen viitataan tekstitiedostossa, jolla muuttujat konfiguroi-

daan. Tämän nimen tulee alkaa *ua:-*etuliitteellä, esimerkiksi *ua:TAG_CODE*. Jokaisella toimilohkolla DNA-suunnitteluympäristössä on oma suoritusjärjestysnumero, joka määrää lohkojen suoritusjärjestyksen piirin sisällä. Kahdella eri lohkolla ei voi olla samaa suoritusjärjestysnumeroa yhden piirin sisällä. Myös kaikille piireille tulee määrittää omat suoritusjärjestysnumeronsa.

Sovelluksen kantahakemistoon lisättiin myös edellä mainittu valvontapiiri (kuva 27). Sen dynaamiset tekstikentät korvattiin muuttujilla. Esimerkiksi tekstikenttään, joka kertoo piirin tunnuksen, lisättiin teksti *pr:?Process_Station_ID?_STATUS.F*, jossa *?Process_Station_ID?*-osa korvautuu Excel-määrittelytyökaluun täytetyllä prosessiaseman tunnuksella generointivaiheessa.

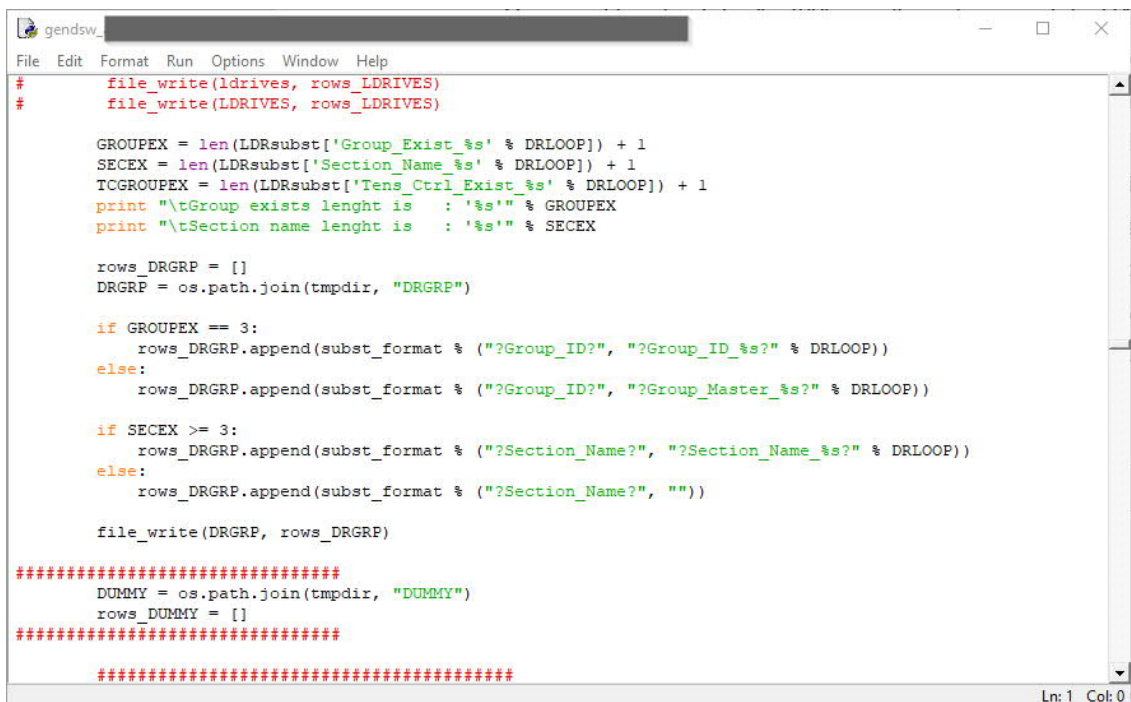


KUVA 27. OPC UA -valvontapiiri

Lisäksi vanhoja OPC DA -käyttöliittymäsovelluspohjia muokattiin muilla tavoin pyrkien tekemään niistä yksinkertaisempia, helpommin muokattavia ja yleisesti nykyaikaisempia. Operointinäyttöpohjat eivät vaatineet muutoksia, sillä näyttöjen ja sovelluksen välinen linkki on nimipohjainen, eikä näytöille luettavien DNA-muuttujien nimiin tehty muutoksia.

6.1.3 Muokkaukset generointikoodissa

Pythonilla koodattuun generointiskriptiin tuli päivittää sovelluspohjien uudet `_UA`-loppuiset tiedostonimet, jotta generaattori löytää uuden pohjan kantahakemistosta. Uuden valvontapiirin generointi lisättiin koodiin. Koodia muokattiin myös muilta osin liittyen muutoksiin, joita sovelluspohjiin tehtiin. Esimerkiksi sovelluksen konfigurointivaiheessa muutettiin tapaa, jolla määritetään erilaisten binäärisignaalien näkyvyys operointinäytöllä. Sovellusta muutettiin siten, että monessa eri sovelluspohjassa samanlailla määritellyt näkyvyysasettelut korvattiin kokonaan uudella piirillä, jossa signaalien näkyvyysasettelut tehdään vain kerran hyödyntäen hierarkkisia muuttujia. Tämä vaati muutoksia muuttujien arvojen kirjoitukseen generointikoodissa. Kuvassa 28 on esitetty pieni osa muokatusta koodista.



```

# file_write(ldrives, rows_LDRIVES)
# file_write(LDRIVES, rows_LDRIVES)

GROUPEX = len(LDRsubst['Group_Exist_%s' % DRLOOP]) + 1
SECEX = len(LDRsubst['Section_Name_%s' % DRLOOP]) + 1
TCGROUPEX = len(LDRsubst['Tens_Ctrl_Exist_%s' % DRLOOP]) + 1
print "\tGroup exists lenght is : '%s'" % GROUPEX
print "\tSection name lenght is : '%s'" % SECEX

rows_DRGRP = []
DRGRP = os.path.join(tmpdir, "DRGRP")

if GROUPEX == 3:
    rows_DRGRP.append(subst_format % ("?Group_ID?", "?Group_ID_%s?" % DRLOOP))
else:
    rows_DRGRP.append(subst_format % ("?Group_ID?", "?Group_Master_%s?" % DRLOOP))

if SECEX >= 3:
    rows_DRGRP.append(subst_format % ("?Section_Name?", "?Section_Name_%s?" % DRLOOP))
else:
    rows_DRGRP.append(subst_format % ("?Section_Name?", ""))

file_write(DRGRP, rows_DRGRP)

#####
DUMMY = os.path.join(tmpdir, "DUMMY")
rows_DUMMY = []
#####

#####

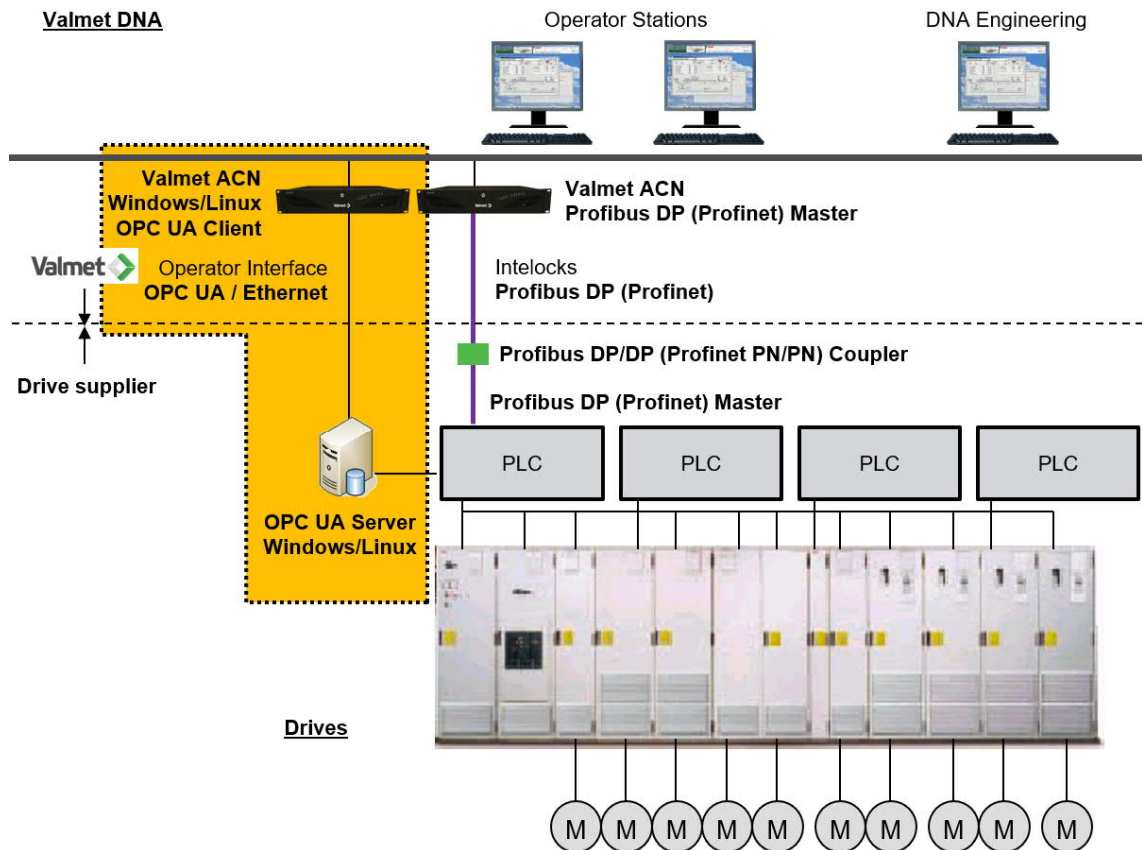
```

KUVA 28. OPC UA Single Window -sovelluksen generointikoodia

6.2 OPC UA Single Window -linkin järjestelmärakenne

OPC UA Single Window -linkki toteutetaan samalla periaatteella kuin OPC DA Single Window -linkki. Ainoa eroavaisuus järjestelmän puolesta on, että OPC UA

-linkkiä käytettäessä Valmet DNA OPC UA Client -sovellusta pyörittävä prosessiohjain ja käyttöjen toimittajan OPC UA -palvelin eivät ole Windows-riippuvaisia, vaan voivat käyttää myös esimerkiksi Linux-käyttöjärjestelmää. Kuvassa 29 on esitetty OPC UA Single Window -linkin yleinen järjestelmäkaavio.

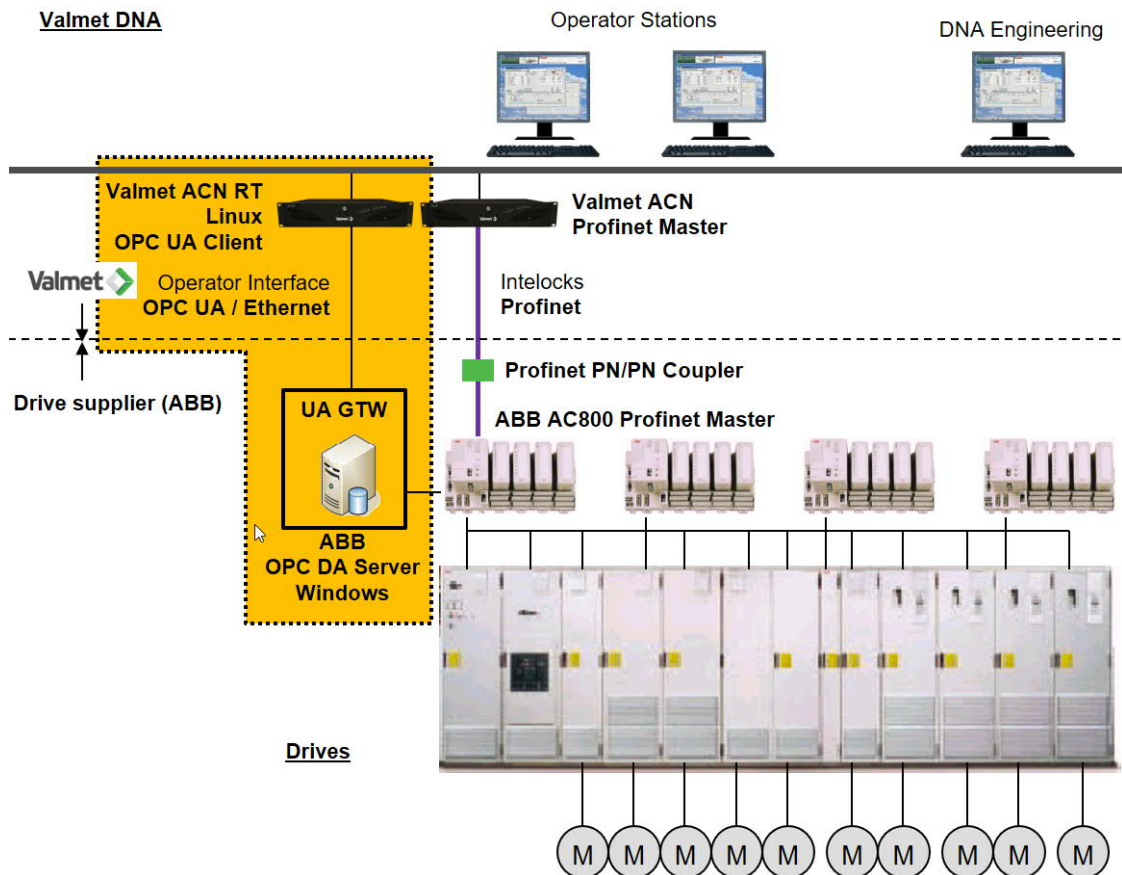


KUVA 29. OPC UA Single Window -linkin yleinen rakenne (Valmet DNA Drive Controls 2016, muokattu)

6.3 Tuotteistuksen soveltaminen asiakasprojektissa

Edellä kuvatun tuotteistuksen pohjalta suunniteltiin Single Window -sovellus LWC-paperikoneeseen osana toimitusprojektia. LWC-paperi (Light Weight Coated) on kevyesti päällystettyä ja lujaa paperia, jota käytetään mm. aikakauslehdissä ja myyntiluetteloissa sen hyvän kiillon ja vaaleuden vuoksi (KnowPap 2017). Paperikoneessa on Valmet DNA -prosessinohjausjärjestelmä ja ABB:n toimittama käyttöjenohjausjärjestelmä, joka ohjaa 82 sähkökäyttöä, jotka kaikki ovat tyypiltään ACS880. Linkki ABB:n järjestelmästä DNA-operointinäyttöihin oli aiemmin toteutettu Modbus TCP:llä, ja projektin tarkoituksena oli korvata tämä linkki OPC UA:lla käyttäen TCP/IP-kommunikointia.

Kyseisen projektin Single Window -linkin järjestelmäkaavio on esitetty kuvassa 30. ABB:n palvelin on toteutettu vielä OPC DA -protokollalla, mutta se on konvertoitu palvelimella sijaitsevalla Prosys UaGateway -sovelluksella UA:ksi, jolloin se käytännössä näkyy DNA OPC UA Client -ohjelmalle OPC UA -palvelimena. Luokituslinkki toteutettiin Profinet-väyläliitynnällä, mutta siihen ei tässä työssä ole tarkoitus perehtyä enempää.



KUVA 30. Projektin Single Window -linkin järjestelmäkaavio

Valmet DNA OPC UA -asiakasohjelma asennettiin Linux-pohjaiselle ACN RT -prosessiohjaimelle (Application and Control Node Real-Time). Samalla prosessiohjaimella oli myös prosessinohjauspalvelin, johon Single Window -sovelluksen operointinäytöt ja liitännäsovellus ladattiin. ACN RT -prosessiohjain on varustettu 4 gigatavun RAM-muistilla ja 2 gigatavun SD-muistikortilla, joka sisältää käyttöjärjestelmän. ACN RT:ssä ei ole kovalevyä. (ACN RT G4 Technical... 2019, 10.)

Sovelluksen generointi

Asiakkaalta saatiin lähtötietoina OPC-signaalimäärittely ja käyttöpisteluetelo. OPC-signaalimäärittelyssä oli listattuna kaikki OPC-palvelimella saatavissa olevat signaalit, joita oli noin 5400 kappaletta. Näistä vajaa puolet eli noin 2300 signaalia päätyivät DNA OPC UA Client -sovelluksen käyttöön. Määrittelystä kävi ilmi käyttöjen ryhmittely, ja mitä operointeja tietyille käytöille ja käyttöryhmille haluttiin. Käyttöpisteluetelossa oli myös tarvittavat tiedot mm. käyttöjen säätöta-voille ja skaalat eri suureille.

Näiden listausten pohjalta tiedot syötettiin generointia varten Valmetin Excel-määrittelytyökaluun. Määrittelytyökaluun listattiin kaikki käytöt tarvittavine tietoi-neen, määriteltiin käyttökohtaiset operointimahdollisuudet ja konfiguroitiin ryhmä-näytöt. Excel-työkaluun listattiin myös projektin yleiset tiedot, kuten konetunnus, prosessi- ja operointiasemat sekä rajat paperikoneen toimintanopeudelle.

Kun kaikki tarvittavat tiedot oli työkaluun syötetty, generoitiin väliaikaistiedostot, joiden perusteella varsinainen sovellus generoitiin. Valmis sovellus ladattiin näyt-toineen virtuaalikoneen testiprosessiasemalle. Sovellus ja operointinäytöt vii-meisteltiin käsin ja testattiin testausta varten ohjelmoitujen simulointipiirien avulla, ja havaitut virheet korjattiin joko manuaalisesti sovellusta muuttamalla, tai päivit-tämällä generoinnin konfigurointia ja uudelleen generoimalla.

Huomion arvoista on kuitenkin se, että testattavien signaalien lukumäärän ollessa näin suuri, olisi perusteellinen testaaminen jokaisen signaalin osalta hyvin työ-lästä, ilman että vastapuoli eli OPC-palvelin on käytettävissä. Tämä siksi, että jos sovellusta ja operointinäyttöjä testataan ilman, että signaalit ovat kytkettynä mi-hinkään, näkyy operointinäytöillä kaikki luettavat arvot viallisena, mikä tekee to-dellisten vikojen havaitsemisen hankalaksi.

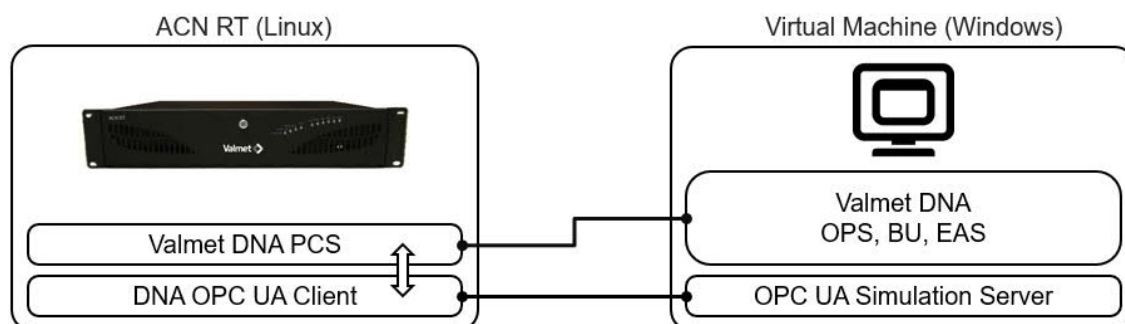
Tämän esitestaamisen tarkoituksena oli siis testata sovelluksen ja operointinäyt-töjen toiminnot siltä osin kuin kohtuullisella ajankäytöllä mahdollista. Testauksen yhteydessä kartoitettiin vikoja, jotka olivat korjattavissa generointipohjia paranta-malla, millä pyrittiin vähentämään projektikohtaista työtä vastaavanlaisten tule-vien projektien osalta.

7 TESTAUS JA TULOKSET

Testauksen tavoitteena oli varmistua OPC UA -rajapinnan toimivuudesta generoidussa Single Window -sovelluksessa ja testata sekä sovelluksen että ope-
rointinäyttöjen toiminta muilta osin. Testauksen yhteydessä havaitut virheet kirjattiin ylös niiden myöhempiä korjauksia varten. Suuri osa virheistä, joilla oli olennainen merkitys testauksen kulun kannalta, korjattiin jo testauksen ohessa. Testaus tapahtui testialueella ja se jakautui sisäiseen testaukseen, joka tapahtui pelkästään Valmet Automation Oy:n henkilöstöllä sekä varsinaiseen FAT-testiin, johon osallistuivat myös asiakas sekä OPC UA -palvelimen ja käyttöjenohjausjärjestelmän toimittaja.

7.1 Sisäinen testaus

Sisäisessä testauksessa haluttiin varmistua OPC UA -rajapinnan toiminnasta ennen varsinaista FAT-testiä. Testausjärjestelmän rakenne on esitetty kuvassa 31. Testauksessa käytetty ACN RT oli projektin toimitukseen kuuluva prosessiohjain, johon oli valmiiksi alustettu DNA-prosessinohjauspalvelin. Tälle koneelle asennettiin uusin versio DNA OPC UA Client -sovelluksesta (DNA Client). Prosessiohjain kytkettiin testausta varten luotuun virtuaalikoneeseen, jossa oli valmiina ope-
rintiasema, varmennuspalvelin sekä suunnitteluasema. Tällainen järjestely siksi, koska projektin kohdetehtaalla nämä ovat jo valmiina. OPC UA -rajapinnan testauksessa käytettiin Prosys-simulointipalvelinta ja UaExpert-ohjelmistoa, jonka avulla voitiin monitoroida simulointipalvelimen dataa. Nämä asennettiin virtuaalikoneelle, joka oli kytkettynä samaan verkkoon ACN-koneen kanssa.



KUVA 31. Testijärjestelmä 1

Ensimmäisessä testitapauksessa haluttiin testata saako DNA Client yhteyden simulointipalvelimelle yksinkertaisimmilla asetuksilla. DNA Client konfiguroitiin DNA System Configurator -työkalulla (SC) mm. seuraavin määrittelysin: palvelimen osoitteeksi asetettiin simulointipalvelimen osoite, kommunikointitapa määritettiin sykliseksi ja tietoturva-asetukset otettiin pois käytöstä. Tietoturva-asetukset määritettiin samaan tilaan myös simulointipalvelimella. Varmennuspalvelimelle lisättiin tekstitiedosto muuttujien määrittelyä varten. Siihen lisättiin muutamia simulointipalvelimen muuttujia, jotka kytkettiin prosessiasemalla suoritettavan DNA-sovelluksen OPC UA -portteihin. Tekstitiedoston nimi syötettiin SC-työkaluun sille tarkoitettuun kenttään. Konfiguraatio tallennettiin, ladattiin ja aktivoitiin, minkä jälkeen ACN-kone käynnistyy automaattisesti uudelleen ja lukee varmennuspalvelimelta konfigurointityökalun generoimat konfiguraatiotiedostot.

Yhteyttä ei heti saatu muodostettua, mutta ongelma ratkesi, kun DNA Client -sovelluksen OPC UA -sertifikaatti hyväksyttiin luotetuksi simulointipalvelimen asetuksista. Tämän jälkeen arvot saatiin päivittymään linkin ylitse molempiin suuntiin. Simulointipalvelimelta luetut muuttujat (kokonaisluku ja liukuluku) vaihtoivat arvoaan jatkuvasti, ja samat arvot saatiin näkyviin DNA:ssa lukemalla vastaavaan OPC UA -nimeen kytketyn rajapintaportin arvoa Debugger-työkalulla. Kirjoitus DNA:sta simulointiserverille testattiin kirjoittamalla Debugger-työkalulla arvoja sovelluksen OPC UA -lähtöihin. Arvojen päivittymistä palvelimelle seurattiin UaExpert-selaimella.

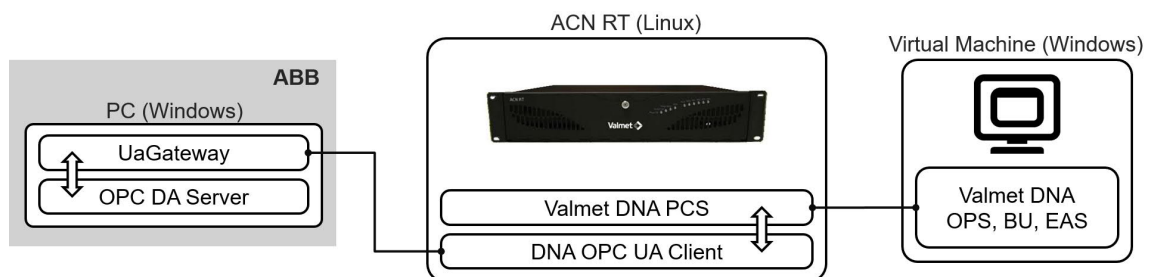
Seuraavaksi testattiin yhteyden muodostamista tiukemmilla tietoturva-asetuksilla. Tietoturvasävy asetettiin sekä simulointipalvelimelta että DNA Client -sovellukselta tilaan "Sign and Encrypt", mikä vaati SC-työkalulla määritettyjen käyttäjätunnusten hyväksyntää simulointipalvelimella. Salausalgoritmiksi valittiin Basic256Sha256. Yhteyden muodostaminen onnistui myös korkeammalla tietoturvan vaatimustasolla.

Tulokset sisäisessä OPC UA -linkkitestissä olivat lupaavia. Yhteyden toimimattomuus johtui lähes aina siitä, että joko DNA Client -sovelluksen sertifikaatti tai käyttäjätunnukset oli unohdettu hyväksyä palvelimelta, tai siitä, että tietoturva-asetukset eivät täsmänneet palvelimella ja DNA Client -sovelluksessa. Sisäisen

testauksen ansiosta varmistuttiin Single Window -sovelluksen OPC UA -rajapinnan toiminnasta ja saatiin tärkeää tietoa DNA Client -sovelluksen konfiguroinnista. Testeissä esiin nousseet huomiot kirjattiin muistiin varsinaista FAT-testiä varten.

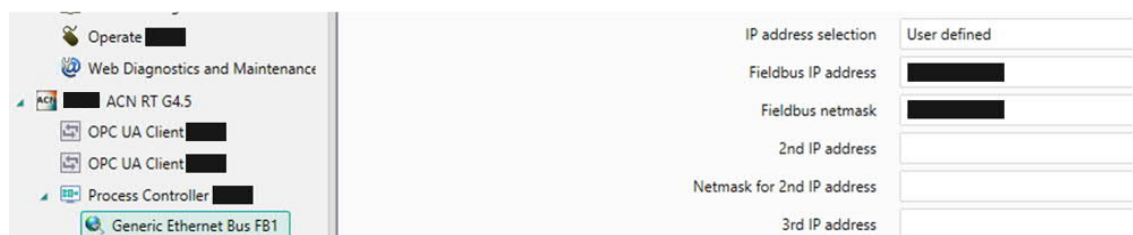
7.2 Varsinainen FAT-testaus

Varsinaisessa FAT-testissä tarkoituksena oli testata Single Window -sovelluksen ja OPC UA -linkin kokonaisvaltainen toimivuus projektin toimitukseen kuuluvalla laitteistolla. Sisäisen testauksen testijärjestelmään lisänä tuli ABB:n Windows-tietokone, joka toimi OPC DA -palvelimena (OPC Classic). Tietokoneelle oli asennettu lisäksi UaGateway, joka mahdollistaa yhteyden OPC UA -asiakasohjelman ja OPC DA -palvelimen välillä. ABB:n OPC-palvelin oli kytketty PLC-laitteistoon, joka piti sisällään sähkökäyttöjen ohjaussovelluksen. Varsinaisen FAT-testin testijärjestelmä on esitetty kuvassa 32.



KUVA 32. Testijärjestelmä 2

Fyysinen yhteys ACN-koneen ja OPC-palvelimen välille muodostettiin Ethernet-linkillä. ACN-koneen portti määriteltiin SC-työkalun laitteistoasetuksista ja IP-osoite valittiin OPC-palvelimen kanssa samasta osoiteavaruudesta (kuva 33).



KUVA 33. Fyysisen Ethernet-yhteyden konfigurointi

Kun fyysinen tiedonsiirtoyhteys oli muodostettu, DNA Client konfiguroitiin ABB:n OPC-palvelimen määrittelyjen mukaisesti. SC-työkaluun asetettiin palvelimen IP-osoite ja TCP-portin numero. Yhteys haluttiin jälleen testata ensin ilman tietoturva vaatimuksia, joten tietoturva-asetukset jätettiin pois päältä kummastakin päästä. Ensimmäisiä testejä varten luotiin uusi tekstitiedosto muuttujien konfiguroinnille. Namespace-indeksinumero OPC-palvelimen muuttujille määriteltiin UaGateway-ohjelmassa, koska Namespace-rakenne on vain OPC UA -määrittelyn ominaisuus. OPC-palvelinkoneelle oli asennettu UaExpert-ohjelma, jolla liitettiin UaGateway-yhdyskäytäväsovellukseen. UaExpert-ohjelman avulla voitiin varmistua siitä, että palvelimelta löytyvien muuttujien nimet täsmäävät OPC-signaaliluetteloon. Tekstitiedostoon lisättiin aluksi vain muutama luku-kirjoitus-pari yhteyden testausta varten. Ennen yhteyden muodostamista DNA Client -sertifikaatti tuli taas hyväksyä UaGateway-ohjelmassa (sertifikaatteja ei tunneta OPC Classic -määrittelyssä).

Edellä mainittujen toimenpiteiden ja määrittelyiden jälkeen yhteys saatiin toimimaan, ja sekä muuttujien lukeminen palvelimelta että kirjoittaminen palvelimelle onnistuivat DNA Client -sovelluksella. Samat testit toistettiin jälleen tiukemmilla tietoturva vaatimuksilla, ja tulokset OPC UA -linkkitestissä olivat hyviä, kun linkkiin oli määritelty vain muutama muuttuja. Seuraavaksi testausta laajennettiin siten, että koko Single Window -sovellus ja siihen kuuluvat operointinäytöt otettiin testaukseen mukaan. Kaikki 2300 signaalia määriteltiin OPC-signaaliluettelon mukaisesti tekstitiedostoon ja uusi konfigurointi synkronoitiin DNA Client -sovellukseen.

Tässä kokonaistestissä tarkoituksena oli testata sovelluksen ja operointinäyttöjen toiminta sekä varmistua siitä, että kaikki tekstitiedostoon määritellyt luvut ja kirjoitukset toimivat OPC UA -linkissä. Testauksessa käytiin läpi operoinnit, ohjearvojen asettelut, mittaukset ja käyttöjen tilatietojen indikoinnit DNA-operointinäytöillä. Vian ilmetessä selvitettiin, onko vika DNA-operointinäytössä, DNA-toimilohkosovelluksessa, OPC UA -linkissä vai ABB:n OPC-palvelimella. Vianetsinnässä hyödynnettiin Debugger-työkalua ja muita DNA:n testiominaisuuksia. Testauksessa käytettiin myös tekstitiedoston pohjalta generoitavia testinäyttöjä, joissa esitettiin kaikki linkissä kulkevat signaalit. Tällä helpotettiin viallisten signaalien havaitsemista.

Kokonaistestissä Single Window -sovelluksen OPC UA -rajapinnasta löytyi joitakin vikoja liittyen datatyypin ristiriitoihin ja porttien virheellisiin nimiin. Ristiriitoja oli esimerkiksi erilaisia tilasanoja lukevissa sovelluspiireissä. Tilasana on kokonaislukusignaali, josta sovelluksessa erotellaan yksittäiset bitit. Jokaisella bitillä on oma merkityksensä, esimerkiksi kommunikointivian indikointi. Tilasanasignaali, joka on datatyypiltään 32-bittinen kokonaisluku (Int32), oli luettu sovellukseen 32 bitin kokonaislukutaulukkona (Bin32). Tätä ratkaisutapaa oltiin käytetty, koska sen ajateltiin yksinkertaistavan sovellusta ja sen muokkaamista, kun ylimääräisiä itobcnv-lohkoja (Integer to Binary Conversion) ei tarvitsisi käyttää yksittäisten bittien erotteluun kokonaisluvusta, vaan bittien arvot olisivat heti I/O-lohkosta saatavilla. Tämä ei kuitenkaan toiminut, sillä Bin32-tulolohko ei muunna 32-bittistä kokonaislukua 32 bitin binääritaulukoksi, vaan se osaa tuoda OPC UA -palvelimelta vain saman muotoisen 32 bitin binääritaulukon. Sovellus korjattiin siten, että 32 bitin binääritaulukoita lukevat OPC UA I/O -lohkot vaihdettiin 32 bitin kokonaislukuja lukeviksi, ja bitit eroteltiin itobcnv-lohkoja käyttäen. Kyseisten signaalien osalta muutokset päivitettiin myös tekstitiedostoon, jossa muuttujat määriteltiin uudelleen.

Korjaustoimenpiteet edellä mainitun ongelman osalta saatiin tehtyä FAT-testauksen aikana ja linkin toiminta saatiin myös niiltä osin testattua positiivisin tuloksin. Kokonaisuudessaan OPC UA -linkin osalta Single Window -sovellus voitiin todeta toimivaksi. Linkkiin liittyvät ongelmat saatiin kaikki korjattua testausten aikana. Sovelluksen toiminnallisuuksiin liittyen jäi vielä hieman korjattavaa FAT-testin jälkeen. Virheet liittyivät pääosin operointinäyttöjen symboleihin ja animaatioihin sekä kireyssäätäjien operointiin. Myös toiminnallisuuksiin liittyvät viat saatiin korjattua pian FAT-testien jälkeen.

8 JOHTOPÄÄTÖKSET JA POHDINTA

Tässä opinnäytetyössä kehitettiin Valmet Automation Oy:lle generoitava Single Window -sovellustuote, jossa liityntä DNA-operointinäyttöjen ja kolmannen osapuolen toimittaman linjakäyttöjen ohjausjärjestelmän välillä voidaan toteuttaa OPC UA -kommunikointilinkillä. Tuotteistuksen pohjalta generoitiin käyttöliittymäsovellus asiakasprojektiin. Sovellus ja OPC UA -liityntä testattiin osana projektin FAT-testejä.

Generoitava OPC UA -liityntää käyttävä Single Window -sovellustuote saatiin toteutettua suunnitelmien mukaisesti, ja työn tuloksena syntynyttä generointialustaa voidaan hyödyntää myös jatkossa vastaavien käyttöliittymäsovellusten tuottamisessa. Asiakasprojektiin generoitu Single Window -sovellus saatiin sekä toimintojen että OPC UA -liitynnän osalta toimimaan, kun testauksissa esiin nousseet viat korjattiin.

Testauksista saatiin tärkeää tietoa DNA OPC UA Client -sovelluksen konfiguroinnista ja optimaalisista parametreista Single Window -linkkiä varten. OPC UA -linkin hitaus tuotti ongelmia testausten alussa. Käytettävyystakuuehtona oli, että linkissä siirtyisi tietty määrä signaaleja sekunnissa. Tiedonsiirtonopeuteen vaikuttavia tekijöitä olivat muun muassa sovellusten (DNA, UaGateway ja ABB) suoritusvälit, DNA Client -sovellukselle määritettävä sykleittäin luettavien muuttujien maksimilukumäärä, tekstitiedostoon konfiguroidut ylimääräiset muuttujat ja DNA Client -sovelluksen kommunikointitapa (jatkuva tai tilauspohjainen).

Tilauspohjaista kommunikointia käytettäessä huomattiin, että joidenkin muuttujien arvot lakkasivat päivittymästä toisinaan. Ongelman aiheuttajaksi selvisi OPC UA Watchdog -piiriin määritetty liian lyhyt valvontakatkaisuaika. Ongelma ratkesi, kun aika muutettiin kolmesta sekunnista 15 sekuntiin. Kolmen sekunnin valvontakatkaisuaika oli liian lyhyt, sillä joillain sykleillä tiedon siirtymiseen saattoi kulua enemmän kuin kolme sekuntia, mikä lopetti arvojen päivittymisen. Syklistä kommunikointia käytettäessä ongelmaksi muodostui se, että vain noin puolet muuttujien arvoista päivittyivät ja osa arvoista ei päivittynyt ollenkaan. Ratkaisu tähän

löytyi sykleittäin luettavien muuttujien maksimilukumäärästä, joka muutettiin arvosta 1000 arvoon 2000. Tämä parametri määrittää DNA OPC UA Client -sovelluksen konfigurointitiedostossa.

Lopulta päädyttiin käyttämään syklistä kommunikointitapaa, koska ACN-prosessiohjaimella ei ollut muita käyttötarkoituksia kuin OPC UA -liitäntäsovelluksen suorittaminen, ja suorituskyyä tarkasteltaessa kuormitus pysyi riittävän maltillisena. Linkki saatiin nopeammaksi puolittamalla DNA-sovelluksen suoritusväli 1000 millisekunnista 500 millisekuntiin. Myös vastapuolen, eli ABB:n käyttöjenohjaussovelluksen suoritusväliä pienennettiin, mikä edelleen paransi linkin suorituskyyä. Suorituskyyä ei pystytty luotettavasti mittaamaan johtuen sovellusten eri suoritusväleistä, mutta testausten perusteella voitiin todeta linkin nopeus riittäväksi.

Linkin muuttujien konfigurointi osoittautui työlääksi, kun signaaleita oli paljon. Tekstitiedosto luotiin hyödyntämällä taulukointiohjelmaa ja erilaisia skriptejä, minkä myötä siihen generoitui myös ylimääräisiä muuttujia, joita OPC-palvelimelta ei löytynyt. Ylimääräiset muuttujat pidensivät linkin ylösnousuun kuluva aikaa. Testauksissa nousi esiin lukuisia muitakin huomioita, jotka kaikki kirjattiin ylös. Opinnäytetyön tuloksena syntyneitä generointialustaa voidaan kehittää korjaamalla testauksissa esiin nousseet viat ja puutteet sovellus- ja operointinäyttöpohjiin siltä osin kuin mahdollista, ja kehittää muuttujien konfigurointia varten työkalu, jolla tekstiedosto voidaan generoida. Joitakin korjauksia generointipohjiin tehtiin jo työn aikana sellaisissa tapauksissa, joissa sama virhe toistui useissa sovelluspiireissä. Generointipohjien päivittämisen jälkeen vialliset sovelluspiirit generoitiin uudelleen korjatuilla pohjilla. Edellä mainittuja kehitystoimenpiteitä edistämällä tulevia sovellustoimituksia voidaan edelleen nopeuttaa.

Johtopäätöksenä voidaan todeta, että Valmet Automation Oy:n toimitusvalmius OPC UA Single Window -sovelluksen osalta parantui kehitystyön ansiosta merkittävästi. Tarjonnan lisääminen OPC UA -tiedonsiirtoprotokollaa hyödyntävien tuotteiden osalta parantaa yrityksen kilpailukykyä nykyaikaisen teknologian toimittajana. Työ kartutti myös tietoa ja käytännön kokemusta jatkuvasti yleistyvän liittymän käyttämisestä teollisuuden järjestelmien välisessä kommunikoinnissa.

LÄHTEET

ACN RT G4 Technical Manual. 2019. Valmet Automation Oy. Luettu 18.7.2019.

Center Driven Reeling. 2014. Valmet. Luettu 2.7.2019. <https://www.valmet.com/board-and-paper/board-and-paper-machines/reeling/center-driven-reeling/>

Damm, M., Leitner, S. & Mahnke, W. 2009. OPC Unified Architecture. Berliini: Springer GmbH.

DNA OPC UA User's Guide. 2018. Valmet Automation Oy. Luettu 29.5.2019.

Drives Operator Training. 2008. Valmet Automation Oy. Luettu 2.7.2019.

Duty, K. 2010. 3 Reasons Linux Is Preferred for Control Systems. Inductive Automation. Luettu 5.7.2019. <https://www.automation.com/library/articles-white-papers/opc-articles-and-white-papers/3-reasons-linux-is-preferred-for-control-systems>

Hunkar, P. 2014. OPC UA vs OPC Classic. DS Interoperability. Luettu 8.7.2019. <http://www.dsinteroperability.com/OPCClassicVSUA.pdf>

Kaijankoski, J. 2012. PPK6:n sähkökäyttöjen elinkaaren hallinta ja kehittäminen. Sähkötekniikan koulutusohjelma. Kemi: Kemi Tornion ammattikorkeakoulu. Opinnäytetyö.

Karaila, M. 2018. Valmet DNA OPC UA Server & Client. Valmet Automation Oy. Luettu 26.6.2019. https://www.automaatioseura.fi/site/assets/files/1824/13_dna_opc_ua_server_ua_client_for_customers_-_mika_karaila.pdf

KnowPap 20.0. 2017. AEL/Proledge Oy. Luettu 11.6.2019. Saatavilla rajoitustusti. http://www.knowpap.com/extranet/suomi/knowpap_system/user_interfaces/knowpap.htm

Kominek, D. 2009. OPC: The Ins and Outs to What It's About. MatrikonOPC. Luettu 3.7.2019. https://www.automation.com/pdf_articles/Guide_to OPC.pdf

Modbus Protocol Interface Manual. 2019. Valmet Automation Oy. Luettu 23.7.2019.

Nopea tiedonsiirto automaatio- ja kenttälaitteiden välillä: MODBUS. 2018. WAGO. Luettu 24.7.2019. <https://www.wago.com/fi/modbus>

Nopeussäädettyjen käyttöjen opas. 2001. ABB Automation Group Ltd. Luettu 1.7.2019. https://library.e.abb.com/public/32f0404329db7689c1256d2800411f0a/Tekninen_opas_nro4.pdf

OPC and OPC UA explained. 2015. Novotek. Luettu 3.7.2019. <https://www.novotek.com/en/solutions/kepware-communication-platform/opc-and-opc-ua-explained>

OPC Foundation. 2019. OPC Foundation. Luettu 3.7.2019. <https://opcfoundation.org/>

OPC Unified Architecture Specification Part 1: Overview and Concepts, Release 1.04. 2017a. OPC Foundation. Luettu 5.7.2019.

OPC Unified Architecture Specification Part 2: Security Model, Release 1.04. 2018. OPC Foundation. Luettu 8.7.2019.

OPC Unified Architecture Specification Part 3: Address Space Model, Release 1.04. 2017b. OPC Foundation. Luettu 9.7.2019.

OPC Unified Architecture Specification Part 6: Mappings, Release 1.04. 2017c. OPC Foundation. Luettu 9.7.2019.

PN/PN Coupler Getting Started. 2014. Siemens. Luettu 31.7.2019. <https://support.industry.siemens.com/cs/document/88737127/pn-pn-coupler-getting-started?dti=0&lc=en-WW>

Romppainen, J. 2002. Linjakäytön ennakkohuolto. Sähkötekniikan osasto. Oulu: Lappeenrannan teknillinen korkeakoulu. Diplomityö.

Simatic 3964(R) RK512 Protocol Interface Manual. 2019. Valmet Automation Oy. Luettu 23.7.2019.

Single Window Interface for Electrical Drives. 1998. Valmet Automation Oy. Luettu 2.7.2019.

Valmet DNA Drive Controls. 2016. Valmet Automation Oy. Luettu 1.7.2019.

What is OLE (object linking and embedding)?. 2014. Computer-Notes Inc. Luettu 3.7.2019. <http://ecomputernotes.com/visual-basic/basic-of-visual-basic/ole>

Why OPC UA Matters. 2019. National Instruments. Luettu 15.7.2019. <https://www.ni.com/fi-fi/innovations/white-papers/12/why-opc-ua-matters.html>