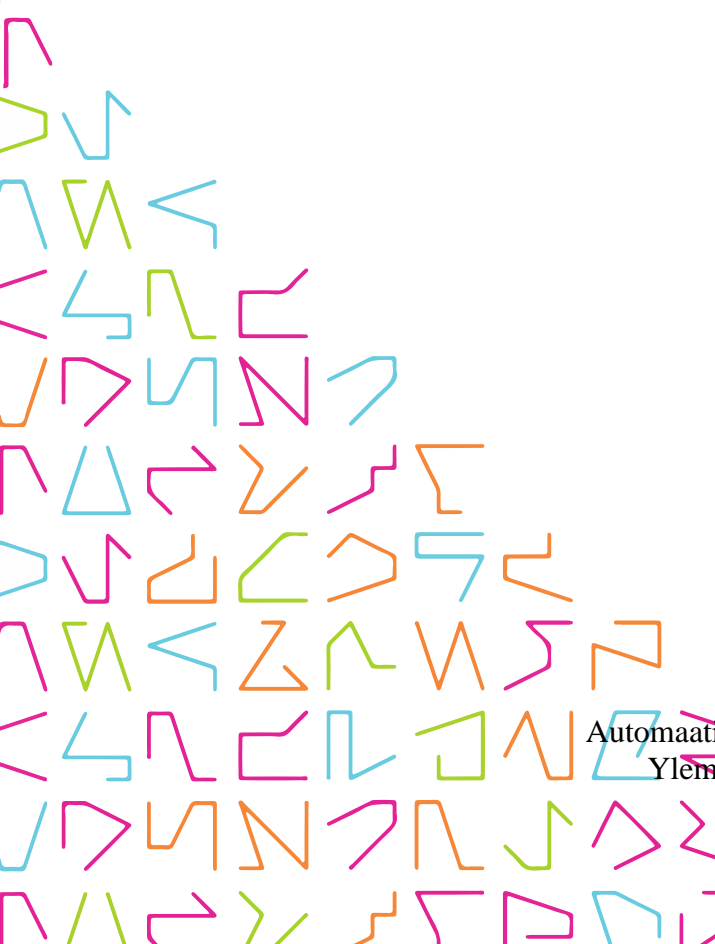




TAMPEREEN
AMMATTIKORKEAKOULU

PNEUMAATTISEN SIIRTOJÄRJESTELMÄN INTEGROINTI TUOTANNONOHJAUSJÄRJESTELMÄÄN

Antti Leppänen



Opinnäytetyö
Marraskuu 2018

Automaatiotekniikan koulutusohjelma
Ylempi ammattikorkeakoulu

TIIVISTELMÄ

Tampereen ammattikorkeakoulu
Automaatiotekniikan koulutusohjelma, ylempi ammattikorkeakoulu

LEPPÄNEN ANTTI:

Pneumaattisen siirtojärjestelmän integrointi tuotannonohjausjärjestelmään

Opinnäytetyö 80 sivua, joista liitteitä 0 sivua
Marraskuu 2018

Nokian Renkaiden sekoitusosastolle toteutettiin vuoden 2018 aikana uusi tuotannonohjausjärjestelmä. Tiettyjä sekoitusosaston automaatiojärjestelmiä oli tarkoitus hallita tuotannonohjausjärjestelmästä. Tämä vaati paljon integraatiotyötä.

Opinnäytetyössä perehdyttiin, kuinka pneumaattinen siirtojärjestelmä integroidaan tehtaaseen uuteen tuotannonohjausjärjestelmään. Pneumaattisen siirtojärjestelmän toiminnallisuksia siirrettiin hallinnoitavaksi tuotannonohjausjärjestelmään. Opinnäytetyössä tutkittiin, kuinka automaatiojärjestelmissä kerätään, siirretään ja varastoidaan tietoa. Lisäksi perehdyttiin tuotannonohjausjärjestelmiin ja ISA-95 -standardiin. Näin kohdeyrityksen järjestelmähierarkia pystyttiin kuvaamaan standardin mukaisesti. Opinnäytetyön tavoitteena oli integroida sekoitusosaston pneumaattinen siirtojärjestelmä uuteen tuotannonohjausjärjestelmään tutkittua tietoa hyödyntäen.

Nokian Renkaiden sekoitusosaston tuotannonohjausjärjestelmän ja ohjelmoitavien logiikoiden välistä tiedonsiirtoa varten asennettiin kaksi OPC-palvelinta. Pneumaattisen siirtojärjestelmän tiedonsiirron lisäksi asennetut OPC-palvelimet siirtäisivät tulevaisuudessa myös muiden integroitavien järjestelmien tietoja tuotannonohjausjärjestelmään. Pneumaattisen siirtojärjestelmän integraatiota varten OPC-palvelimille luotiin tarvittavat OPC-tagit, jotka linkitettiin ohjelmoitaviin logiikoihin. Ohjelmoitaviin logiikoihin toteutettiin tarvittavat PLC-ohjelmat, jotta tiettyjä pneumaattisen siirtojärjestelmän toiminnallisuksia voitaisiin hallita tuotannonohjausjärjestelmästä. PLC-ohjelmia testattiin erillisellä testilogiikalla, jotta sekoitusosaston tuotanto ei häiriintyisi.

Pneumaattinen siirtojärjestelmän integraation toteutus oli tehtyjen testien perusteella onnistunut. Järjestelmän lopullista käyttöönottoa ei aikataulullisista syistä käsitelty opinnäytetyössä. Järjestelmän käyttöönottoa varten toteutettiin riskianalyysi, jota hyödyntäen tehtiin käyttöönottosuunnitelma. Ohjelmoitavan logiikan ja tuotannonohjausjärjestelmän välinen kommunikointi olisi mahdollista toteuttaa useammalla eri tavalla. Kommunikoinnin toteutustapa riippuu siitä, halutaanko tuotannonohjausjärjestelmän ja ohjelmoitavien logiikoiden OPC-tageja hallita samalta palvelimelta vai järjestelmäkohtaisesti.

Asiasanat: ohjelmoitava logiikka, opc, tuotannonohjausjärjestelmä, integraatio

ABSTRACT

Tampereen ammattikorkeakoulu
Tampere University of Applied Sciences
Master's Degree Programme in Automation Technology

LEPPÄNEN ANTTI:

Integration of pneumatic transfer system into manufacturing execution system

Master's thesis 80 pages, appendices 0 pages
November 2018

In 2018, a new manufacturing execution system was implemented for mixing department of Nokian Tyres. Certain mixing department's automation systems were intended to be managed by the manufacturing execution system and this required much integration work.

The thesis focused on how a pneumatic transfer system is integrated into the factory's new manufacturing execution system. The functionalities of the pneumatic transfer system were transferred to the manufacturing execution system. The thesis made research how automation systems collect, transfer and store information. In addition, the manufacturing execution systems and the ISA-95 standard were studied. In this way, the target company's system hierarchy could be described in accordance with the standard. The aim of the thesis was to integrate a pneumatic transfer system of the mixing department into the new manufacturing execution by using the information studied.

Two OPC servers were installed in Nokian Tyres mixing department, for communication between manufacturing execution system and programmable logic controller. In addition of data transfer of the pneumatic transfer system the installed OPC servers would also transfer the information of other integrating systems to the manufacturing execution system in the future. A OPC tags were created to the OPC servers for the integration of pneumatic transfer system. The OPC tags were linked to programmable logic controllers. A PLC programs were implemented to programmable logic controllers for controlling certain functionalities of pneumatic transfer system from manufacturing execution system. The PLC programs were tested with a separate test logic controller that the mixing department production would not be disturbed.

The implementation of the pneumatic transfer system integration was successful based on the tests performed. The final commissioning of the system was not discussed in the thesis because of time schedule reasons. A risk analysis was made for commissioning the system. Commissioning plan was made by utilizing the risk analysis. Communication between the programmable logic controller and the manufacturing execution system could be implemented in several different ways. The way the communication is implemented depends on whether the OPC tags of the manufacturing execution system and the programmable logic controllers are managed from the same server or system-specific.

Key words: plc, opc, mes, integration

SISÄLLYS

1	JOHDANTO.....	8
1.1	Työn tausta.....	8
1.2	Työn tavoite ja tarkoitus	8
1.3	Työn rajaukset.....	8
1.4	Tutkimusmenetelmä.....	9
2	TIEDONKERÄÄMINEN, SIIRTO JA VARASTOINTI AUTOMAATIOJÄRJESTELMISSÄ	10
2.1	Automaatiojärjestelmä	10
2.1.1	Ohjelmoitava logiikka.....	11
2.1.2	Kenttäväylä	15
2.1.3	Valvomo-ohjelmisto.....	17
2.1.4	Automaatiojärjestelmien integrointi	18
2.2	OPC.....	19
2.2.1	OPC:n historia.....	19
2.2.2	OPC Classic	23
2.2.3	OPC Unified Architecture.....	30
2.3	Tiedonhallinta ja varastointi	40
2.3.1	Tietokanta.....	41
2.3.2	Erityyppiset tietokannat	42
3	TUOTANNONOHJAUS.....	44
3.1	Tuotannonohjausjärjestelmä	44
3.2	ISA-95 -standardi.....	45
4	JÄRJESTELMÄHIERARKIA	49
4.1	Järjestelmien välinen kommunikointi.....	51
5	PNEUMAATTISEN SIIRTOJÄRJESTELMÄN INTEGRAATIO.....	52
5.1	Pneumaattisen siirtojärjestelmän yleiskuvaus.....	52
5.2	Vanhan järjestelmän rakenne.....	55
5.3	Uuden järjestelmän rakenne.....	57
6	INTEGRAATION TOTEUTUS	58
6.1	OPC-palvelin ja -tagit	58
6.2	PLC-ohjelma	62
6.2.1	Siilojen lupataulu ja siilokartta	63
6.2.2	Siilojen rajatiedot	66
6.2.3	Päiväsiilojen täyttökielto.....	67
7	TESTAUS JA KÄYTTÖÖNOTTOSUUNNITELMA	70
7.1	Käyttöönoton riskianalyysi	71

7.2 Käyttöönottosuunnitelma.....	72
8 POHDINTA.....	75
8.1 Järjestelmän jatkokehitys	76
LÄHTEET.....	79

LYHENTEEN JA TERMIT

ADI	Analyzer Devices Integration
AGV	Automated/Automatic Guided Vehicle
API	Application Programming Interface
COM	Component Object Model
CPU	Central Processing Unit
DB	Database
DB	Data Block
DCS	Distributed Control System
ERP	Enterprise Resource Planning
FB	Function Block
FBD	Function Block Diagram
FC	Function
FDI	Field Device Integration
GIS	Geographical Information System
HMI	Human Machine Interface
HTTP	Hypertext Transfer Protocol
HW	Hardware
IL	Instruction List
IP	Internet Protocol
LD	Ladder Diagram
MES	Manufacturing Execution System
MIS	Management Information System
MOM	Manufacturing Operations Management
OB	Organization Block
ODBC	Open Database Connectivity
OLAP	Online Analytical Processing
OLE	Object Linking and Embedding
OPC	Open Platform Communications (aiemmin OLE for Process Control)
OPC UA	OPC Unified Architecture
PC	Personal Computer
PLC	Programmable Logic Controller

POU	Program Organization Unit
RDB	Relational Database
RTU	Remote Terminal Unit
SCADA	Supervisory Control and Data Acquisition
SCM	Supply Chain Management
SDK	Software Development Kit
SFC	Sequential Function Chart
SOAP	Simple Object Access Protocol
SQL	Structured Query Language
ST	Structured Text
TCP	Transmission Control Protocol
WMS	Warehouse Management System
XML	Extensible Markup Language

1 JOHDANTO

Työn ensimmäisessä luvussa käsitellään taustaa, tavoitteita, tarkoitusta, rajouksia ja työssä käytettäviä tutkimusmenetelmiä.

1.1 Työn tausta

Opinnäytetyön toimeksiantajana toimii Nokian Renkaat Oyj. Työ toteutetaan yrityksen Nokialla sijaitsevan tehtaan sekoitusosastolle vuoden 2018 aikana. Sekoitusosasto palvelee tehtaan muita osastoja tarjoten erilaisia kumisekoituksia kunkin osaston tarpeisiin. Yrityksen sekoitusosastolle toteutetaan uusi tuotannonohjausjärjestelmä, johon osaston automaatiojärjestelmät tulisi integroida.

1.2 Työn tavoite ja tarkoitus

Opinnäytetyön tavoitteena on perehtyä, kuinka pneumaattinen siirtojärjestelmä integroida tehtaan uuteen tuotannonohjausjärjestelmään. Lisäksi tavoitteena on perehtyä tuotannonohjausjärjestelmiin ja ISA-95 -standardiin, jotta kohdeyrityksen järjestelmähierarkia pystyttäisiin kuvaamaan standardin mukaisesti.

Pneumaattisen siirtojärjestelmän toiminnallisuuksia on tarkoitus siirtää hallinnoitavaksi tuotannonohjausjärjestelmään. Ennen integroidun järjestelmän käyttöönottoa tarkoituksena on tehdä riskianalyysi ja käyttöönottosuunnitelma. Työn tarkoituksena on järjestelmien onnistunut toteutus, testaus ja lopullista käyttöönottoa varten laadittu käyttöönottosuunnitelma. Työn lopuksi pohditaan vaihtoehtoisia toteutustapoja ohjelmoitavan logiikan ja tuotannonohjausjärjestelmän väliseen tiedonsiirtoon.

1.3 Työn rajaukset

Opinnäytetyön aihe on päätetty rajata käsittelemään pneumaattisen siirtojärjestelmän integrointia tuotannonohjausjärjestelmään. Integrointityötä tullaan tekemään muihinkin tehtaan järjestelmiin, mutta resursseista ja aikataulusta johtuen, opinnäytetyö on rajattu koskemaan vain pneumaattisen siirtojärjestelmän integraatiota.

1.4 Tutkimusmenetelmä

Opinnäytetyössä tutkimusmenetelmänä käytetään kirjallisuustutkimusta. Kirjallisuustutkimuksen tarkoituksena on perehtyä, kuinka tiedonsiirtoa automaatiojärjestelmissä toteutetaan. Aluksi kuvataan automaatiojärjestelmä yleisellä tasolla. Tämän jälkeen automaatiojärjestelmien eri osia tutkitaan alhaalta ylöspäin. Kirjallisuustutkimusta varten tietoa on kerätty Internet-lähteistä, kirjallisuudesta ja standardeista. Kirjallisuustutkimuksen lähteinä käytetään suomenkielistä ja englanninkielistä lähdekirjallisuutta.

Työssä perehdytään automaation standardeihin. Standardeja tutkitaan, jotta voitaisiin hahmottaa yrityksen koko järjestelmähierarkia. Standardeja tutkitaan myös, jotta tiedettäisiin, mille tasolle automaatiojärjestelmät hierarkiassa kuuluvat.

Opinnäytetyön toisessa ja kolmannessa kappaleessa kirjallisuustutkimus käsittelee järjestelmien osia yleisellä tasolla. Opinnäytetyön loput kappaleet käsittelevät pneumaattisen siirtojärjestelmän integraatiota tuotannonohjausjärjestelmään. Kirjallisuustutkimuksen tietoa sovelletaan käytäntöön, kun automaatiojärjestelmiä integroidaan yrityksen uuteen tuotannonohjausjärjestelmään.

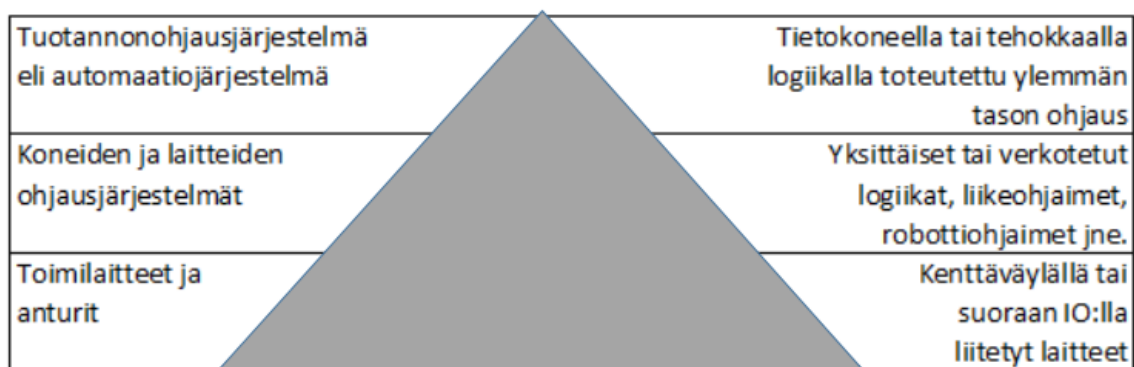
2 TIEDONKERÄÄMINEN, SIIRTO JA VARASTOINTI AUTOMAATIOJÄRJESTELMISSÄ

Tässä kappaleessa käsitellään, kuinka tietoa kerätään, siirretään ja tallennetaan automaatiojärjestelmissä. Aluksi käsitellään automaatiojärjestelmiä yleisellä tasolla ja tämän jälkeen tiedonsiirtämistä ohjelmoitavista logiikoista kohti ylätasojärjestelmiä.

2.1 Automaatiojärjestelmä

Ohjausjärjestelmä on automaattisesti toimivien koneiden ja laitteiden ohjausjärjestelmien yleisnimitys. Ohjausjärjestelmän tehtävä on ohjata koneiden ja tuotantolinjojen toimintoja laitteiden tilatietojen ja annettujen komentojen mukaisesti. Usein käytetään termejä automaatiojärjestelmä, ohjausjärjestelmä ja säätöjärjestelmä. Selkeitä rajoja tai eroja termien välillä on vaikeaa määrittää ja niiden käyttö vaihtelee esittäjän mukaan. (Keinänen, Kärkkäinen, Lähetkangas & Sumujärvi 2007, 209.)

Automaatiojärjestelmä termiä käytetään yleisesti ylemmän tason ohjausjärjestelmästä. Automaatiojärjestelmä on joko yksittäinen tietokone tai useamman tietokoneen muodostama verkko, joka valvoo ja ohjaa tuotantolinjan tai koko tehtaan toimintaa. Kuvassa 1 on esitetty automaation eri tasot tuotannonohjausjärjestelmästä alaspäin. (Keinänen ym. 2007, 209.)



KUVA 1. Automaation eri tasot (Yli-Knuutila 2007, 11)

Tarvittaessa automaatiojärjestelmällä on mahdollista kerätä ja tallentaa erilaista tuotantoon ja laitteisiin liittyvää tietoa, jota voisi olla esimerkiksi koneiden käyntiaika, valmistusmäärät ja tuotannonhäiriöt. Tietoa voidaan hyödyntää muun muassa ennakkohuollon

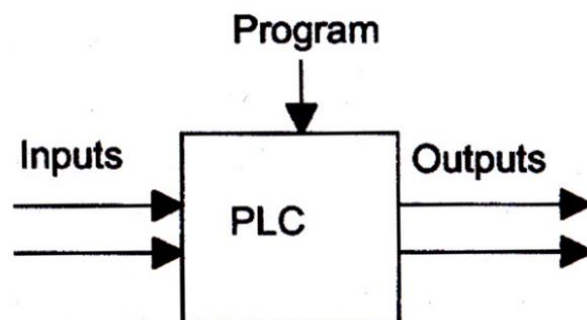
ja tuotannon suunnitteluun, varaston tietojen päivitykseen sekä raaka-aineiden tilaamiseen. (Keinänen ym. 2007, 210.)

Yleensä automaatiojärjestelmä ei suoraan ohjaa yksittäisen koneen toimintoja, vaan se antaa käskyjä alemman tason ohjausjärjestelmille, jotka toteuttavat koneiden suoran ohjauksen. (Keinänen ym. 2007, 210).

2.1.1 Ohjelmoitava logiikka

Ohjelmoitava logiikka (*engl. Programmable Logic Controller*) on erityinen mikroprosessoripohjainen ohjain, joka käyttää ohjelmoitavaa muistia tallentaakseen ohjeita ja suorittaa toiminnallisuuksia. Toiminnallisuuksia ovat esimerkiksi logiikka, sekvenssi, ajastin, laskuri ja aritmeettiset funktiot, joilla voidaan ohjata koneita ja prosesseja. Ohjelmoitavan logiikan ohjelma on suunniteltu sellaiseksi, että ohjelmoijan ei tarvitse olla erityisen kokenut perinteisissä ohjelmointikielissä. (Bolton 2006, 3.)

Logiikka -termiä käytetään, koska ohjelmointi on pääasiassa logiikka- ja kytkentäoperaatioiden toteuttamista. Tuloyksiköihin kytketyt anturoinnit välittävät tietoa ohjelmoitavaan logiikkaan. Ohjelmoitavan logiikan lähtöyksiköillä voidaan esimerkiksi ohjata moottoreita tai venttiileitä. Ohjelmoitavan logiikan muistiin ladattu ohjelma suorittaa ohjauksia asetettujen sääntöjen mukaan. Kuvassa 2 on esitetty ohjelmoitavan logiikan rakenne. (Bolton 2006, 3.)



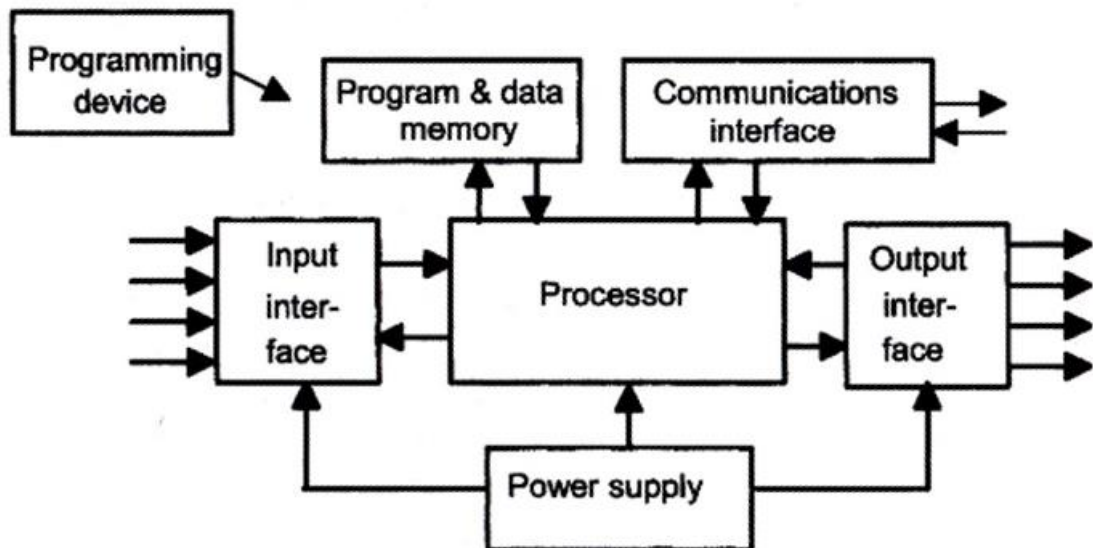
KUVA 2. Ohjelmoitavan logiikan pelkistetty toiminta (Bolton 2006, 3)

Ohjelmoitavan logiikan etu on se, että samaa perustyyppin ohjainta voidaan käyttää laajasti erilaisissa järjestelmissä. Ohjainjärjestelmän muutoksia on mahdollista tehdä joustavasti ja tehokkaasti vain muuttamalla ohjelmaa, jolloin kytkentöihin ei välttämättä tarvitse tehdä muutoksia. Ohjelmoitavat logiikat ja tietokoneet (*engl. Personal Computer*) ovat

samankaltaisia, mutta tietokoneet on optimoitu pääasiassa laskenta- ja visualisointitehtäviin. Ohjelmoitavat logiikat on optimoitu ohjaustehtäviin ja teolliseen ympäristöön. (Bolton 2006, 3.)

Laitteisto

Tyypillisen PLC-systeemin toiminnalliset komponentit ovat suoritin, muisti, virtalähde, tulo- ja lähtöyksikkö, viestintä rajapinta ja ohjelmointilaite. Kuvassa 3 on esitetty ohjelmoitavan logiikan perusjärjestely. (Bolton 2006, 4; Hackworth & Hackworth 2004, 39-40.)



KUVA 3. Ohjelmoitava logiikka (Bolton 2006, 4)

- Suoritin (*engl. Central Processing Unit / Processor*) on yksikkö, joka sisältää mikroprosessorit. Suoritin tulkitsee tulosignaaleja ja tekee ohjauksia lähtöyksiköihin muistissa olevan ohjelman mukaisesti.
- Virtalähde (*engl. Power Supply*) muuntaa vaihtojännitteen matalaksi tasajännitteeksi, jota suoritin, viestintäyksikkö ja tulo- ja lähtöyksiköt käyttävät.
- Ohjelma kehitetään ohjelmointilaitteella (*engl. Programming Device*), jolla tarvittava ohjelma myös ladataan suorittimen muistiin.
- Muistiyksikköön (*engl. Program & Data Memory*) on tallennettu ohjelma, jota käytetään ohjaustoimintojen suorittamiseen mikroprosessorilla. Muistiin on myös

tallennettu tuloyksikköjen tiedot prosessointia ja lähtöyksikköjen ohjaamista varten.

- Tulo- ja lähtöyksiköillä (*engl. Input & Output Interface*) suoritin saa tietoa ulkoisilta laitteilta ja lähettää tietoa ulkoisille laitteille. Tulo- ja lähtölaitteet voidaan luokitella signaalin mukaan diskreettiin, digitaaliseen tai analogiseen. Tulosignaaleja voidaan saada esimerkiksi kytkimistä, valokennoantureista, lämpötila-antureista tai virtaussensoreista. Lähtösignaaleilla voidaan ohjata esimerkiksi moottorin kontaktorin kela tai venttiilin solenoidia.
- Viestintärajapintaa (*engl. Communication Interface*) käytetään tiedon lähettämiseen ja vastaanottamiseen viestintäverkossa. Ohjelmoitava logiikka voi vastaanottaa ja lähettää tietoa esimerkiksi toisella ohjelmoitavalle logiikalle viestintäverkon välityksellä. (Bolton 2006, 4.)

IEC 61131-3 standardi

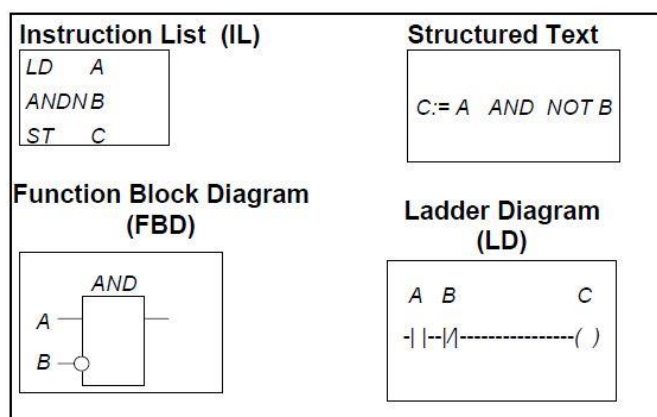
IEC 61131-3 standardin ensimmäinen versio julkaistiin vuonna 1993. Standardia on päivitetty vuosina 2003 ja 2013. IEC 61131-3 standardi määrittää ohjelmoitavien logiikkojen käyttämiä ohjelmointikieliä sekä niiden syntaksia ja toimintaa. Standardi on yleisesti hyväksytty laitevalmistajien keskuudessa. Standardi ei velvoita laitevalmistajia noudattamaan standardin antamia määräytyksiä, vaan niitä pidetään suositeltavana tapana toimia. (Asmala 2005, 10; Hansen 2015, 137.)

IEC 61131-3 standardi on jaettu kahteen osaan: yleisiin elementteihin (*engl. Common Elements*) ja ohjelmointikieliin (*engl. Programming Languages*). Ensimmäisessä osassa käsitellään logiikkaohjelmien yleisesti käyttämiä elementtejä, joista ohjelma muodostuu (taulukko 1). (PLCOpen 2017, 1-3.)

TAULUKKO 1. IEC 61131-3 standardin ensimmäisen osan yleiset elementit (PLCOpen 2017, 1.)

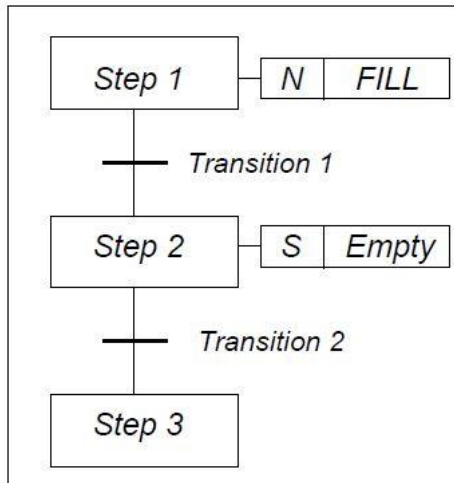
Yleiset elementit
Datatyypit <ul style="list-style-type: none"> • Bool • Integer • Real • Byte • Word • Date • Time_Of_Day • String
Muuttujat
Kokoonpanot, resurssit & tehtävät
Ohjelman eri rakenneyksiköt POU (Program Organization Units) <ul style="list-style-type: none"> • Function • Function Block (FB)
Vuokaavio-ohjelmointi (Sequential Function Chart)

Standardin toisessa osassa käsitellään standardin mukaisia ohjelmointikieliä. Niitä ovat käskylista (*engl. Instruction List*), rakenteinenteksti (*engl. Structured Text*), tikapuu- tai relekaavio (*eng. Ladder Diagram*) ja toimilohkokaavio (*engl. Function Block Diagram*). Kuvassa 4 on esitetty sama ohjelma kaikilla IEC 61131-3 mukaisilla kielillä. (PLCOpen 2017, 1.)



KUVA 4. IEC 61131-3 standardin määrittämät ohjelmointikieliset (PLCOpen 2017, 1)

Standardin osassa yksi mainittu vuokaavio-ohjelmointi (*engl. Sequential Function Chart*) esittää graafisesti sekvenssimuodossa suoritettavan ohjelman käyttäytymisen. SFC järjestää ohjelman sisäisen rakenteen hallittaviin osiin, mikä helpottaa ohjelman hahmottamista. Kuvassa 5 on esitetty vuokaavio-ohjelman rakenne. (PLCOpen 2017, 1.)

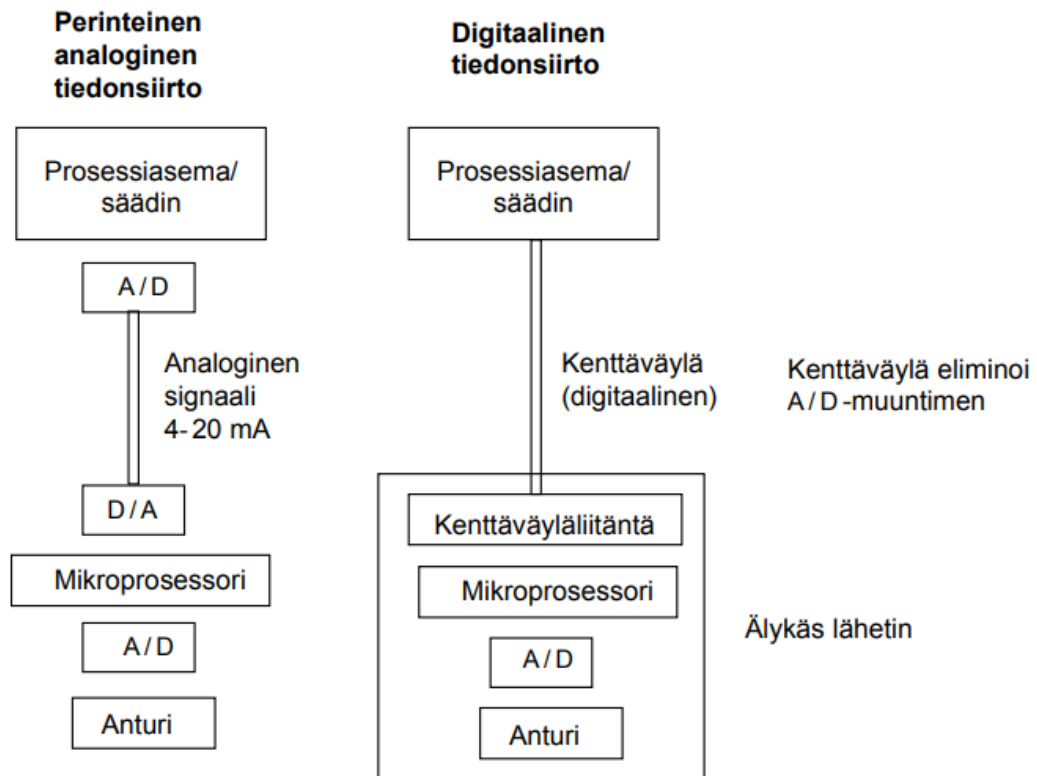


KUVA 5. Vuokaavio-ohjelman rakenne (PLCOpen 2017, 2)

Vuokaavio-ohjelma sisältää askelia, jotka ovat linkitettyinä toimintalohkoihin ja siirtoehtoihin. Jokainen askel esittää ohjattavan systeemin tiettyä tilaa. Siirtoehdon täytyessä aiempi askel deaktivoituu ja seuraava askel aktivoituu. Aktivoituun askeleeseen linkitetty toimintalohko suorittaa sille määritellyn ohjaustoiminnon. (PLCOpen 2017, 2.)

2.1.2 Kenttäväylä

Kenttäväylä on laiteverkko, jossa verkon kenttälaitteet ovat yhdistetty toisiinsa tiedonsiirtokaapelilla. Kenttäväylä käsitettä kuvaa parhaiten digitaalisuus, sarjamuotoisuus ja kaksisuuntainen kommunikointi. Ohjelmoitavan logiikan tulo- ja lähtöportit ovat sijoitettuina erillisiin kenttäväylämoduuleihin. Ohjelmoitava logiikka ja kenttäväylämoduulit kommunikoivat keskenään kenttäväyläkaapeloinnin välityksellä. Kenttäväylä mahdollistaa tiedonsiirron kenttälaitteiden ja automaatiojärjestelmän välillä. Kuvassa 6 on esitetty analogisen ja digitaalisen tiedonsiirron erot lähettimestä säätötasolle. (ABB 2000-07, 3; Keinänen ym. 2007, 214)



KUVA 6. Analogisen ja digitaalisen tiedonsiirron erot lähettimestä säätötasolle (ABB 2000-07, 2)

Kenttäväylän etuja ovat edullisemmat kaapelointi- ja kunnossapitokustannukset sekä asennus- ja muutostöiden nopeutuminen. Toiminnallisia etuja ovat kaksisuuntainen tiedonsiirto ja laitteiden keskinäinen käytettävyys (*engl. Interoperability*). (ABB 2000-07, 2.)

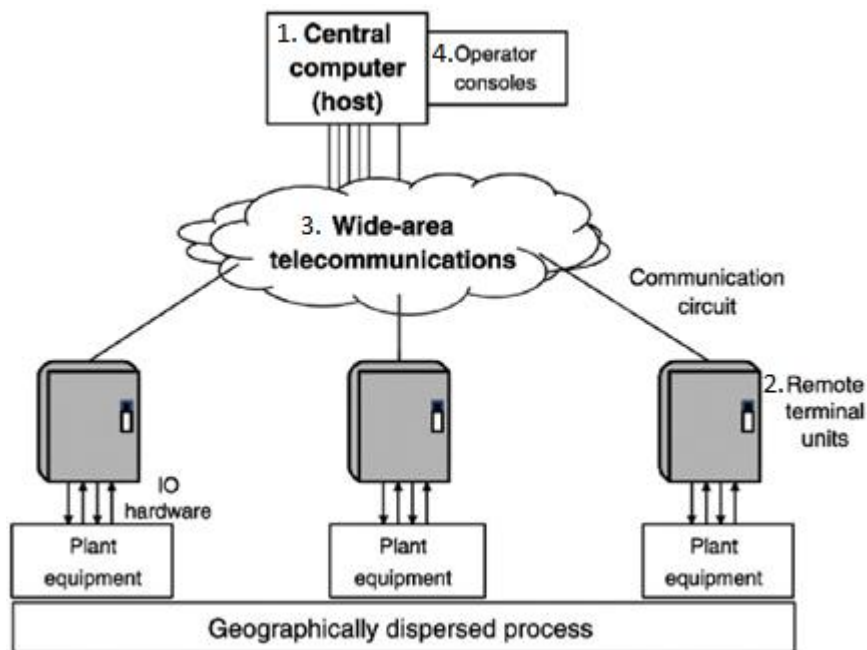
Markkinoilla on useita eri kenttäväyläprotokollia. Protokolla tarkoittaa kieltä tai kielioppia, jota laitteet käyttävät kussakin kenttäväylässä. Yleisimpiä käytettäviä kenttäväyliä ovat avoimen protokollan kenttäväylät Profibus, Profinet, DeviceNet ja Asi. Avoimuudella tarkoitetaan, että kaikki valmistajat voivat valmistaa ja myydä tähän protokollaan yhteensopivia tuotteita. Kenttäväylä luo pohjaa OPC-palvelimen hyödyntämiselle, joka mahdollistaa automaatiojärjestelmätason kytkeytymisen tehdastason hallinnollisiin tietojärjestelmiin. (ABB 2000-07, 2; Keinänen ym. 2007, 214.)

2.1.3 Valvomo-ohjelmisto

Valvomo-ohjelmistoja (*engl. Supervisory Control And Data Acquisition, SCADA*) käytetään yleisesti eri teollisuuden aloilla prosessien valvontaan ja ohjaamiseen. SCADA-järjestelmä mahdollistaa ohjausmuutoksien tekemisen, hälytyksien seurannan ja mittatietojen keräämisen yhdestä paikasta, vaikka prosessi olisi satojen tai tuhansien kilometrien laajuinen. SCADA-järjestelmästä saatava etu kasvaa, mikäli ohjattavat prosessi on laaja tai laitteistolle on hankala päästä. Tyypillinen SCADA-järjestelmä koostuu neljästä alijärjestelmästä (kuva 7).

1. Keskustietokone (*engl. Central Computer*)
2. Hajautetut kenttälaitteet (*engl. Remote Terminal Unit*)
3. Laaja-alainen viestintäjärjestelmä (*engl. Wide-area Telecommunications*)
4. Käyttöliittymä, joka tarjoaa käyttäjälle hallinnan järjestelmään (*engl. Operator Consoles*)

(Mehta & Reddy 2015, 237.)



KUVA 7. SCADA-järjestelmän rakenne (Mehta & Reddy 2015, 238)

Vuosikymmenien aikana SCADA-järjestelmän käyttäminen ja taustalla olevat teknologiat ovat kehittyneet, mutta arkkitehtuuri neljän alijärjestelmän suhteen ei ole juurikaan muuttunut. SCADA-järjestelmällä voidaan toteuttaa monia eri toimintoja, joista keskeisiä ovat:

1. Tiedon hankinta hajautetuilta kenttälaitteilta.
 2. Kenttälaitteilta tulevan tiedon prosessointi hälytyksien ja muiden merkittävien muutoksien havaitsemiseksi prosessista.
 3. Johdonmukaisen tietokannan tarjoaminen prosessidatasta.
 4. Tiedon esittäminen helposti ymmärrettävän graafisen käyttöliittymän avulla, josta selviää prosessin hälytykset, trendit ja raportit.
 5. Kenttälaitteiden etäohjaus.
 6. Järjestelmän seuranta, diagnostiikka ja tarvittavien toimintojen suorittaminen.
 7. Historiallisen tiedon arkistointi myöhempää analysointia varten.
 8. Teknisten tietojen siirtäminen reaaliajassa mallinnusjärjestelmään ja sieltä ulos.
 9. Järjestelmän tietojen tarjoaminen johdon tietojärjestelmiin (*engl. Management Information Systems*) ja toimitusketjun hallintaan.
 10. Tarjota integraatio maantieteellisiin tietojärjestelmiin (*engl. Geographical Information System*).
- (Mehta & Reddy 2015, 238.)

2.1.4 Automaatiojärjestelmien integrointi

Järjestelmien välinen tiedonsiirto on keskeinen ongelma avointen toimintaympäristöjen toteuttamiselle. Ongelman ratkaisuksi on kehitetty erilaisia rajapintoja, joiden avulla tieto siirtyy eri järjestelmien ja ohjelmien välillä. Rajapinnan toteutuksessa tulee ottaa huomioon tiedon eheys niin, että kolmas osapuoli ei voi muokata välitettäviä viestejä. ”Rajapinta on kahden erillisen järjestelmän tai järjestelmän kahden eri osan välinen yhteyksien kokoelma, jossa siirretään tietoa tai pyydetään palvelua toiselta osapuolelta.” (Härkönen, Mikkola, Piikkilä, Sahala, Sahsten, Sandström, Sirviö, Spangar & Sulku 2012, 148.)

Automaatiojärjestelmien integroituminen perustuu esimerkiksi XML-formaattiin tai standardimuotoiseen OPC-rajapinnan välityksellä toteutettuun tiedonsiirtoon järjestelmien välillä. Integroidulla automaatiojärjestelmällä tarkoitetaan kahden tai useamman ohjauvan tai valvovan järjestelmän yhteen liittämistä. Lopputuloksena yhteen liitetyt järjestelmät muodostavat uuden toiminnallisen kokonaisuuden tai järjestelmän. Avoimet standardirajapinnat mahdollistavat hajautettujen osajärjestelmien liittämisen integroituun järjestelmään sekä rinnakkaisten järjestelmien kommunikoinnin keskenään. (Härkönen ym. 2012, 148-149.)

2.2 OPC

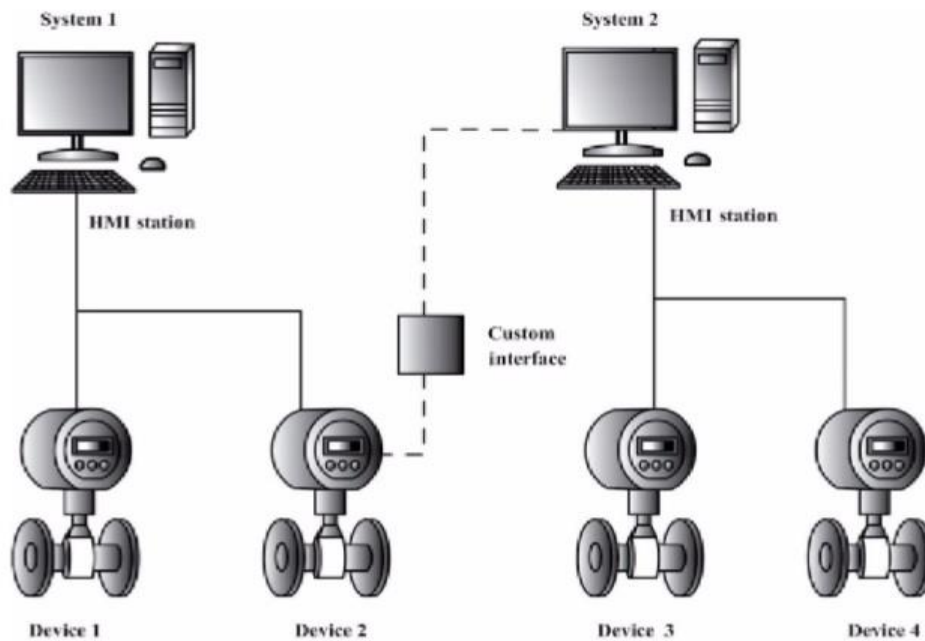
OPC Foundation on viime vuosina määritelty lukuisia ohjelmistorajapintoja standardoidakseen tiedonkulun prosessitasolta hallintotasolle. Pääkäyttötarkoitukset rajapinnoille ovat teollisuusautomaation sovellukset, kuten HMI- ja SCADA-järjestelmät. Niillä voidaan käyttää laitteiden tämän hetkistä dataa ja tarjota hallintoversioiden tietoa tästä hetkestä ja menneistä tapahtumista. (Mahnke, Leitner & Damm 2009, 3.)

OPC on yhteentoimivuusstandardi (*engl. Interoperability Standard*) luotettavan ja turvallisen tiedonvaihdon varmistamiseksi teollisuusautomaatiossa sekä muilla teollisuudenaloilla. OPC on alustasta riippumaton ja se kykenee varmistamaan tiedon saumattoman kulun useiden eri laitevalmistajien laitteiden välille. (OPC Foundation, What is OPC.)

OPC-standardi on sarja teknisiä määritelmiä, joita kehittävät alan laitevalmistajat, ohjelmistokehittäjät ja loppukäyttäjät. Standardissa määritetään asiakas–palvelin ja palvelin–palvelin rajapinnat mukaan lukien reaaliaikaisen tiedon saanti, hälytysten ja tapahtumien monitorointi sekä historiallisten tietojen ja muiden sovellusten saatavuus. OPC Foundation vastaa standardin kehittämisestä ja ylläpidosta. (OPC Foundation, What is OPC.)

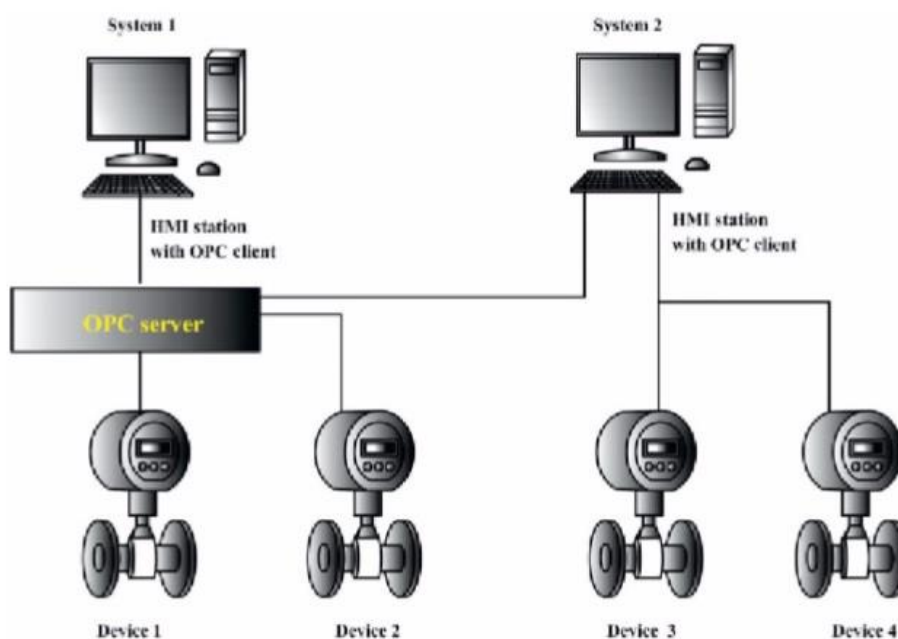
2.2.1 OPC:n historia

HMI- ja SCADA-toimittajat olivat 1990-luvun alkupuolella vastuussa omien järjestelmiensä ajurikirjastojen rakentamisesta. Järjestelmät olivat hyvin kattavia, mikäli kaikki käyttäjän vaatimat yhteysvaatimukset pystyttiin toteuttamaan, mutta vajanaisia mikäli kaikkia ominaisuuksia ei ollut. Järjestelmätoimittajien oli sijoitettava resursseja joko sovellustason toiminnallisuuteen tai kehittääkseen sovelluksien liitettävyyttä. Kuvassa 8 on esitetty prosessi ilman OPC-palvelimen hyödyntämistä. (Paine 2013, 1.)



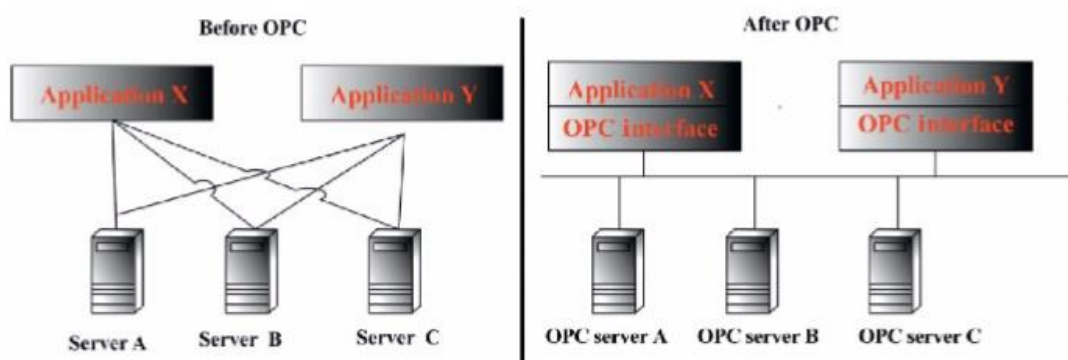
KUVA 8. Prosessi ilman OPC-palvelinta (Mehta & Reddy 2015, 460)

Osa järjestelmätoimittajista päätti kehittää omaa API-rajapintaa tai ajurityökalua, mikä täytti heidän liitettävyystarpeet, mutta rajoitti loppukäyttäjän mahdollisuuksia laajentaa järjestelmää. Pian asiakkaat suostuttelivat järjestelmätoimittajat yhteistyöhön keskenään, jotta toimitettavat järjestelmät täyttäisivät loppukäyttäjän kaikki tarpeet. Kuvassa 9 on esitetty prosessi, jossa hyödynnetään OPC-palvelinta. (Paine 2013, 1.)



KUVA 9. OPC-palvelinta hyödyntävä prosessi (Mehta & Reddy 2015, 460)

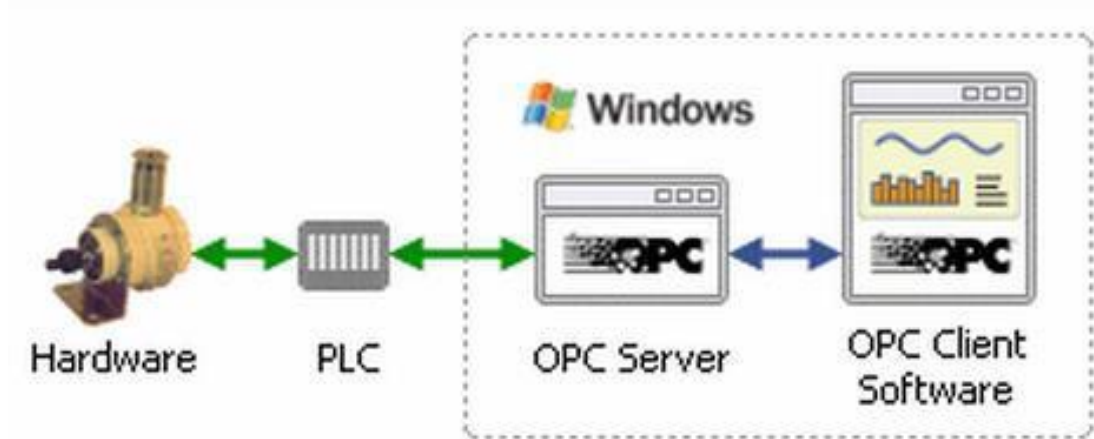
Alkuperäinen työryhmä sisälsi muutamia yrityksiä muun muassa Fischer-Rosemount, Intellution ja RockWell Software. Tuolloin kaikki ohjelmistokehitys tähtäsi Microsoft Windows alustaan. Microsoftin OLE asiakas–palvelin teknologiaa käytettiin tiedon jakamiseen sovellusten välillä hyödyntäen toimittajakohtaisia rajapintoja ja sääntöjä. Työryhmän alkuperäinen tavoite oli kehittää ratkaisu, joka yhdenmukaistaisi tiedonsiirron sovellusten ja erilaisten tietolähteiden välillä. Lopputuloksena oli ensimmäinen OPC-määrittely OPC Data Access vuonna 1996. Kuvassa 10 on sovellusten ja palvelimien välinen kommunikointi OPC-palvelimen avulla ja ilman sitä. (Paine 2013, 1.)



KUVA 10. Sovellusten ja palvelimien välinen kommunikointi (Mehta & Reddy. 2015, 461)

OPC lyhennettä kutsuttiin aluksi OLE for Process Control, mutta sen merkitys on muuttunut vuosien aikana markkinamuutoksen seurauksena. Microsoft tuotteisti OLE:n uudelleen COM:ksi sen jälkeen, kun OPC DA oli julkaistu. OPC:ta on myös alettu käyttää prosessiautomaation lisäksi monissa muissa automaatioympäristöissä. Näin ollen OPC Foundation koki tarpeelliseksi päivittää OPC:n kuvaus markkinaterminologiaan ja teknisiin sovelluksiin sopivaksi. OPC:tä kutsutaan nykyisin Open Connectivity via Open Standards eli avoin liitettävyys avoimilla standardeilla. (Paine 2013, 1.)

Jonkin aikaa OPC hyödynsi Microsoftin COM-teknologiaa. COM-teknologia oli perustana OPC Alarms and Events, OPC Historical Data Access ja eräille vähemmän käytetyille määrittelyille. Nämä yhdessä OPC Data Accessin kanssa muodostavat OPC Classicin. Kuvassa 11 on esitetty OPC Classicin käyttötapa. (Paine 2013, 1; OPC Foundation, What is OPC; OPC Datahub.)

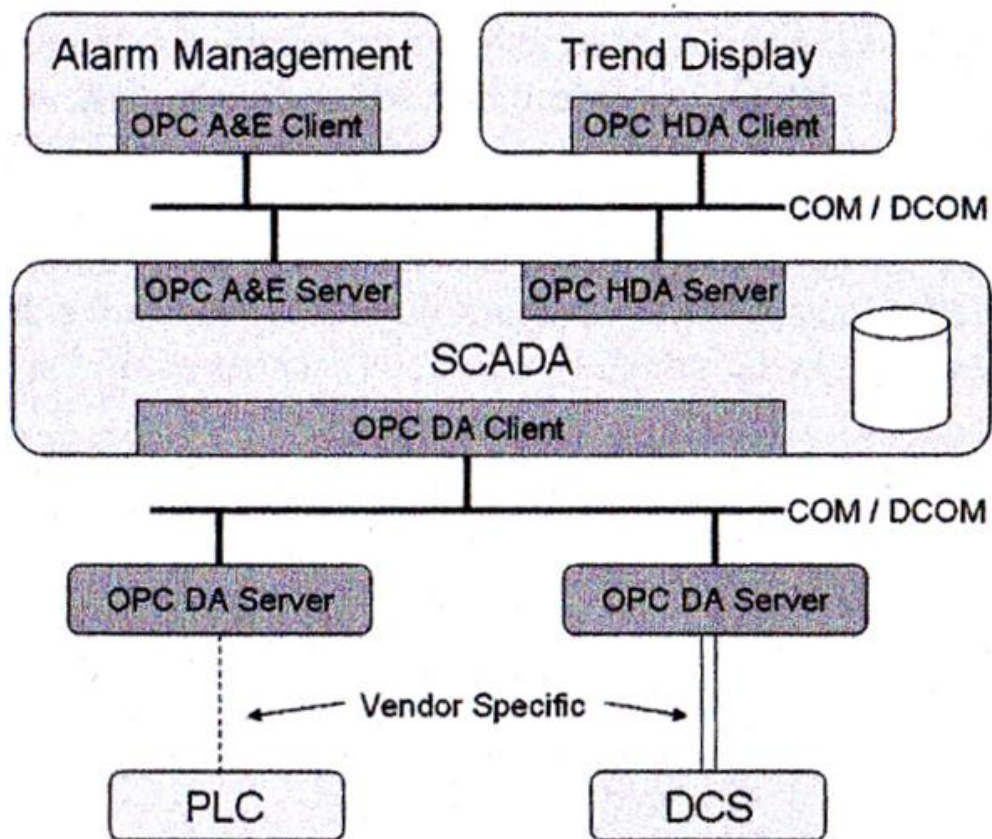


KUVA 11. OPC Classicin käyttötapa teollisuusympäristössä (OPC Datahub)

Palvelusuuntautuneiden arkkitehtuurien käyttöönotto valmistusjärjestelmissä aiheutti uusia haasteita tietoturvassa ja datan mallinnuksessa. OPC Foundation kehitti OPC Unified Architecture -määritelmät näiden tarpeiden täyttämiseksi. Samalla se tarjosi monipuolisen avoimen arkkitehtuurin, joka oli skaalautuva, laajennettavissa ja jatkokehittävissä tulevaisuuden tarpeisiin. OPC UA on saavuttanut IEC-standardi statuksen ja näin ollen tullut kansainväliseksi standardiksi. (OPC Foundation, What is OPC; Paine 2013, 2.)

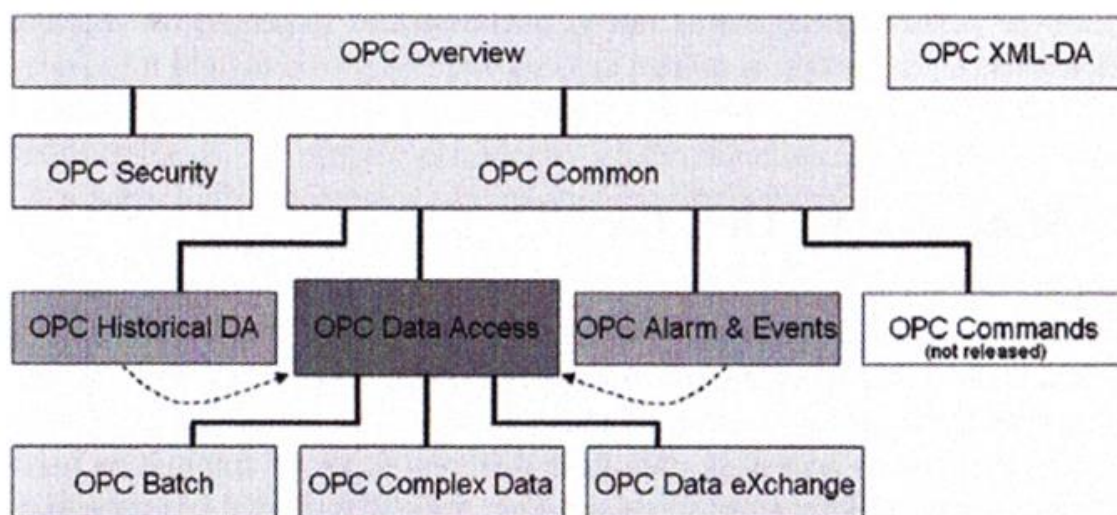
2.2.2 OPC Classic

OPC käyttää asiakas–palvelin tekniikkaa tiedon vaihtamiseen. OPC-palvelin tiivistää lähteen, esimerkiksi laitteen, lähettämän tiedon ja mahdollistaa siihen pääsyn rajapinnan kautta. OPC-asiakas ottaa yhteyttä OPC-palvelimeen käyttääkseen saatavana olevaa dataa. Asiakas ja palvelin kykenevät molemmat käyttämään ja vastaanottamaan dataa. Kuvassa 12 on esitetty tyypillinen OPC Classic käyttötapa. (Mahnke ym. 2009, 3.)



KUVA 12. Tyypillinen OPC asiakas–palvelin käyttötapa (Mehta & Reddy 2015, 463)

Teollisten sovellusten tarpeisiin on kehitetty kolme teknistä päämäärittelyä: Data Access (DA), Alarms & Events (A&E) ja Historical Data Access (HDA). Data Access tarjoaa reaaliaikaisen pääsyn prosessin dataan. Asiakas voi pyytää palvelimelta esimerkiksi viimeisimmät mitta-arvot virtauksesta, paineesta, lämpötilasta tai pinnankorkeudesta käyttäen OPC Data Accessia. Alarms & Eventsia käytetään eri toimijoiden väliseen tapahtumien ja hälytyksien vaihtoon. Historical Data Accessia käytetään arkistoidun datan noutamiseen ja käsittelyyn. Kuvassa 13 on esitetty kaikki Classic OPC-rajapinnan määrittelyt. (Mahnke ym. 2009, 3; Mehta & Reddy 2015, 463.)



KUVA 13. Classic OPC-rajapinnan määrittelyt (Mahnke ym. 2009, 7)

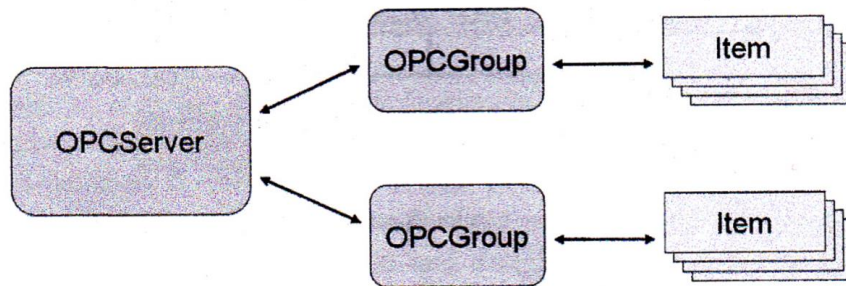
OPC Classic sisältää myös muita määrittelyitä, kuten OPC Data Exchange (DX), OPC Extensible Markup Language (XML), OPC Batch ja OPC Security (Mehta & Reddy 2015, 463).

OPC Data Access

OPC Data Access -rajapinta mahdollistaa muuttujien lukemisen, kirjoittamisen ja monitoroinnin tämän hetkisestä prosessidatasta. Data Access toimii rajapintana asiakas- ja palvelinohjelmille prosessitiedon välityksessä. Data Access -palvelin tarjoaa yhdelle tai useammalle Data Access -asiakkaalle läpinäkyvän saatavuuden tietolähteisiin. Tietolähteet on mahdollista kytkeä PC:n I/O-yksikköön tai väylän avulla ohjaimelle tai hajautusyksikköön. Data Access -asiakkaat voivat myös olla yhteydessä useampaan Data Access -palvelimeen samanaikaisesti. (Iwanitz & Lange 2002, 22; Mahnke ym. 2009, 4.)

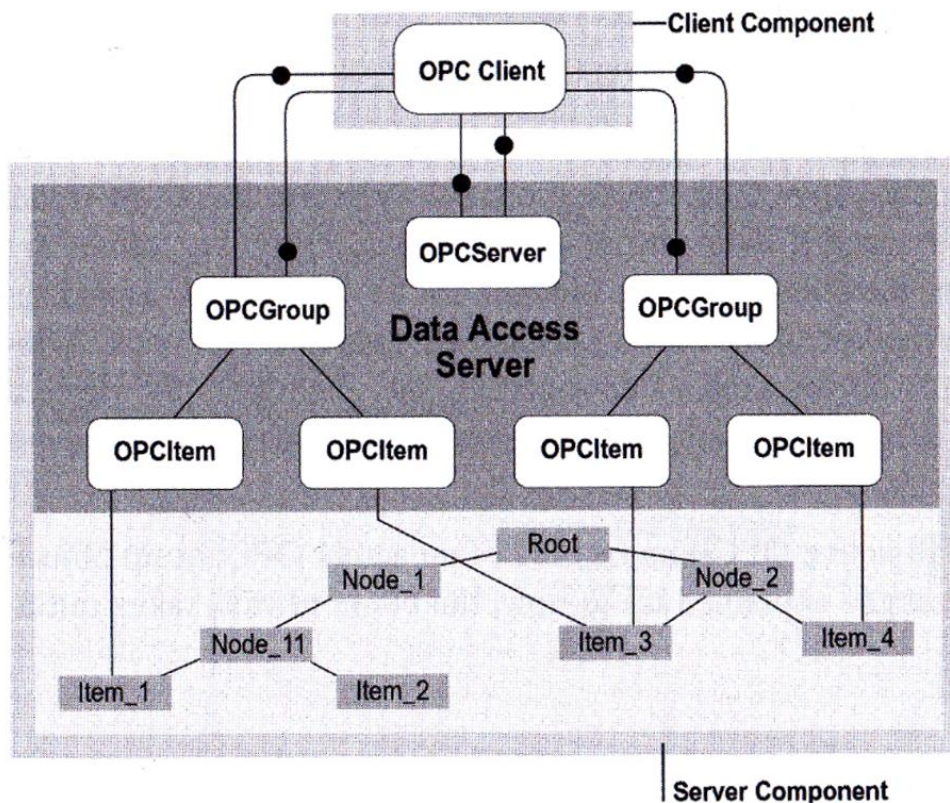
Data Access -asiakkaat voivat olla esimerkiksi yksinkertaisia Excel-sivuja tai monimutkaisia Visual Basic -visualisointeja. Data Access -asiakas voi myös olla osa isompaa käyttöliittymää tai valvomo-ohjelmistoa. Pääasiallinen käyttökohde on siirtää reaaliaikaista dataa ohjelmoitavista logiikoista, hajautetuista ohjausjärjestelmistä tai muista ohjauslaitteista käyttöliittymiin ja muihin valvomo-ohjelmistoihin. OPC Data Accessia käytetään suurimmassa osassa OPC-sovelluksista ja muut OPC-rajapinnat ovat lähinnä implementoitu järjestelmään Data Accessin lisänä. (Iwanitz & Lange 2002, 22; Mahnke ym. 2009, 4.)

OPC Data Access-asiakas valitsee tietolähteet (*engl. OPC Item*), joita halutaan lukea, kirjoittaa tai monitoroida palvelimelta. Asiakas muodostaa yhteyden Data Access palvelimelle luomalla OPC-palvelinolio (*engl. OPC Server Object*). Palvelinolio mahdollistaa menetelmät, joiden avulla tietolähde ja sen tiedot voidaan paikantaa osoitehierarkiasta. Kuvassa 14 on esitetty asiakkaan luomat tietolähteet ja oliot (Mahnke ym. 2009, 4).



KUVA 14. OPC-palvelimelle luodut tietolähteet ja oliot (Mahnke ym. 2009, 4)

Käsitelläkseen dataa OPC-asiakas ryhmittelee tietolähteet identtisillä asetuksilla OPC-ryhmäolioihin. Kun tietolähteet on ryhmitelty, voi asiakas lukea niitä tai kirjoittaa niihin. Kuvassa 15 on esitetty Data Access rajapinnan rakenne. (Iwanitz & Lange 2002, 23; Mahnke ym. 2009, 5.)



KUVA 15. Data Access -palvelimen olio- ja nimihierarkia (Iwanitz & Lange 2002, 23)

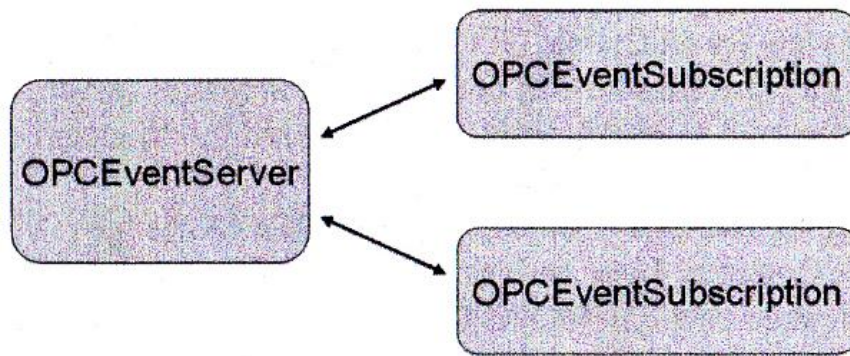
Suosittelava tapa asiakkaalle sykliseen datan lukemiseen on tarkkailla tietolähteiden arvon muutoksia palvelimella. Asiakas määrittelee päivitystaajuuden olioryhmälle käyttötapaan mukaan. Päivitystaajuutta käytetään palvelimella arvojen muutoksien tarkkailuun. Jokaisen päivityssyklin lopuksi palvelin lähettää vain muuttuneiden tietolähteiden arvot asiakkaalle. (Mahnke ym. 2009, 5.)

OPC-palvelin tarjoaa reaaliaikaista dataa. Data ei ole reaaliaikaista tilanteessa, jossa yhteys laitteeseen on katkennut. OPC Classic -teknologia tarjoaa dataan aikaleiman ja laatu-tiedon diagnostiikan parantamiseksi. Laatu-tiedot on lajiteltu kolmeen luokkaan oikea (*engl. Accurate*), ei saatavilla (*engl. Not Available*), tuntematon (*engl. Unknown*). (Mahnke ym. 2009, 5.)

OPC Alarms & Events

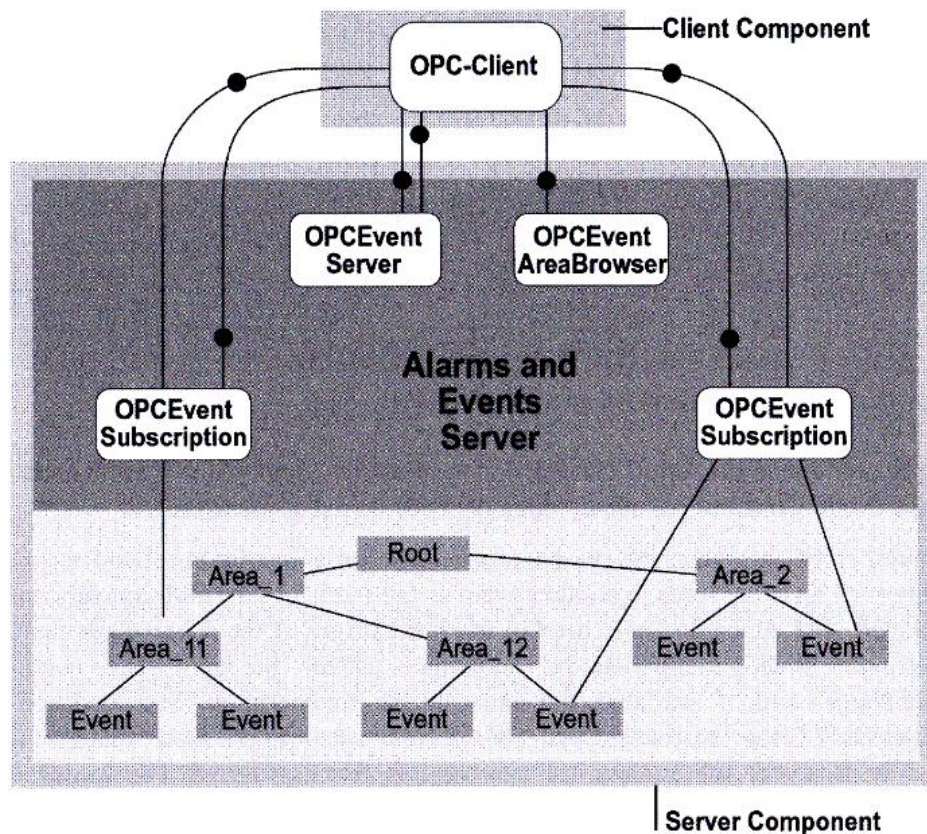
OPC Alarms & Events -rajapinta mahdollistaa tapahtuma- ja hälytysilmoitusten vastaanottamisen. Tapahtumat ovat yksittäisiä ilmoituksia, jotka informoivat asiakasta tapahtuman ilmenemisestä. Hälytykset ovat ilmoituksia, jotka informoivat asiakasta tilan muutoksesta prosessissa. Hälytyksiä ovat esimerkiksi tankin pinnankorkeuden ylitys tai alitus. Moni hälytyksistä vaatii kuittauksen, mikä on mahdollista toteuttaa OPC A&E -rajapinnan avulla. OPC A&E -palvelin voi tallentaa ja arvioida arvoja eri datalähteistä ja päättää, onko tapahtuma ilmennyt. (Iwanitz & Lange 2002, 43; Mahnke ym. 2009, 5.)

Vastaanottaakseen ilmoituksia OPC A&E-asiakas yhdistää palvelimelle ja tilaa ilmoitukset. Tämän jälkeen asiakas saa kaikki liipaistut (*engl. Trigger*) ilmoitukset palvelimelta. Rajatakseen ilmoitusten määrää, OPC-asiakas voi määrittää tietyt suodatuskriteerit ilmoituksille. OPC-asiakas yhdistää tekemällä OPC-tapahtumapalvelinolon (*OPC-Event-Server Object*) OPC A&E -palvelimelle, jonka jälkeen asiakas luo OPC-tapahtumatilauksen (*OPCEventSubscription*) ja alkaa vastaanottaa tapahtumaviestejä. Suodatukset tapahtumailmoituksille voidaan määrittellä tilauskohtaisesti. Kuvassa 16 on esitetty eri objektit, jotka OPC-asiakas voi luoda palvelimelle. (Mahnke ym. 2009, 5.)



KUVA 16. OPC-asiakkaan luomat oliot (Mahnke ym. 2009, 5)

OPC Data Accessiin verrattuna Alarms & Events -rajapinnassa ei ole tarkoin määritettyä pyyntöä halutulle informaatiolle. Kaikki prosessin tapahtumat tarjotaan asiakkaalle, josta asiakas voi rajoittaa tapahtumien määrää asettamalla suodatuskriteerejä esimerkiksi tapahtumatyyppin, -prioriteetin tai -lähteen mukaan. Kuvassa 17 on esitetty A&E-palvelimen rakenne. (Mahnke ym. 2009, 6.)



KUVA 17. Alarms & Events -palvelimen olio hierarkia ja tapahtuma-alue (Iwanitz & Lange 2002, 47)

Alarms & Events ja Data Access -palvelimet voivat käyttää samoja datalähteitä, mutta palvelimien välillä on eroja. Data Access -palvelin mahdollistaa jatkuvan datavirran.

Arvonmuutoksien automaattinen lähetys voidaan hyväksyä prosentuaalisista arvonmuutoksista. Tämä on kuitenkin mahdollista vain analogisille muuttujille, jotka eivät ole struktuurista datatyyppiä. Alarms & Events -palvelin ei lähetä arvoja asiakkaalle, vaan tiedon siitä, että jotain on tapahtunut. Tapahtumien raportointia varten palvelimelle on määriteltävä kriteerit, joilla tapahtuma ilmenee. Esimerkiksi lämpötilan saavutettua raja-arvon 40°C:sta. (Iwanitz & Lange 2002, 43.)

Alarms & Events -palvelimet voidaan ottaa käyttöön yksittäisinä komponentteina, jotka voivat saada tietoa suoraan laitteilta, sovelluksilta tai Data Access -palvelimilta. Data Access -palvelimilta tiedon vastaanottamista varten Alarms & Events -palvelimella tulee olla Data Access -asiakas ohjelmiston osia käyttöönotettuna. Alarms & Events -palvelin voidaan ottaa käyttöön Data Access -palvelimen osana. Alarms & Events -määrittely mahdollistaa useiden eri asiakkaiden ja palvelimien välisen kommunikoinnin. (Iwanitz & Lange 2002, 43-44.)

OPC Historical Data Access

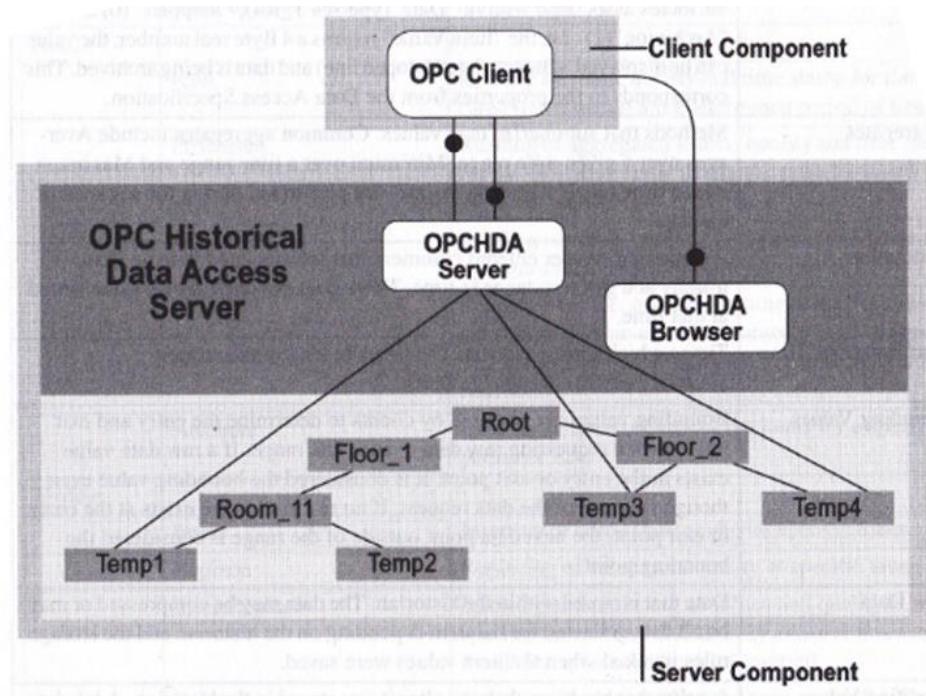
OPC Data Access tarjoaa jatkuvasti muuttuvaa reaaliaikaista dataa, mutta OPC Historical Data Access tarjoaa pääsyn aiemmin tallennettuun dataan. Datalla on kaksi perustyyppiä raakadata (*engl. Raw Data*) ja koottu data (*engl. Aggregate Data*). Raakadatalta tarkoitetaan tallennettua historiallista dataa ja kootulla datalla tarkoitetaan dataa, joka on poimittu raakadatasta. Koottu data muodostetaan vain asiakkaan pyynnöstä. Dataa voidaan käsitellä kirjoitus-, luku- tai muutosoikeuksin. Luku- tai muutosoikeuksin asiakas voi ylikirjoittaa ja kommentoida olemassa olevaa historiallista dataa tai lisätä uutta dataa. (Iwanitz & Lange 2002, 58; Mahnke ym. 2009, 6.)

Historical Data Access -määrittely mahdollistaa kahdenlaisten palvelinten käyttöönottamisen:

- Yksinkertaisten trendi datapalvelinten, jotka toteuttavat vain joitain valinnaisia rajapintoja sekä mahdollistavat pääsyn raakadataan.
- Monimutkaisten tietojen pakkaus- ja analyysipalvelinten, jotka mahdollistavat tietojen prosessoinnin, eheytyksen, merkintöjen lisäämisen ja tietojen muutoshistorian lukemisen.

(Iwanitz & Lange 2002, 58; Mahnke ym. 2009, 6.)

OPC-asiakas yhdistää palvelimelle tekemällä OPCHDA-palvelinolon (*engl. OPC-HDAServer*) HDA-palvelimelle. Tämä olio tarjoaa kaikki rajapinnat ja menetelmät lukea ja päivittää historiallista dataa. OPCHDA-selain (*engl. OPCHDABrowser*) määrittellään HDA-palvelimelle osoiteavaruuden selaamista varten. Kuvassa 18 on esitetty HDA-palvelimen nimiavaruus ja objektihierarkia. (Iwanitz & Lange 2002, 59; Mahnke ym. 2009, 6.)



KUVA 18. Historical Data Access -palvelimen nimiavaruus ja objektihierarkia (Iwanitz & Lange 2002, 59)

Historical Data Access -palvelimella ei ole olioita, jotka olisivat verrattavissa Data Access -palvelimen ryhmäolioihin (*engl. OPCGroup*) tai tietolähteisiin (*engl. OPCItem*). Historical Data Access -asiakas osoittaa suoraan tietolähteisiin käyttötärpeen mukaan ilman tähän tarkoitukseen luotuja olioita. (Iwanitz & Lange 2002, 58.)

Historical Data Accessin päätoiminnallisuus on lukea historiallista dataa kolmella eri tavalla:

- Ensimmäinen menetelmä on lukea raakadataa arkistosta, jonne asiakas määrittelee yhden tai useamman muuttujan sekä ajanjakson, jolta dataa halutaan lukea. Asiakas määrittelee arkistoitujen arvojen maksimimäärän, jonka palvelin palauttaa halutulta ajanjaksolta.

- Toinen menetelmä lukee yhden tai useamman muuttujan arvot määritellyllä aikaleimalla.
- Kolmas menetelmä kokoaa yhteen yhden tai useamman muuttujan arvot tietyltä ajanjaksolta historiallisesta tietokannasta. Lukumenetelmien lisäksi, OPC HDA määrittää menetelmät lisätä, korvata tai poistaa dataa historiatietoja sisältävästä tietokannasta.

(Mahnke ym. 2009, 6.)

2.2.3 OPC Unified Architecture

OPC Unified Architecture julkaistiin vuonna 2008. OPC UA on käyttöjärjestelmästä riippumaton, palvelukeskeinen arkkitehtuuri, joka yhdistää kaikkien OPC Classic -määrittelyiden toiminnallisuudet yhteen laajennettavaan rakenteeseen. OPC UA on palomuuuri ystävällinen ja monipuolisilla ominaisuuksillaan se tarjoaa ratkaisun tietoturvaongelmiin. Tietoturva on yksi OPC UA:n tärkeimmistä ominaisuuksista. (OPC Foundation, What is OPC UA.)

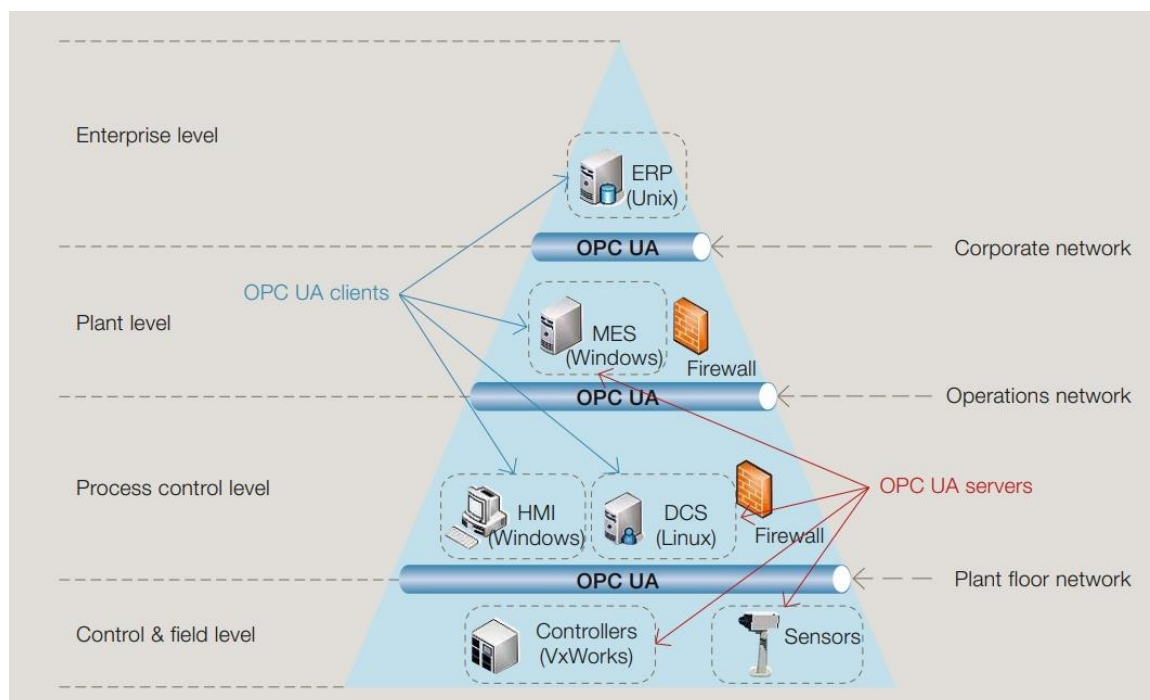
OPC Unified Architecture muodostui tarpeesta kehittää korvaaja kaikille COM-pohjaisille määrittelyille menettämättä ominaisuuksia tai suorituskykyä. Sen piti kattaa kaikki vaatimukset alustariippumattomasta rajapinnasta sisältäen laajennettavat mallinnusominaisuudet. OPC-palvelimen käyttökohteet vaativat skaalautuvuutta sulautetuista järjestelmistä, SCADA- ja DCS-, MES- ja ERP-järjestelmiin. (Mahnke ym. 2009, 8.)

Vaatimukset voidaan luokitella kahteen luokkaan. Ensimmäisessä luokassa käsitellään hajautettujen järjestelmien välistä viestintää ja toisessa tiedon mallintamista. Tärkeimmät OPC UA:n vaatimukset (taulukko 2). (Mahnke ym. 2009, 8.)

TAULUKKO 2. OPC UA:n tärkeimmät vaatimukset (Mahnke ym. 2009, 9)

Hajautettujen järjestelmien välinen kommunikointi	Tiedonmallinnus
<ul style="list-style-type: none"> • Luotettavuus <ul style="list-style-type: none"> ○ Käyttövarmuus ja vikasietoisuus ○ Redundanssi • Järjestelmäriippumattomuus • Skaalautuvuus • Korkea suorituskyky • Internet ja palomuuuri • Tietoturva ja pääsyn hallinta • Yhteentoimivuus 	<ul style="list-style-type: none"> • Yleinen tietomalli kaikelle OPC-tiedolle • Olio-malli • Laajennettavissa oleva järjestelmä • Metatiedon tarjoaminen • Kykenee käsittelemään monipuolista tietoa • Tarjoaa monipuoliset menetelmät tiedon käsittelyyn • Skaalautuvuus • Abstrakti perusmalli • Pohja muille standarditietomalleille

OPC UA:ta voidaan käyttää kaikilla käyttöjärjestelmillä ja sillä ei ole rajoitteita käytettävän laitteiston (*engl. Hardware*) suhteen. OPC UA:ta voidaan käyttää kaikkialla kenttälaitteiden ja toiminnanohjausjärjestelmän väliltä. Kuvassa 19 on esitetty OPC UA erilaisissa käyttökohteissa. (OPC Foundation; Paine 2013.)



KUVA 19. OPC UA -palvelimet eri käyttökohteissa (Mahnke & Leitner)

Monikerroksinen arkkitehtuuri on toteutettavissa myös tulevaisuudessa. Monipuoliset teknologiat ja menetelmät, kuten uudet siirtoprotokollat, tietoturva-algoritmit, koodausstandardit tai sovelluspalvelut, voidaan sisällyttää OPC UA:han säilyttäen olemassa olevien tuotteiden yhteensopivuus. (OPC Foundation, What is OPC UA; Paine 2013. Understanding OPC.)

OPC UA -informaatiomalli

Kun OPC Classic tarjoaa yksinkertaisen metamallin, jossa tunnisteet (*engl. Tag*) ovat yksinkertaisessa hierarkiassa, tarjoaa OPC UA monipuolisen informaatiomallin käyttäen oliopohjaisia tekniikoita. OPC UA kykenee esimerkiksi mittatiedon lisäksi kertomaan, minkä tyyppisestä lämpötila-anturista mittaustieto on peräisin. Tämän tyyppinen tieto on hyödyllistä monissa OPC Classic käyttötapauksissa. Tapauksissa samaa ohjelmistokomponenttia ja asetuksia, voidaan käyttää saman tyyppisille järjestelmässä olevilla laitteilla. Lisäksi samaa tietoa voidaan hyödyntää laajemmin eri sovellusalueella, kuten MES- ja ERP-järjestelmissä. Näin tietoja voidaan integroida ilman tarvetta välittää tunnistelistaa, joka kuvaa tunnisteiden semantiikan. OPC UA tarjoaa joustavuuden määrittellä monipuolisia informaatiomalleja, mutta ei vaadi niiden käyttämistä. (Mahnke & Leitner ABB review 3/2009.)

Suurin etu OPC UA:n käyttämisessä Classic OPC:hen verrattuna on, että se mahdollistaa informaationmallinnuksen ja helpottaa monia lisätoimia. OPC UA:ssa määritellään yksinkertainen peruskokoelma, jota voidaan laajentaa joko sovellus- ja toimittajakohtaisilla informaatiomalleilla tai standardi informaatiomalleilla. OPC UA:ssa idea on määrittellä, kuinka tietoa vaihdetaan, kuin taas standarditietomallissa määritellään mitä tietoa vaihdetaan. (Mahnke & Leitner ABB review 3/2009.)

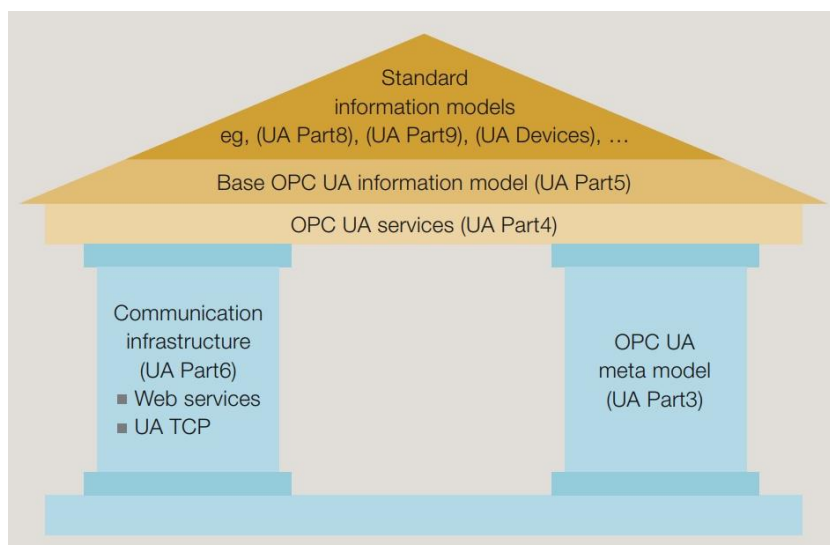
Tietomallinnuksen kehittyminen on aikaansaanut sen, että informaatiomalleja on alettu standardoimaan OPC UA:n pohjalta. Yleisesti käytetyt kenttälaitteet voivat käyttää standardoitua informaatiomallia, joka mahdollistaa valmistajasta riippumattoman ”plug and play” yhteentoimivuuden. Standardoitu informaatiomalli kehitettiin kenttälaitteintegroinnin pohjalta (*engl. Field Device Integration Initiative*) ja sitä on parannellut analysointilaitteintegraatio työryhmän toimesta (*engl. Analyzer Devices Integration*). (Mahnke & Leitner ABB review 3/2009.)

Vakioinformaatiomallien käyttö mahdollistaa yhteentoimivuuden uudella tasolla. Tiedonvaihdon yhteentoimivuuden lisäksi vakioinformaatiomallit mahdollistavat myös mallien yhteentoimivuuden. Pitkällä aikavälillä vakioinformaatiomallien käyttö voi vähentää merkittävästi suunnittelukustannuksia silloin, kun integroidaan järjestelmiä, joissa käytetään eri valmistajien tuotteita. (Mahnke & Leitner ABB review 3/2009.)

OPC UA:n skaalautuvuus mahdollistaa OPC UA -sovellusten toimimisen sekä rajoitetulla laitteistolla sulautetuissa järjestelmissä että myös tehokkaissa keskustietokoneissa. Tyypillisesti edellä mainitut järjestelmät eivät kuitenkaan tarjoa samaa tietoa. Sulautetun järjestelmän palvelin tarjoaa dataa lyhyeltä ajalta ja tukee vain muutamaa asiakasta, kun muut palvelimet voivat tarjota tietoa vuosien ajalta sekä tukea tuhansia asiakkaita. (Mahnke & Leitner ABB review 3/2009.)

Tietomallinnuksen kannalta OPC UA on skaalautuva. OPC UA -palvelin voi tarjota kaikkea OPC Classicin tapaisesta yksinkertaisesta mallista erittäin kehittyneisiin malleihin, jotka tarjoavat kehittyneitä metatietoja saadusta datasta. Asiakas voi käyttää yksinkertaista dataa tai hyödyntää palvelimen tarjoamaa metadataa. (Mahnke & Leitner ABB review 3/2009.)

OPC UA määrittää kaksi päämäärittelyä yhteentoimivuudelle, jotka ovat kommunikaatioinfrastruktuuri ja OPC UA-metamalli. Päämäärittelyt ovat esitetty kuvassa 20. Kommunikaatio infrastruktuuri määrittää, kuinka informaatiota vaihdetaan ja metamalli määrittää, mitä informaatiota vaihdetaan. (Mahnke & Leitner ABB review 3/2009.)

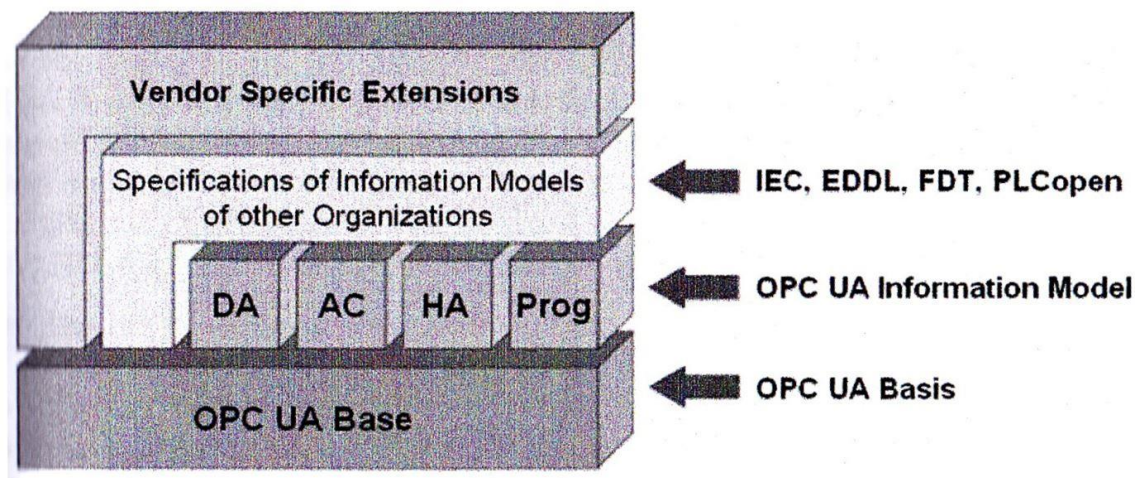


KUVA 20. OPC UA:n perusta (Mahnke & Leitner)

Kommunikaatioinfrastruktuurista riippumattomana, OPC UA määrittää joukon abstrakteja palveluita. Abstrakteja palveluita voidaan käyttää erilaisissa kommunikaatioinfrastruktuureissa. Metamallia käytetään pohjana määriteltäessä sopivia parametreja palveluille. (Mahnke & Leitner ABB review 3/2009.)

OPC UA-tietomalli

OPC UA-perustietomalli antaa perustiedot ja sisääntulopisteet palvelimen osoitealueeseen. Perustietomallin päälle voidaan rakentaa toimittajakohtaisia tai standardin mukaisia tietomalleja. OPC UA määrittää useita vakiotietomalleja tiedonsiirtoon, hälytyksiin, ehtoihin, ohjelmiin, historiatietoihin ja kokonaistoimintoihin. OPC UA tarjoaa myös toiminnallisuuksia tukemaan useita tietomalleja yhdellä palvelimella. Tietomallien tietoja voidaan lukea palveluilla. Asiakkaat, jotka käyttävät kyseisiä palveluita, pääsevät käsiksi kaikkeen informaatioon. Kuvassa 21 on esitetty OPC UA-tietomallin eri kerrokset. (Mahnke & Leitner ABB review 3/2009.)



KUVA 21. OPC UA:n kerroksittainen arkkitehtuuri (Mahnke ym. 2009, 11)

OPC UA ei ole suoraan yhteensopiva Classic OPC:n kanssa, koska se käyttää eri teknologiaa tiedonsiirtämiseen. Tämän vaatimuksen täyttämiseksi OPC Foundation ei pelkää tarjota ohjelmistoinfrastruktuuria OPC UA:n tiedonsiirtoon, vaan myös paketoi ja valtuuttaa jokaisen tietopaketin OPC UA-asiakkaille. Classic OPC -asiakkaat käsittelevät tietopaketteja OPC UA -palvelimelta välityspalvelimen avulla. (Mahnke & Leitner ABB review 3/2009.)

OPC UA -määrittely

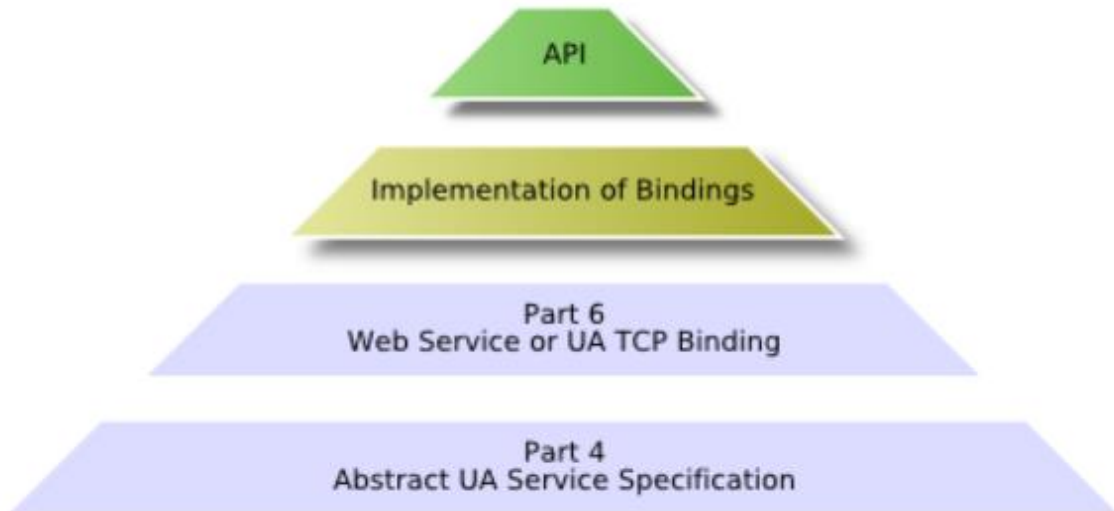
OPC UA määrittely on jaoteltu erilaisiin osiin, jotka vaaditaan IEC-standardointia varten. OPC UA tunnetaan IEC 62541 -standardina. Kuvassa 22 on esitetty yhteenveto kaikista määrittelyn osista. Määrittely on jaettu ydinmäärittelyihin, jotka määrittävät OPC UA:n perustan ja pääsyytyypin (*engl. Access Type*) mukaisiin määrittelyihin, jotka pääasiassa määrittävät OPC UA:n tietomallin. (Unified Automation.)



KUVA 22. OPC UA -määrittelyt (Unified Automation)

Standardin kaksi ensimmäistä osaa eivät ole normatiivisia: osa 1 kuvaa OPC UA:ta yleisellä tasolla ja osa 2 kuvaa tietoturva-vaatimukset ja OPC UA:n turvallisuusmallin. Osat 3 ja 4 ovat tärkeimmät määrittelyt OPC UA:n sovelluksien suunnittelun ja kehityksen kannalta. (Mahnke ym. 2009, 12.)

Osoiteavaruusmallissa osa 3 määritetään rakennuspalikat, joilla paljastetaan (*engl. Expose*) tietomalleja ja rakennetaan OPC UA -palvelimen osoiteavaruus. Palvelut osassa 4 määritetään abstraktit UA-palvelut, joilla kuvataan UA-asiakkaan ja UA-palvelimen ohjelmistojen välistä vuorovaikutusta. Asiakas käyttää palveluita löytääkseen ja käsitelläkseen tietoa, jota palvelin tarjoaa. Kuvassa 23 on esitetty OPC UA:n kerroksiksi pinottu tiedonsiirtoarkkitehtuuri. (Unified Automation.)



KUVA 23. Kerrostettu OPC UA:n tiedonsiirtoarkkitehtuuri (Unified Automation)

Perustietomalli on määritetty standardin osassa 5 (kuva 22). Se tarjoaa rungon kaikille tietomalleille, jotka käyttävät OPC UA:ta. Siinä määritetään seuraavat asiat:

- Asiakkaiden pääsykohdan OPC UA -palvelimen osoiteavaruuteen osoitteiden ja tyyppien selaamista varten (*engl. Type*).
- Perustyyppit, jotka rakentavat pohjan erityyppisille hierarkioille.
- Sisäänrakennetut mallit, jotka ovat laajennettavia, kuten objekti- ja tietotyyppit.
- Palvelinobjekti, joka tarjoaa valmius- ja diagnostiikkatietoja.

(Mahnke ym. 2009, 13.)

Osassa 6 (kuva 22) määritetään viestien konkreettinen tiedonsiirtoväylä sekä kuinka UA-palvelut ja turvallisuusmenetelmät sovelletaan käytettäväksi viesteihin. Vain OPC UA-pinon toteuttajien (*engl. Implementers*) tarvitsee täysin ymmärtää tämä määrittely. OPC-säätiö tarjoaa oikeat UA-pinot. OPC UA -sovellusarkkitehdin tai -ohjelmoijan ei tarvitse lukea tätä määrittelyä. (Unified Automation.)

Osassa 7 (kuva 22), profiilit, määritellään hyödyllisiä alajoukkoja (*engl. Subset*) OPC UA-ominaisuuksille. Tällainen alaryhmä täytyy toteuttaa kokonaisuudessaan OPC UA -sovellukseen varmistaakseen yhteentoimivuuden alajoukon kanssa. Alajoukot määritetään kahdelle eri tasolle. Ensimmäisellä tasolla ovat vaatimustenmukaisuusyksiköt, jotka määrittelevät pienen joukon toimintoja. Toimintoja käytetään aina yhdessä ja niitä voidaan testata vaatimustenmukaisuuden testaustyökaluilla ja todentaa yksiköksi. Toinen taso ovat profiilit, jotka koostuvat vaatimustenmukaisuusyksiköiden luettelosta. Profiili

on toteutettava kokonaan, ja se todetaan täydelliseksi asetukseksi OPC UA -tuotteiden sertifiointin aikana. Tuettujen ja käytettyjen profiilien luettelo vaihdetaan yhteyden muodostuksen aikana asiakkaan ja palvelimen välillä. Samalla myös sovellus ottaa selvää, onko kommunikaatiopartnerilla tarvittavat ominaisuudet tuettuna. (Mahnke ym. 2009, 13.)

Muita standardin osia ovat:

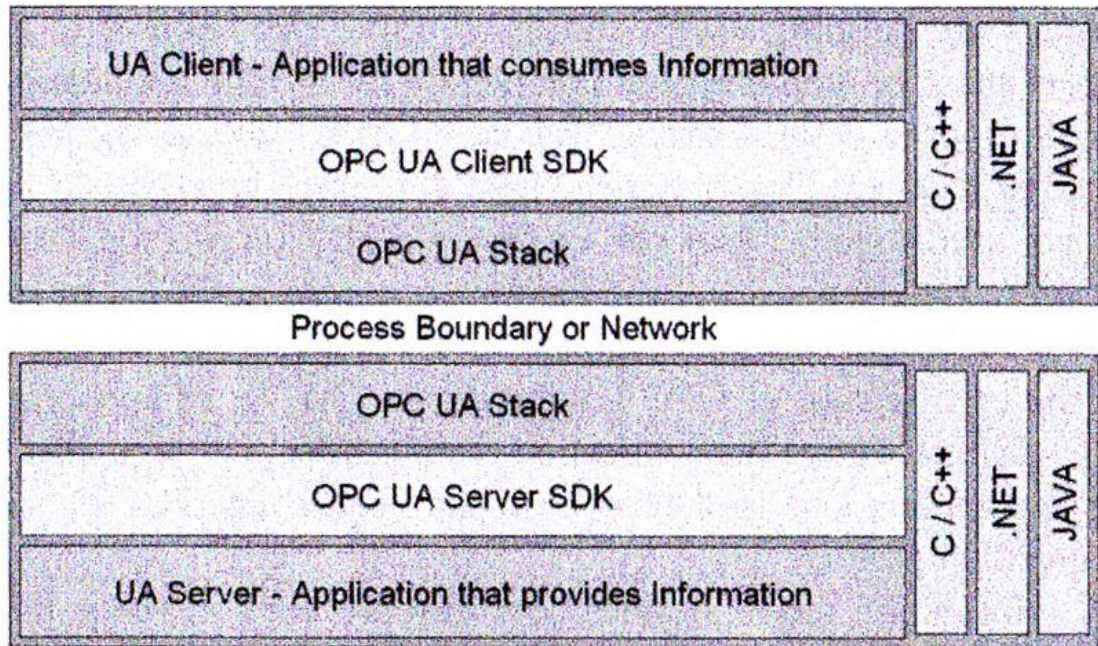
- Osa 8 (kuva 22), Data Access -tietomalli, määrittää, kuinka automaattitietoa ja ominaisuuksia tulee esittää ja käyttää.
- Osa 9 (kuva 22), Alarm and Conditions -tietomalli, määrittää, prosessin hälytys- ja ehtoseurannan.
- Osa 10 (kuva 22), Programs -tietomalli, määrittää, perustilakoneen, ohjelmien suorittamista, käsittelyä ja seuranta varten.
- Osa 11 (kuva 22), Historical Access -tietomalli, määrittää, kuinka historiatietopalveluja (*engl. Historical Data Access*) käytetään ja miten tämän hetkisiä konfiguraatitietoja ja tapahtumahistoriaa esitetään.
- Osa 12 (kuva 22), Discovery, määrittää, kuinka palvelimia löydetään verkosta ja kuinka asiakas saa tarvittavat tiedot yhdistääkseen tiettyyn palvelimeen.
- Osa 13 (kuva 22), Aggregates, määrittää, kuinka koottuja arvoja lasketaan raakadatanäytteistä. Koottuja (*engl. Aggregates*) arvoja käytetään nykyisten- ja historiallisten arvojen seurantaan.

(Unified Automation.)

OPC UA -ohjelmistokerrokset

OPC UA käyttää samantapaista asiakas–palvelin-konseptia, kuin Classic OPC. Sovellusta, joka haluaa paljastaa omaa tietoaan muille sovelluksille, kutsutaan UA-palvelimeksi. UA-asiakkaaksi kutsutaan sovellusta, joka haluaa käyttää tietoa toisilta sovelluksilta. Pääsääntöisesti voidaan olettaa, että UA-palvelin ja UA-asiakas sisältää enemmän sovelluksia kuin Classic OPC sisältäisi. Yksi syy tähän on, että UA-palvelimia integroidaan suoraan laitteisiin. Toinen syy on OPC UA:n käyttäminen kommunikointirajapintana, missä UA-asiakkaat ovat myös UA-palvelimia, jotka konfiguroidaan OPC UA:n kautta. UA-asiakkaan käyttö mahdollistaa myös laite–laite tyyppisen kommunikoinnin. (Mahnke ym. 2009, 13-14.)

Tyypillinen OPC UA-sovellus on koottu kolmesta ohjelmakerroksesta, jotka ovat esitetty kuvassa 24. Koko ohjelmistopino voidaan toteuttaa C/C++, .NET tai JAVAn avulla. OPC UA ei rajoitu näihin ohjelmointikieliin tai kehitysympäristöihin, mutta tällä hetkellä vain näillä ympäristöillä voidaan hyödyntää OPC Foundation tarjoamaan peruspinoa. (Unified Automation.)



KUVA 24. OPC UA:n ohjelmistokerrokset (Mahnke ym. 2009, 14)

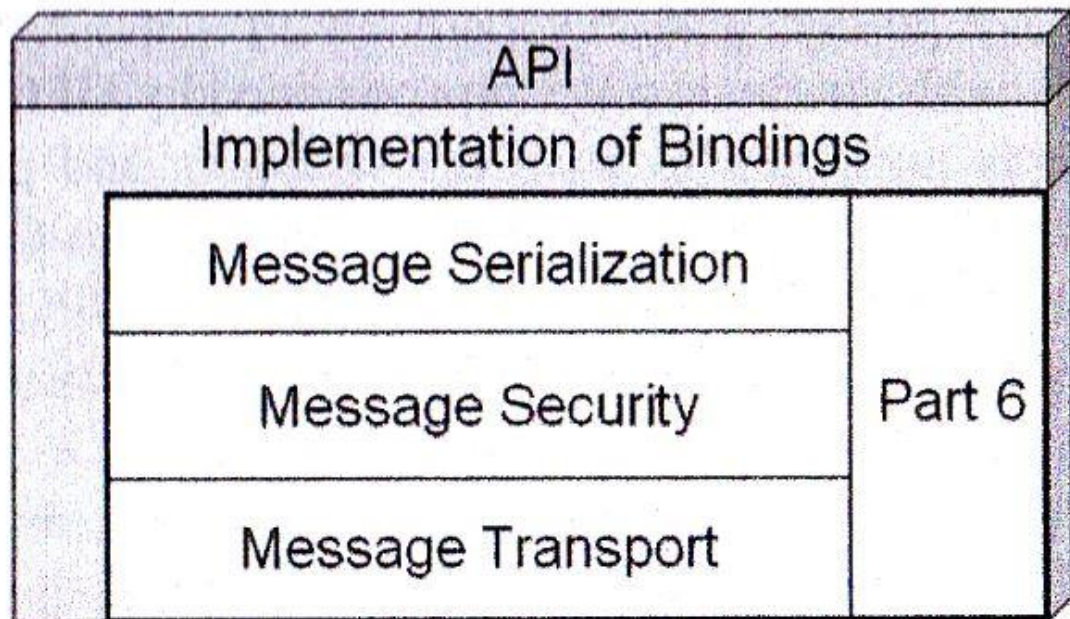
OPC UA-sovellus on järjestelmä, joka haluaa paljastaa tai käyttää tietoja OPC UA:n kautta. Sovellus sisältää erityistoiminnon, joka kartoitetaan OPC UA:lle käyttäen OPC UA-pinoa ja OPC UA -ohjelmakehitystyökalupakettia (*engl. Software Development Kit*). (Mahnke ym. 2009, 14)

OPC UA-asiakas- tai -palvelinohjelmistokehitystyökalupaketti toteuttaa yleisiä OPC UA-toiminnallisuuksia, jotka ovat osa sovelluserrosta. OPC UA-pino toteuttaa vain tiedonsiirtokanavia. OPC UA SDK vähentää kehitykseen tarvittavia resursseja ja helpottaa yhteentoimivuutta OPC UA-sovellukselle. (Unified Automation.)

OPC UA-pino toteuttaa erilaisia tiedon kuljetukseen tarkoitettuja kartoituksia, jotka ovat määriteltynä UA-määrittelyn osassa 6 (kuva 22). Pinoa käytetään kutsumaan UA-palveluita prosessi- tai verkkorajojen yli. OPC UA määrittää kolme pinokerrosta ja jokaiselle kerrokselle erilaisia profiileja:

- Viestien enkoodaus kerros (*engl. Message Serialization*) määrittää palveluparametrien sarjanumeroinnin binaari- ja XML-formaatissa.
 - Viestien tietoturvaso (*engl. Message Security*) määrittää, miten viestit on suojattava käyttämällä verkkopalvelun turvallisuusstandardeja (*Web Service Security Standards*) tai verkkopalvelu standardin UA-binääriverstiä.
 - Viestien kuljetuskerros (*engl. Message Transport*) määrittää käytettävän verkko-protokollan, mikä voi olla UA TCP tai HTTP ja SOAP verkkopalveluille.
- (Mahnke ym. 2009, 14-15.)

Kuva 25 havainnollistaa OPC UA:n eri kommunikointipinon kerroksia. Kerroksien implementointi UA-pinossa ja -sovelluksille käytettävä ohjelmointirajapinta eivät kuulu OPC UA -määrittelyyn. UA-pino tarjoaa kieliriippuvaisia ohjelmointirajapintoja UA-asiakas ja UA-palvelin sovelluksille. Palvelut ja niiden parametrit ovat samankaltaisia ja pohjautuvat abstraktiin palvelumäärittelyyn OPC UA osa 4 (kuva 23). (Mahnke ym. 2009, 14-15.)



KUVA 25. OPC UA:n kommunikaatiopinon kerrokset (Mahnke ym. 2009, 15)

Pääkehitysympäristöt ja ohjelmointikieliset (ANSI C/C++, .NET ja JAVA) kuuluvat OPC Foundationin kehittämään ja ylläpitämään UA-pinoon. (Mahnke ym. 2009, 15).

2.3 Tiedonhallinta ja varastointi

Yritykselle tieto on tärkeä resurssi ja sen varastoiminen vaatii investointeja. Tietokantatekniikkaa käytetään tietojen tallentamiseen kaikissa tärkeissä tietojärjestelmissä. Yritykset ovat monesti täysin riippuvaisia tietokannoista, joten on tärkeää, että tiedot on tallennettu selkeässä muodossa. Tieto tulee olla nopeasti saatavissa ja sen tulee olla erilaisiin tarpeisiin soveltuvaa. (Hovi, Huotari & Lahdenmäki 2005, 4.)

Tiedonhallinta tarkoittaa kaikkea tiedon määrittelemiseen ja käyttämiseen liittyvää toimintaa. Vastaavanlaisia esimerkkejä olisi yrityksen materiaali tai henkilöstö. Molemmat ovat yritykselle tärkeitä resursseja, joten puhutaan materiaalinhallinnasta ja henkilöstön hallinnasta. Tietoresurssin hallinta on tiedonhallintaa. (Hovi ym. 2005, 4.)

Tekniikan käyttöönoton myötä yritykset keräävät suuria määriä dataa. Datan kerääminen on tullut halvemmaksi ja helpommaksi esimerkiksi viivakoodien ja lukijoiden avulla. Datan määrä maailmassa arvioidaan kaksinkertaistuvan 20 kuukauden välein. Monet suuret yritykset käsittelevät rutiininomaisesti teratavuja dataa. Tiedonhallinnasta on tullut keskeinen toiminto monille organisaatioille. (Watson 2006, 1.)

Tiedonhallinta on vanha ongelma, mutta siitä on tullut tärkeämpää nykyaikaisessa taloudessa. Tiedonkerääminen on tärkeää organisaatioille, jotta ne voivat toimia tehokkaasti. Organisaatioiden tulee hallita tietojaan niin, että liiketoiminnan harjoittamiseen tarvittavat tiedot ovat saatavilla. (Watson 2006, 23.)

Data ja informaatio termitä käytetään usein synonyymeinä, mutta ne tarkoittavat eri asioita. Data on käsittelemätöntä ja analysoimatonta tietoa ja informaatio on dataa, joka on käsitelty merkitykselliseen muotoon. Tieto (*engl. Knowledge*) on kapasiteetti käyttää informaatiota. (Watson 2006, 24, 27.)

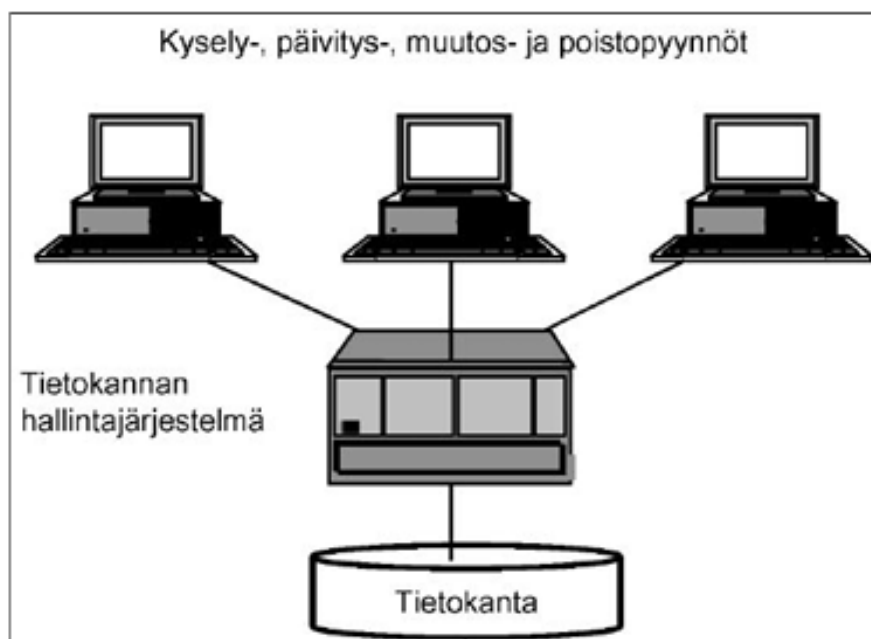
Organisaatioilla on suuri haaste käyttää reaaliaikaista dataa tehokkaasti moniulotteisissa tiedonhallintajärjestelmissään. Ongelma muodostuu siitä, etteivät järjestelmät ole integroituja keskenään, eivätkä käyttäjät ole saaneet riittävää koulutusta järjestelmiin tai eivät ole tietoisia niiden olemassaolosta. Ennen kuin ongelma voidaan ratkaista, täytyy ymmärtää, kuinka tietokantoja käytetään organisaatioissa. (Watson 2006, 25.)

Organisaatioiden täytyy ylläpitää tietokantoja liiketoiminnan käsittelyyn ja päätöksentekoon. Organisaation data tulisi olla jaettavissa tarkasti ja turvallisesti sekä helposti siirrettävissä. Organisaation data tulisi tarjota ajankohtaista ja olennaista tietoa. Tiedon hallitsemiseen voidaan käyttää monenlaisia eri tekniikoita. Ulkopuoliset tietokannat laajentavat organisaatioiden käytettävissä olevaa dataa. (Watson 2006, 27.)

2.3.1 Tietokanta

Tietokanta (*engl. Database*) on käsitteenä moniselitteinen ja osittainen tuotekohtainen. Tietokanta on kokoelma loogisesti yhteenkuuluvaa tietoa, joka on tallennettu siten, että se on usean käyttäjän käytettävissä eri tarkoituksiin ja jota voidaan helposti käsitellä tietokantakielellä. Tietokannan hallintajärjestelmä (*engl. Database Management System, DMBS*) on erityinen ohjelmisto, jolla hallinnoidaan tietokannassa olevaa tietoa. Muutamia tunnetuimpia tietokannan hallintajärjestelmiä ovat Oracle, DB2, Microsoft SQL Server, My SQL ja Access. (Pathak 2008, 1; Hovi ym. 2005, 4.)

Tietokannan hallintajärjestelmät ovat isoja ja monimutkaisia ohjelmistoja, jotka tarjoavat monipuolisia palveluja ohjelmoijille ja käyttäjille. ”Tiedot tallennetaan tietokantaan muutosjoustavuuden lisäämiseksi, tietoeheyden turvaamiseksi, suorituskyvyn parantamiseksi sekä sovellusohjelmoinnin helpottamiseksi.” (Hovi ym. 2005, 4.) Jos tietokannan hallintajärjestelmää ei olisi käytettävissä, jouduttaisiin käyttämään tiedostoja. Tuolloin monimutkaisten tietokokonaisuuksien ohjelmointi olisi työläämpää, tietokannan sisällön eheys olisi heikkoa ja tietojen hakeminen olisi vaikeaa. Kuvassa 26 on esitetty tiedonhallintaa kuvaava järjestelmä. (Hovi ym. 2005, 4.)



KUVA 26. Tietokannan hallintajärjestelmä käsittelee tietokantaan kohdistuvat toimenpiteet (Hovi ym. 2005, 5)

Tietokannan hallintajärjestelmä mahdollistaa sen, että useat käyttäjät voivat käyttää yhteistä tietokantaa ja yhden käyttäjän tekemät päivitykset näkyvät heti muille käyttäjille. Lisäksi tiedot tallennetaan vain kerran, jolla vältetään tiedon turhaa tallentamista. Tietokannan hallintajärjestelmä takaa myös, että tieto on eheää ja reaaliaikaista. (Hovi ym. 2005, 5.)

2.3.2 Erityyppiset tietokannat

Ennen tietokannat olivat rakenteeltaan verkkomallisia, kuten esimerkiksi Tota, IDMS, DMIV ja MDBS ja tietokannan hallintajärjestelmät olivat hierarkkisia, kuten esimerkiksi DL/1. Nykyisin tietokannat ovat pääasiassa SQL-pohjaisia relaatiotietokantoja (*engl. Relational Database, RDB*). Relaatiotietokannat ovat helpompia muuttaa ja käyttää kuin aiemmat hierarkkiset ja verkkomalliset tietokannat. Relaatiotietokannoissa oli alkuaikoina suorituskykyongelmia, mutta nykyisin kaikkein vaativimmatkin järjestelmät on mahdollista toteuttaa relaatiotietokannoilla. Relaatiotietokannoilla toteutetaan operatiivisia sovelluksia sekä tietovarastoja (*engl. Data Warehouse*). (Hovi ym. 2005, 5.)

On olemassa niin sanottuja oliotietokantoja (*engl. Object Database*), jotka eivät noudata relaatiomallia ja normalisointisääntöjä. Niillä on mahdollista luoda monimutkaisia olioita ja niille sovellettavia toimintoja. Oliotietokannat sulautuvat hyvin olio-ohjelmoinnin jatkoksi, mutta ne eivät ole saavuttaneet merkittävää osuutta kaupallisissa sovelluksissa. Relaatiotietokantoihin on myös tullut olio-ominaisuuksia, jolloin niistä käytetään termiä oliorelaationaalinen tietokanta (*engl. Object-Relational Database, ORDB*). Tällaisia ominaisuuksia ovat esimerkiksi vektorit, osoittimet taulujen välillä ja tauluun sijoitettavat taulut. (Hovi ym. 2005, 6.)

Tietovarastoinnissa käytetään paljon tavallisia relaatiokantatuotteita. Tämän lisäksi käytössä on myös erikoistietokantoja, jotka tukevat hyvin moniulotteisen tiedon kyselyjä. Erikoistietokantoja ei ole tarkoitettu yleiseen tai operatiiviseen käyttöön. Moniulotteisia tietokantoja (*engl. Multi-Dimensional Database*) kutsutaan myös OLAP-kannoiksi (*engl. Online Analytical Processing*) tai kuutioiksi (*engl. Cube*). Tekstin tallennukseen ja dokumenttien arkistointiin on kehitetty omia tietokantatyyppejään. Multimediakannat kykenevät tallentamaan myös kuvia, ääntä tai videokuvaa. Muita erikoistuneita tietokantatyyppejä ovat aika- ja paikkatietokannat. (Hovi ym. 2005, 6.)

3 TUOTANNONOHJAUS

Tuotantoprosessin päätöksentekoa ohjataan tuotannonohjauksella niin, että tuotteita tuotetaan vaadittava määrä sovittuun ajankohtaan mennessä. Tuotannonohjauksella minimoidaan tuotantoprosessiin kuluvat resurssit. Määrällisten ja ajallisten vaatimusten lisäksi tuotteiden tulee olla laatuvaatimusten mukaisia. Laatustandardeja hyödyntämällä yritys varmistaa tuotteiden kilpailukykyisen laadun. (Kletti 2007, 8.)

Tuotannonohjauksen tulee perustua kysynnän tarpeisiin. Mikäli yritys ei vastaa kysyntään, sen tarjoamat tuotteet loppuvat. Liiketoiminnan pysähtyminen aiheuttaa tappioita yritykselle niin kauan, kunnes tuotteita saadaan tuotettua lisää markkinoille. Ylituotanto tuottaa ylimääräistä tarjontaa, joka aiheuttaa varastotasojen liiallisen kasvun. Yrityksen oma pääoma kasaantuu varastoon ja sen likvidointi hankaloituu. (Kletti 2007, 9.)

3.1 Tuotannonohjausjärjestelmä

Tuotannonohjausjärjestelmä mahdollistaa tuotannon seurannan ja hallinnan sekä valmistusprosessien yhdenmukaistamisen, kehittämisen ja tehostamisen. Toimitusaikojen ja eräkokojen pienentyessä on tärkeää, että tieto kulkee valmistusketjussa ongelmitta. Tuotannonohjausjärjestelmä tarjoaa eri osapuolille luotettavaa tietoa tuotantoketjun eri osista ja kokonaisuuden toimivuudesta. Tuotantoa täytyy seurata reaaliajassa ja tuotantoketjun muutoksiin on kyettävä reagoimaan nopeasti. Perinteinen toiminnanohjaus ei aina pysty vastaamaan näihin tarpeisiin, vaan avuksi tarvitaan vaativissa tuotanto-olosuhteissa toimiva ja tuotannon erikoistarpeisiin kohdennettu ohjausjärjestelmä. (Digia.)

Toimiva tuotannonohjausjärjestelmä (*engl. Manufacturing Execution System, MES*) yhdistää tehokkaasti tuotannonohjauksen ja tiedonkeruun yrityksen muihin tietojärjestelmiin. Tuotannonohjausjärjestelmällä voidaan hallita tuotannon operatiivisia toimintoja, kerätä tuotantodataa, muodostaa ja ylläpitää tuotannon materiaalivirtojen eräseuranta- ja jäljitettävyyshetjua sekä välittää tietoa toiminnanohjausjärjestelmän (*engl. Enterprise Resource Planning, ERP*) ja tehdasautomaatiojärjestelmien välillä. (Digia.)

Tuotannonohjausjärjestelmä ja valmistuksenohjaus (*engl. Manufacturing Operations Management, MOM*) -termejä käytetään usein synonyymeinä ja lähteestä riippuen niiden

toiminnallisuus määritellään eri tavoin. Eräille tuotannonohjausjärjestelmä on osa valmistuksenohjausta. Olisi selkeämpää, jos käytettäisiin vain yhtä termiä, sillä kahden eri termin käyttäminen aiheuttaa hämmennystä.

Osa järjestelmätoimittajista tarjoaa tuotannonohjausjärjestelmää, joka sisältää kaikki valmistuksenohjauksen osa-alueet. Osalla järjestelmätoimittajista tuotannonohjausjärjestelmä on karsitumpi. ISA-95 -standardi määrittää järjestelmähierarkiassa valmistuksenohjauksen käsittämään koko tasoa 3.

3.2 ISA-95 -standardi

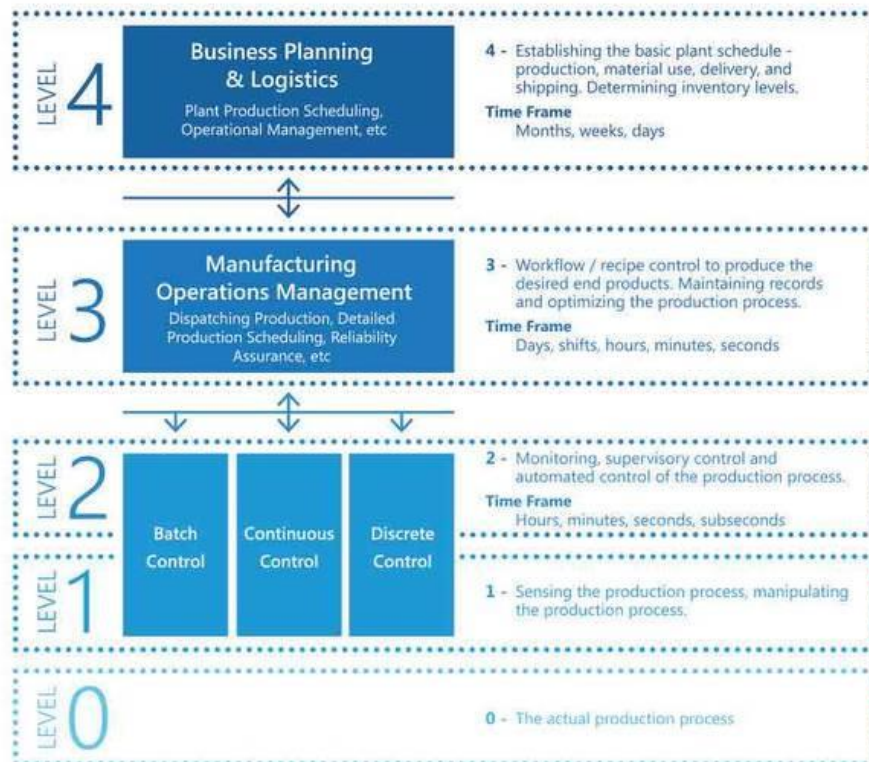
Yksi tärkeimmistä MES-teknologian kehitykseen vaikuttaneista määrittelyistä oli ISA-95 -standardi, jonka kansainvälinen automaatioyhdistys (*engl. International Society of Automation, ISA*) kehitti yhteistyössä tuotantoyhteisöjen ja järjestelmätoimittajien kanssa. ISA-95 on myös saatavana IEC/ISO 62264 standardina. (Almada-Lobo 2015; Brandl 2008, 3.)

ISA-95 -standardi voidaan jakaa kolmeen pääalueeseen:

1. Toiminnanohjaus- ja valmistuksenohjausjärjestelmien välinen tiedonsiirto (standardin osat 1, 2 ja 5).
2. Valmistuksenohjausjärjestelmän toimintamallit (standardin osa 3).
3. Valmistuksenohjausjärjestelmien sisäiset tiedonsiirtomallit (standardin osat 4 ja 6)

(Brandl 2008, 3.)

ISA-95 käsittelee MES-termiä osana isompaa valmistuksenohjaus -käsitettä, mutta molemmat (MES ja MOM) mallintavat tiedonvaihtoa toiminnanohjaus- ja valmistuksenohjausjärjestelmien välillä. Ne mallintavat myös tiedonsiirtoa valmistuksenohjauksen ohjelmistojen välillä, ja mitä toimintoja ohjelmistot voivat suorittaa. Kuvassa 27 on valmistuksen päätöksenteossa käytettävä ISA-95 -standardin mukainen toiminnallinen hierarkia. (Almada-Lobo 2015; Brandl 2008, 3.)

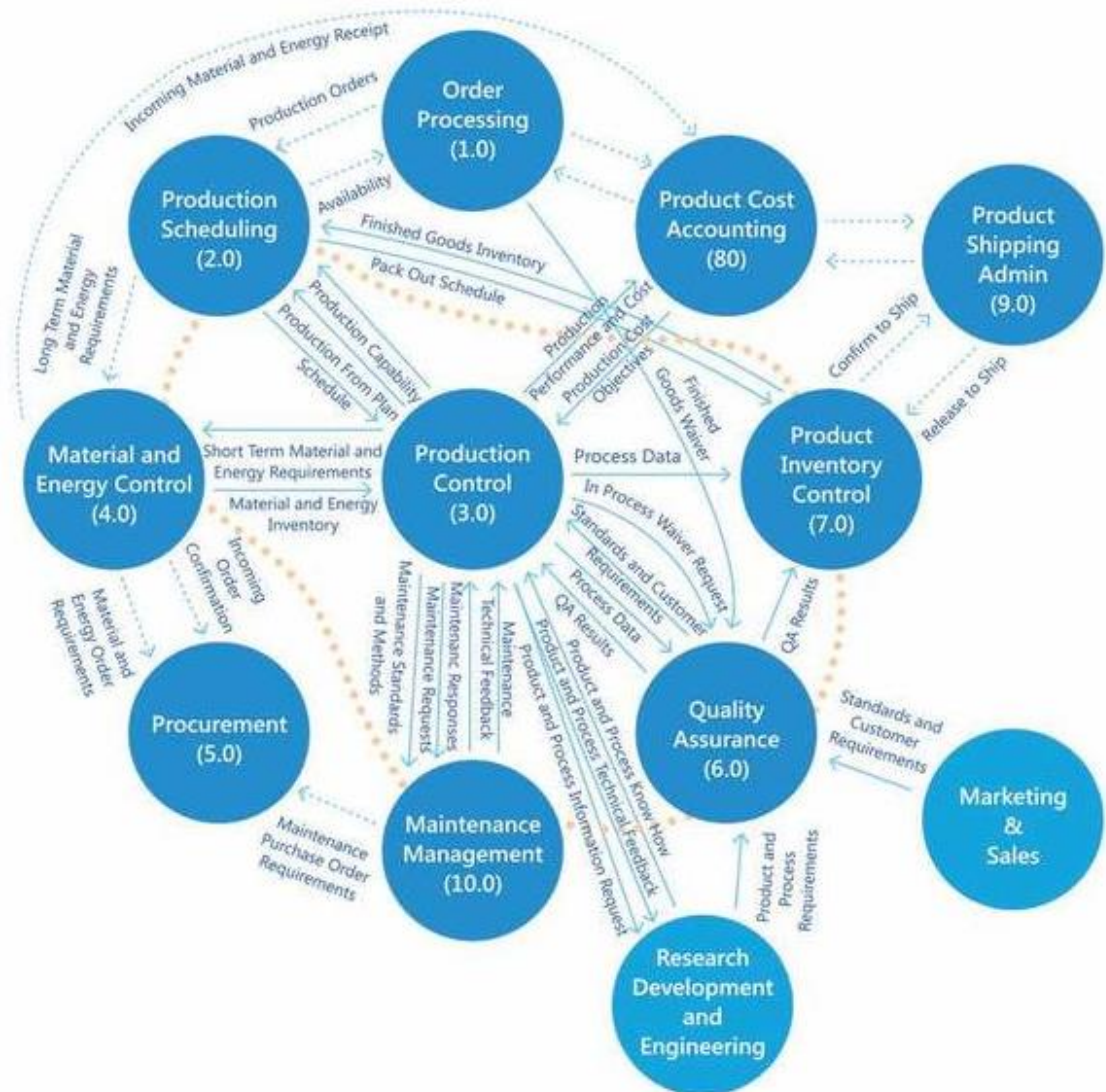


KUVA 27. ISA-95 -standardin määrittämät toiminnalliset hierarkiatasot (Almada-Lobo, 2015)

ISA-95 toiminnalliset hierarkia tasot ovat:

- Tasolla 4 on liiketoiminnan suunnittelu- ja logistiikkapäätökset, joita tuetaan toiminnanohjaus- ja toimitusketjun hallintajärjestelmillä (*engl. Supply Chain Management, SCM*).
- Tasolla 3 on valmistustoimintojen hallinta, jolla määritetään työnvirtauksen toiminta halutun lopputuotteen valmistamiseksi. Tuotannonohjausjärjestelmä mahdollistaa tuotteen valmistuksenohjauksen tarvitsemat toiminnallisuudet.
- Tasolla 2 on tuotantoprosessin automaattinen seuranta ja hallinta, joita ohjataan ohjelmoitavilla logiikoilla, hajautetuilla ohjausjärjestelmillä sekä valvomo-ohjelmistoilla.
- Tasolla 1 on anturointi ja toimilaitteet, joilla havaitaan ja käsitellään fyysistä tuotantoprosessia. Tasolla 0 on fyysinen tuotantoprosessi. Valmistustoimintojen hierarkian määrittämisen lisäksi ISA-95 tavoitteena on parantaa yhteentoimivuutta, erityisesti toiminnanohjaus- ja tuotannonohjausjärjestelmien välillä. (Almada-Lobo, 2015; Brandl 2008, 11.)

Kuvassa 28 on esitetty kompakti ISA-95 -standardin mukainen järjestelmäkuvaus sovellettuna Purdue-referenssimalliin. Katkoviiva kuvaa mahdollista rajaa toiminnanohjauksen ja ohjausjärjestelmän välillä. Tyypillisimmässä tapauksessa aikataulutetut tilaukset kulkevat ohjauskerrokseen, joka palauttaa tuotannon vasteen ja valmiustiedon. (Almada-Lobo, 2015)



KUVA 28. ISA-95 -standardi sovellettuna Purdue-referenssimalliin (Almada-Lobo, 2015)

ISA-95 -standardi kuvaa materiaalia, henkilöstöä, laitteita ja muuta vaadittavaa tietoa, jota vaaditaan tehokkaaseen aikataulutusmalliin. Standardi ei kuitenkaan määritä tarkasti, mitä järjestelmää tulisi käyttää missäkin tilanteessa. Se tarjoaa vain neutraaleja ohjeita ja tiedonvaihtorakenteita, joita yritykset muokkaavat omiin tarpeisiinsa. (Almada-Lobo, 2015)

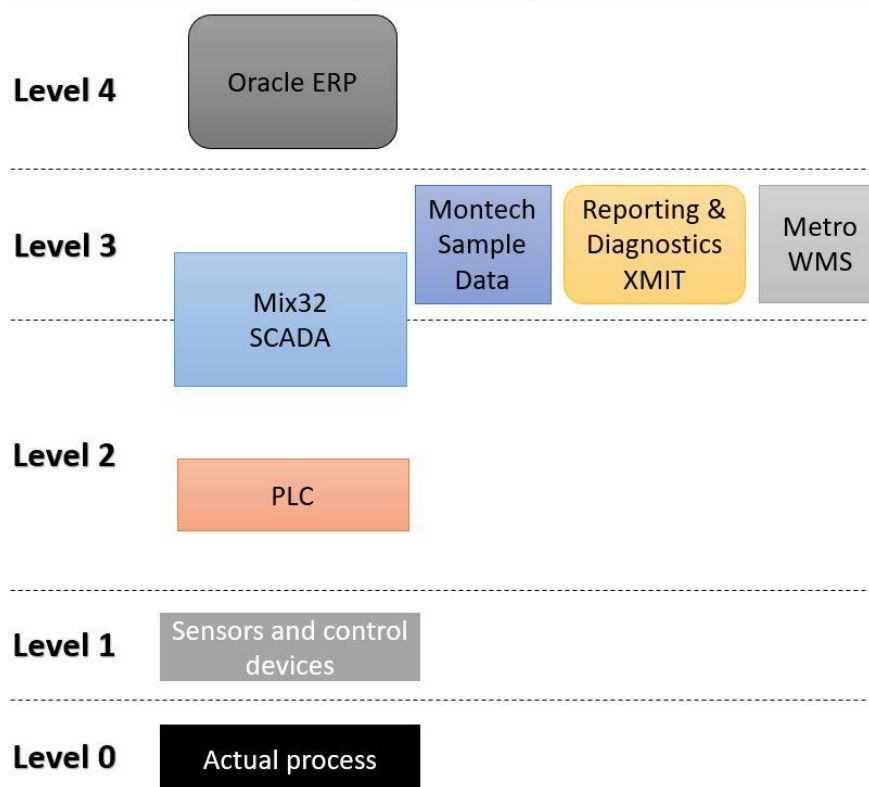
ISA-95 -standardi tarjoaa myös ohjeita valmistustoimintojen hallinnan toiminnallisen laajuuden määrittämiselle. Tämä sisältää resurssien ja toteutuksen hallinnan, yksityiskoh- taisen aikataulutuksen, tuotteiden toimittamisen sekä tiedonkeräämisen, -seurannan ja - analysoinnin, kun ne koskevat tuotanto-, huolto- ja varastotiloja. Perinteisesti tuotannon- ohjausjärjestelmä on rajapintana toiminnanohjaus- ja automaatiokerroksien välillä. Raja- pintana tuotannonohjausjärjestelmän on tarkoitus rekisteröidä ja hallita lähes reaaliajassa kaikki tuotantotilaukset, jotka tulevat toiminnanohjausjärjestelmästä. Reaaliajassa tapah- tuva hallinta toteutetaan niin, että tuotannonohjausjärjestelmä yhdistää valvontatasot ke- räämään reaaliaikaista tuotantotietoa sekä palauttamaan kaikki tarpeelliset logistiikka- ja taloustiedot takaisin toiminnanohjausjärjestelmälle. Tuotannonohjausjärjestelmän toteut- taminen perinteisellä tavalla ei hyödynnä tuotannonohjausjärjestelmän täyttä potentiaalia kehittää tuotannon tuottavuutta, laatua ja joustavuutta. (Almada-Lobo, 2015)

Optimaalisen tuottavuuden saavuttaminen vaatii yhdistelmän toimintoja, joita toiminnan- ohjaus- tai tuotannonohjausjärjestelmä toteuttavat. ISA-95 -standardi auttaa määrittä- mään toiminnot mallintamalla toiminnanohjaus- ja tuotannonohjausjärjestelmien välistä tiedonsiirtoa sekä mallintamalla tiedonsiirtoa valmistuksenohjausjärjestelmien välillä. ISA-95 -standardin implementointi vaatii modulaarisen tuotannonohjausjärjestelmän, jonka on pystyttävä suorittamaan määritettyjä moduuleja ilman, että järjestelmä on täysin implementoitu käyttöön. Moduulien on myös oltava helposti integroitavissa muiden jär- jestelmien moduulien kanssa. (Almada-Lobo, 2015)

4 JÄRJESTELMÄHIERARKIA

Kohdeyrityksen sekoitusosaston vanhoista järjestelmistä ei täysin muodostunut ISA-95 -standardin mukaista järjestelmähierarkiaa. Tämä johtui pääosin siitä, että järjestelmiä oli ajan kuluessa lisätty ja integroitu toisiinsa, eikä kokonaisuutta oltu toteutettu tiettyyn rakenteeseen. Yrityksellä ei myöskään ollut henkilöä, joka olisi tarkastellut järjestelmiä ISA-95 -standardin mukaisena kokonaisuutena. Kuvassa 29 on esitetty yrityksen aiemmat järjestelmät sijoitettuna ISA-95 mukaiseen hierarkiaan.

ISA-95 architecture



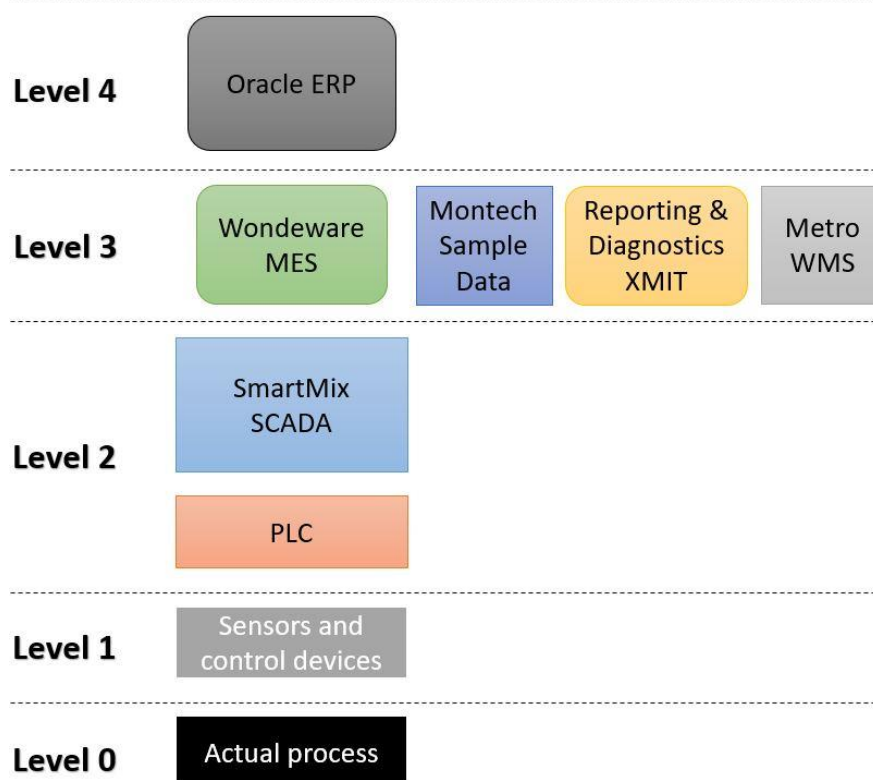
KUVA 29. Yrityksen vanha järjestelmähierarkia

Yrityksen ylimmän tason järjestelmä on Oraclen toiminnanohjausjärjestelmä. Yrityksellä ei ollut käytössä erillistä tuotannonohjausjärjestelmää, vaan tuotantolinjoja ohjaavaan Mix32 -valvomo-ohjelmistoon oli lisätty tuotannonohjauksen toiminnallisuuksia. Montech kerää valmistettavasta tuotteesta mittadataa ja lähettää sitä XMIT-järjestelmälle. XMIT tarjoaa diagnostiikka, analysointi ja raportointi toiminnallisuuksia kerätystä datasta. Metro -varastonhallintajärjestelmä (*engl. Warehouse Management System*) hallitsee

yrittäjien sisäistä logistiikkaa tuotantolinjojen ja varastojen välillä automaattisten vihi-vaunujen (*engl. Automated/Automatic Guided Vehicle*) avulla. Tasolla 2 on ohjelmoitavat logiikat, jotka ohjaavat kentällä olevia laitteita antureiden ja toimilaitteiden avulla.

Uusi järjestelmäarkkitehtuuri on ISA-95 -standardin mukainen. Kuvassa 30 on esitetty uudet järjestelmät sijoitettuna ISA-95 -standardin hierarkiaan. Vanhaan järjestelmärakenteeseen verrattuna uuteen järjestelmään on lisätty Wonderware -tuotannonohjausjärjestelmä sekä tuotantolinjoja ohjaava valvomo-ohjelmisto SmartMix.

ISA-95 architecture

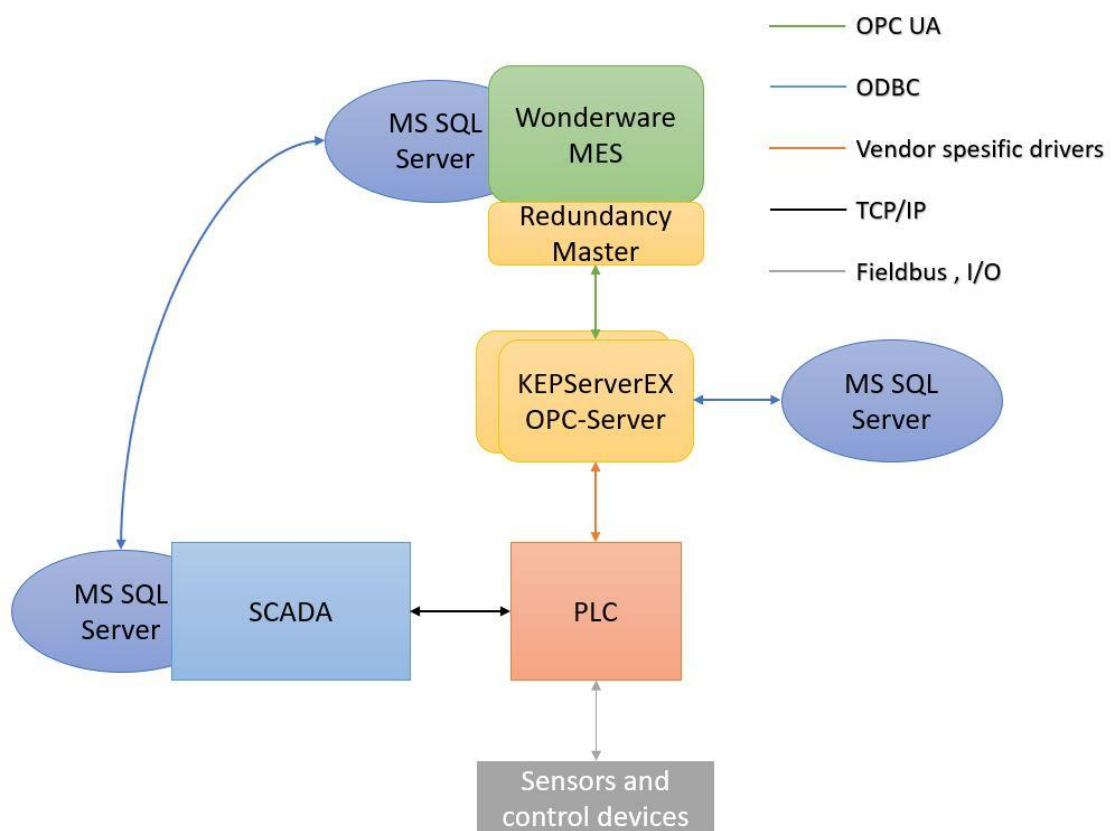


KUVA 30. Päivitetty järjestelmähierarkia

Mix32 -valvomo-ohjelmisto päätettiin korvata uudella järjestelmällä, koska sen kehitysmahdollisuudet olivat hyvin rajoittuneet järjestelmätoimittajan lopetettua ohjelmiston kehittämisen. Mix32-järjestelmän integrointia uuteen tuotannonohjausjärjestelmään ei pidetty järkevänä. SmartMix perustuu Siemensin Simatic WinCC -valvomo-ohjelmistoon. SmartMix -valvomo-ohjelmistoa on mahdollista kehittää yrityksen omilla resursseilla ja se on integroitavissa tuotannonohjausjärjestelmään.

4.1 Järjestelmien välinen kommunikointi

Tuotannonohjausjärjestelmän ja ohjelmoitavien logiikoiden väliseksi OPC-palvelimeksi valittiin Kepwaren KepServerEX -tuote. Valintaan vaikutti aiemmat positiiviset kokemukset tuotteesta. OPC-palvelin kahdennettiin ja tuotannonohjausjärjestelmän palvelimelle asennettiin Kepwaren RedundancyMaster, joka vaihtaa OPC-palvelinta, mikäli yhteys ensisijaiseen palvelimeen katkeaa. Kuvassa 31 on esitetty tuotannonohjausjärjestelmän, ohjelmoitavien logiikoiden ja tietokantapalvelimien väliset kommunikointiprotokollat.



KUVA 31. Yrityksen järjestelmien väliset kommunikointitavat

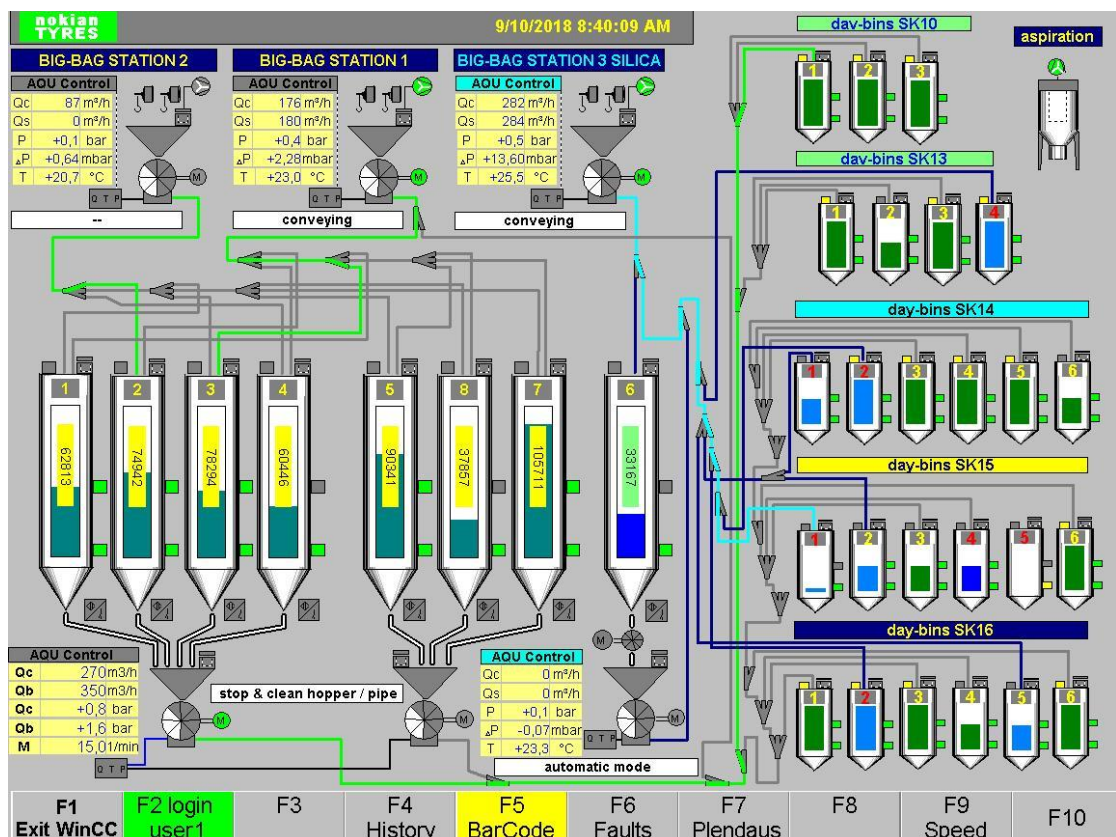
Tuotannonohjausjärjestelmän ja OPC-palvelimen välinen kommunikointi toteutettiin OPC UA -kommunikoinnilla. Syynä oli sen helppo käytettävyys sekä sisäänrakennettu tietoturva. Eri tietokantojen sekä OPC-palvelimen ja tietokantojen välinen kommunikointi toteutettiin ODBC-ajureiden avulla. ODBC on standardoitu avoin tietokannoille tarkoitettu rajapinta. Tiedonsiirto OPC-palvelimen ja ohjelmoitavien logiikoiden välillä tapahtuu valmistajakohtaisilla ajureilla. Valvomo-ohjelmistojen ja ohjelmoitavien logiikoiden välillä tiedonsiirto toteutettiin TCP/IP protokollalla. Ohjelmoitava logiikka kommunikoi kenttälaitteiden kanssa kenttäväylän ja I/O tiedon välityksellä.

5 PNEUMAATTISEN SIIRTOJÄRJESTELMÄN INTEGRAATIO

Tuotannonohjausjärjestelmän hankinnan myötä osaa tehtaan pneumaattisen siirtojärjestelmän toiminnallisuuksista haluttiin hallita tuotannonohjausjärjestelmällä. Vanha Access-tietokanta oli korruptoitunut, eikä järjestelmän siilojen täyttölupia enää pystytty hallinnoimaan vanhalla järjestelmällä. Tuotannonohjausjärjestelmään siirrettiäisiin siilojen lupataulu, päiväsiilojen täyttökielto -toiminto ja siilojen rajatiedot.

5.1 Pneumaattisen siirtojärjestelmän yleiskuvaus

Pneumaattisen siirtojärjestelmän tehtävänä on siirtää nokea ja silikaa tuotantolinjoille. Nokea ja silikaa käytetään materiaaleina kumisekoituksissa. Noki ja silika varastoidaan ulkosiiloihin, jotka ovat sijoitettuna tehtaan ulkopuolelle sekä päiväsiiloihin, jotka ovat sijoitettuna tehtaan sisäpuolelle. Kuvassa 32 on esitetty pneumaattisen valvomo-ohjelmiston pääsivu, josta selviää järjestelmän rakenne.



KUVA 32. Pneumaattisen siirtojärjestelmän valvomo-ohjelmisto

Kuvan oikeassa reunassa ovat esitetty sekoituskonelinjat sekä niiden päiväsiilot. Kuvan vasemmassa reunassa ovat esitetty ulkosiilot. Ulkosiiloihin materiaalia täytetään säkeistä, niille tarkoitetuilta syöttöasemilta, jotka ovat esitetty kuvan yläreunassa (big-bag station).

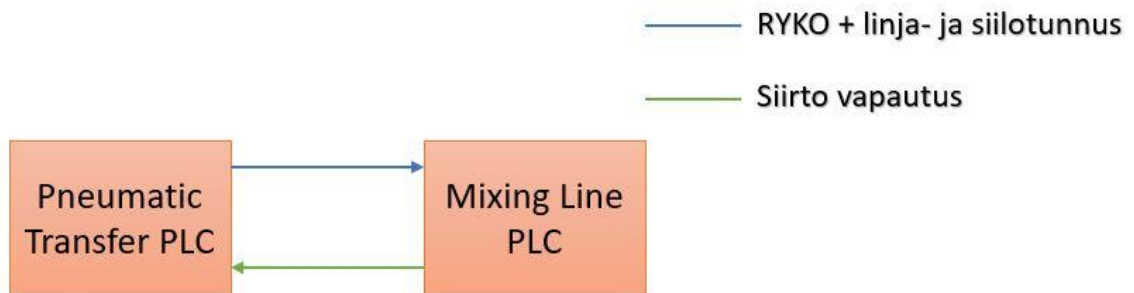
Siiloihin siirrettävien materiaalien tarkistus perustuu pneumaattisessa siirtojärjestelmässä olevaan lupatauluun. Kun operaattori aloittaa materiaalien täytön syöttöasemilta ulko- tai päiväsiiloihin, täytyy hänen lukea materiaalin QR-koodi, joka sisältää materiaalin ryhmäkooditiedon. Järjestelmässä materiaalit ovat eritelty toisistaan ryhmäkoodin eli RYKO-koodin avulla. Pneumaattinen siirtojärjestelmä vertailee RYKO-tiedon ohjelmoitavassa logiikassa olevaan lupatauluun. Jos materiaali on sallittu syötettäväksi haluttuun siiloon, saa säkkinostin luvan siirtää materiaalin syöttöasemaan (kuva 33).



KUVA 33. Noen ja silikan syöttöasema

Kun pneumaattinen siirtojärjestelmä siirtää materiaalia ulkosiiloista päiväsiiloihin, suorittaa pneumaattisen siirtojärjestelmä ja sekoituskonelinja RYKO-vertailun ennen siirron aloittamista. Eräitä päiväsiiloja on mahdollista täyttää myös suoraan syöttöasemilta.

Jokaisella perusekoituslinjalla on omat päiväsiilot, joista materiaalia syötetään tuotettiin sekoituksiin. Kuvassa 34 on esitetty pneumaattisen siirtojärjestelmän ja sekoituskonelinjan välinen kommunikointi.



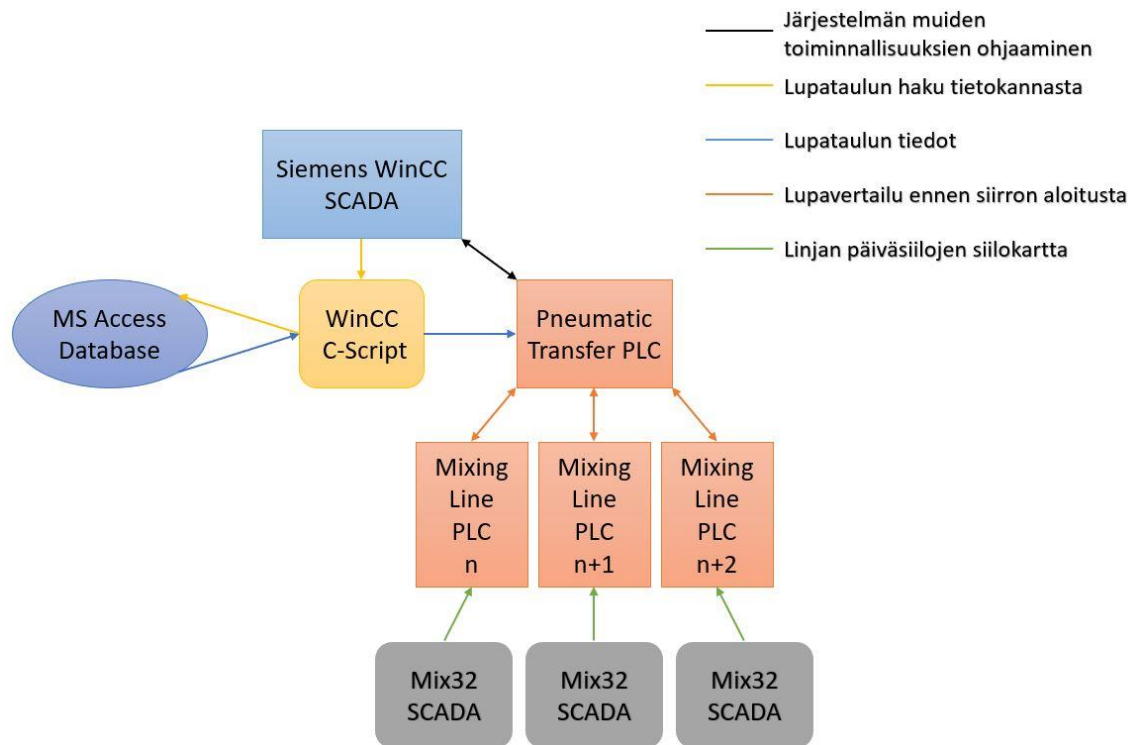
KUVA 34. Pneumaattisen siirtojärjestelmän ja linjalogiikan välinen kommunikointi

Kun pneumaattinen siirtojärjestelmä havaitsee, että materiaalia on alle päiväsiilon pyyntirajan, se aloittaa siirtosekvenssin. Ennen siirron aloitusta pneumaattinen siirtojärjestelmä lähettää siirrettävän materiaalin RYKO-koodin sekoituskonelinjalle. Tämän lisäksi sekoituskonelinjalle lähetetään linja- ja siilotunnus, jotta sekoituskonelinjan ohjelmoitava logiikka pystyy vertailemaan RYKO-koodin oikeaan siiloon siilokartassaan. Siilokartta sisältää linjan kaikki päiväsiilot ja tiedon, mitä materiaalia kuhunkin siiloon saa täyttää. Tarvittaessa käyttäjä voi muokata siilokartan sisältöä valvomo-ohjelmiston avulla.

Pneumaattinen siirtojärjestelmä tekee RYKO-vertailun ennen materiaalsiirron aloittamista. Jos vertailun tulos on tosi (*engl. true*), sekoituskonelinjan ohjelmoitava logiikka lähettää pneumaattisen siirtojärjestelmän logiikalle luvan aloittaa noen tai silikan siirto linjan päiväsiiloon. Pneumaattinen siirtojärjestelmä siirtää materiaalia päiväsiiloon, kunnes siilon yläraja saavutetaan. Tämän jälkeen pneumaattinen siirtojärjestelmä siirtyy suorittamaan seuraavaa tehtävää ja RYKO-tarkastus aloitetaan alusta.

5.2 Vanhan järjestelmän rakenne

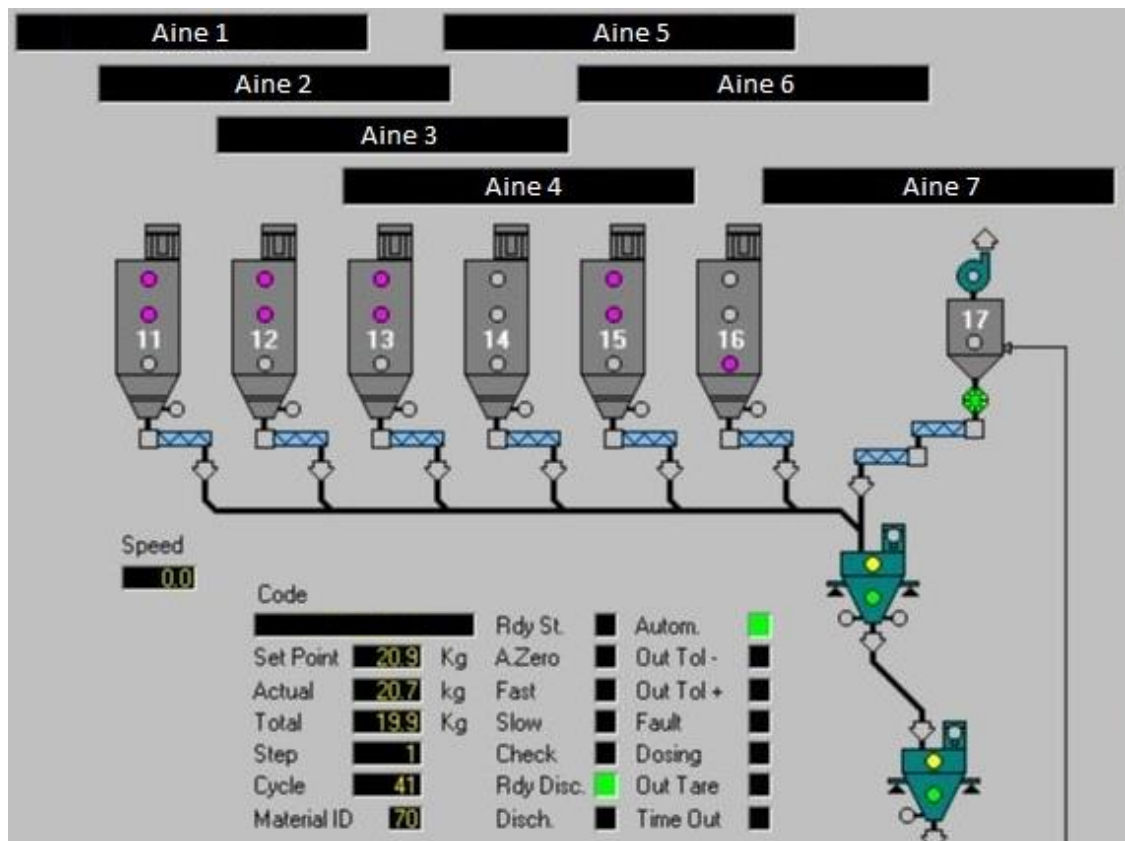
Jotta pneumaattinen siirtojärjestelmä voitaisiin integroida tuotannonohjausjärjestelmää, täytyi selvittää alkuperäisen järjestelmän toiminta. Kuvassa 35 on esitetty vanhan pneumaattisen siirtojärjestelmän rakenne.



KUVA 35. Vanha pneumaattisen siirtojärjestelmän rakenne

Pneumaattisen siirtojärjestelmän valvomo-ohjelmisto suorittaa C-kielellä toteutettua skriptiä tietyllä syklillä tai kun käyttäjä aktivoi sen. Skripti hakee Access-tietokannasta siirrettävien materiaalien lupataulun. Lupataulu sisältää siirrettävien materiaalien RYKO-tiedon ja RYKO-tiedolle sallitut siilot. Ainoastaan sallittuihin siiloihin saa siirtää kyseisellä RYKO-tiedolla olevaa materiaalia. Skripti siirtää lupataulun pneumaattisen siirtojärjestelmän ohjelmoitavaan logiikkaan.

Sekoituskonelinjan Mix32 -valvomo-ohjelmisto mahdollistaa linjan päiväsiilojen siilokartan hallinnoimisen. Valvomo-ohjelmiston avulla käyttäjä voi vaihtaa siiloon siirrettävää materiaalia muokkaamalla siilokarttaa. Kuvassa 36 on esitetty valvomo-ohjelmiston näkymä siilokartan hallinnoimiseen.

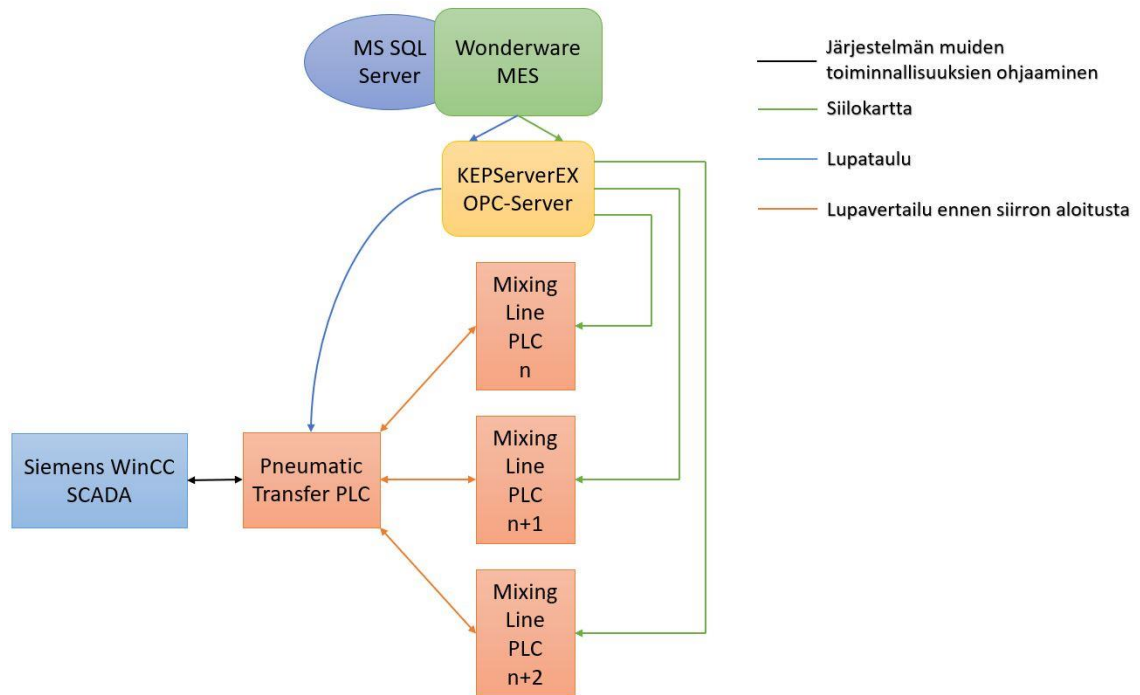


KUVA 36. Mix32 -valvomo-ohjelmiston siilokartta

Käyttäjän tulee tehdä samat muutokset pneumaattisen järjestelmän lupatauluun, jotta RYKO-vertailun tulos olisi tosi. Mikäli muutokset tehdään vain toiseen järjestelmään, siilontäytönvarmistus (RYKO-vertailu) estää siirron aloittamisen.

5.3 Uuden järjestelmän rakenne

Uudessa järjestelmässä sekoituskonelinjan valvomo-ohjelmisto ei enää hallinnoi linjalogiikan siilokarttaa. Uusi tuotannonohjausjärjestelmä lähettää päiväsiiloen siilokartan tiedot suoraan sekoituskonelinjojen ohjelmoitaville logiikoille. Pneumaattisen siirtojärjestelmän ohjelmoitavalle logiikalle lähetetään kaikkien siirrettävien materiaalien RYKO-tiedot ja siiloluvat. Kuvassa 37 on esitetty uuden järjestelmän rakenne.



KUVA 37. Uuden pneumaattisen siirtojärjestelmän rakenne

Pneumaattisen siirtojärjestelmän siirtosekvenssi ei poikkea vanhan järjestelmän siirtosekvenssistä. Lupavertailu ja materiaalin siirtäminen suoritetaan, mikäli pneumaattisen siirtojärjestelmän ja sekoituskonelinjan välinen RYKO-vertailu on tosi.

6 INTEGRAATION TOTEUTUS

Osaa kohdeyrityksen pneumaattisen siirtojärjestelmän toiminnallisuuksista haluttiin jatkossa hallita tuotannonohjausjärjestelmällä. Tämän toteuttaminen vaati pneumaattinen siirtojärjestelmän integroimista osaksi tuotannonohjausjärjestelmää. Järjestelmän toteutus koostui useasta eri vaiheesta. Vaiheet on esitetty kuvassa 38.



KUVA 38. Pneumaattisen siirtojärjestelmän integrointityön toteutuksen eri vaiheet

6.1 OPC-palvelin ja -tagit

Toteutus aloitettiin asentamalla OPC-palvelimet, joilla siirrettäisiin tietoa ohjelmoitavien logiikoiden ja tuotannonohjausjärjestelmän välillä. OPC-palvelimia asennettiin kaksi, jotta tiedonsiirto pystyttäisiin varmistamaan tilanteessa, jossa tiedonsiirtoyhteys ensisijaiseen palvelimeen katkeaisi. Molemmat OPC-palvelimet konfiguroitiin asetuksiltaan samanlaisiksi, eivätkä ne ole toisistaan riippuvaisia. Tuotannonohjausjärjestelmän palvelimelle asennettiin Kepwaren RedundancyMaster -tuote, joka tarkkailee ensisijaista tiedonsiirtoyhteyttä. Mikäli tiedonsiirtoyhteys katkeaa, vaihtaa RedundancyMaster tiedonsiirtoyhteyden automaattisesti toissijaiseen tiedonsiirtoyhteyteen.

Pneumaattisen siirtojärjestelmän sekä sekoituskonelinjojen ohjelmoitaviin logiikoihin luotiin Data Blockit, joihin OPC-palvelimen tagit (item) linkitettäisiin. Pneumaattisen siirtojärjestelmän Data Block 250 sisältää siilojen lupataulun eli materiaalien RYKO-tiedot sekä luvat, mihin siiloon mitään materiaalia saa siirtää (kuva 39). Data Blockiin määritettiin myös integer-tyyppinen muuttuja siilojen lupataulun alustusta varten.

Address	Name	Type	Initial value	Comment
0.0		STRUCT		
+0.0	RYKO_Mat_1	STRING[67]	''	RYKO+Silo permissions
+70.0	RYKO_Mat_2	STRING[67]	''	RYKO+Silo permissions
+140.0	RYKO_Mat_3	STRING[67]	''	RYKO+Silo permissions
+210.0	RYKO_Mat_4	STRING[67]	''	RYKO+Silo permissions
+280.0	RYKO_Mat_5	STRING[67]	''	RYKO+Silo permissions
+350.0	RYKO_Mat_6	STRING[67]	''	RYKO+Silo permissions
+420.0	RYKO_Mat_7	STRING[67]	''	RYKO+Silo permissions
+490.0	RYKO_Mat_8	STRING[67]	''	RYKO+Silo permissions
+560.0	RYKO_Mat_9	STRING[67]	''	RYKO+Silo permissions
+630.0	RYKO_Mat_10	STRING[67]	''	RYKO+Silo permissions
+700.0	RYKO_Mat_11	STRING[67]	''	RYKO+Silo permissions
+770.0	RYKO_Mat_12	STRING[67]	''	RYKO+Silo permissions
+840.0	RYKO_Mat_13	STRING[67]	''	RYKO+Silo permissions
+910.0	RYKO_Mat_14	STRING[67]	''	RYKO+Silo permissions
+980.0	RYKO_Mat_15	STRING[67]	''	RYKO+Silo permissions
+1050.0	RYKO_Mat_16	STRING[67]	''	RYKO+Silo permissions
+1120.0	RYKO_Mat_17	STRING[67]	''	RYKO+Silo permissions
+1190.0	RYKO_Mat_18	STRING[67]	''	RYKO+Silo permissions
+1260.0	RYKO_Mat_19	STRING[67]	''	RYKO+Silo permissions
+1330.0	RYKO_Mat_20	STRING[67]	''	RYKO+Silo permissions
+1400.0	RYKO_Table_Init	INT	0	Integer for init. sequence
-1402.0		END_STRUCT		

KUVA 39. Pneumaattisen siirtojärjestelmän Data Block (DB 250)

Siilojen rajatietoantureita sekä siilojen täyttökieltoa varten luotiin oma Data Block (500) (kuva 40). DB500 tulisi sisältämään kaikkien siilojen anturien ja täyttökieltojen datat. Siilon täyttökielto on tuotannonohjausjärjestelmän kirjoittamaa dataa. Siilojen antureista kerättävä materiaalin rajatieto on pneumaattisen siirtojärjestelmän kirjoittamaa dataa.

Address	Name	Type	Initial value	Comment
0.0		STRUCT		
+0.0	Silo_Line	ARRAY[0..9]		
+0.0		STRUCT		
+0.0	Silo_1_High	BOOL	FALSE	
+0.1	Silo_1_Med	BOOL	FALSE	
+0.2	Silo_1_Low	BOOL	FALSE	
+0.3	Silo_1_Usage_Denial	BOOL	FALSE	
+0.4	Spare_Bool_1	BOOL	FALSE	
+0.5	Spare_Bool_2	BOOL	FALSE	
+0.6	Spare_Bool_3	BOOL	FALSE	
+0.7	Spare_Bool_4	BOOL	FALSE	
+1.0	Silo_2_High	BOOL	FALSE	
+1.1	Silo_2_Med	BOOL	FALSE	
+1.2	Silo_2_Low	BOOL	FALSE	
+1.3	Silo_2_Usage_Denial	BOOL	FALSE	
+1.4	Spare_Bool_5	BOOL	FALSE	
+1.5	Spare_Bool_6	BOOL	FALSE	
+1.6	Spare_Bool_7	BOOL	FALSE	
+1.7	Spare_Bool_8	BOOL	FALSE	
+2.0	Silo_3_High	BOOL	FALSE	
+2.1	Silo_3_Med	BOOL	FALSE	
+2.2	Silo_3_Low	BOOL	FALSE	
+2.3	Silo_3_Usage_Denial	BOOL	FALSE	

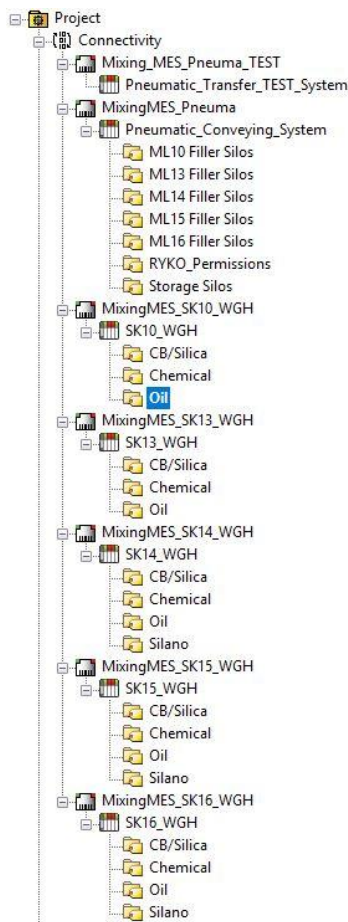
KUVA 40. Pneumaattisen siirtojärjestelmän Data Block (DB500)

Sekoituskonelinjojen Data Blockkeihin (DB501) siirretään kunkin siilon materiaalin RYKO-tieto (kuva 41).

Address	Name	Type	Initial value	Comment
0.0		STRUCT		
+0.0	Filler_Silo1_RYKO	STRING[7]	''	Filler Silo 1 material RYKO
+10.0	Filler_Silo2_RYKO	STRING[7]	''	Filler Silo 2 material RYKO
+20.0	Filler_Silo3_RYKO	STRING[7]	''	Filler Silo 3 material RYKO
+30.0	Filler_Silo4_RYKO	STRING[7]	''	Filler Silo 4 material RYKO
+40.0	Filler_Silo5_RYKO	STRING[7]	''	Filler Silo 5 material RYKO
+50.0	Filler_Silo6_RYKO	STRING[7]	''	Filler Silo 6 material RYKO

KUVA 41. Sekoituskonelinjan Data Block (DB501)

Data Blockit suunniteltiin käytettäväksi kaikilla linjoilla samanlaisina, jolloin ohjelman hallittavuus olisi parempi. Kun tarvittavat Data Blockit oli ladattu ohjelmoitaviin logiikoihin, luotiin OPC-palvelimelle tagit. Tagit linkitettiin Data Block-osoitteisiin. Tagit myös ryhmiteltiin linjakohtaisesti, jotta niiden hallittavuus helpottuisi (kuva 42).



KUVA 42. OPC-palvelimelle luodut tag-ryhmät

Sekoituskonelinjan tageihin tuotannonohjausjärjestelmä lähettää päiväsiiloihin sallittavan materiaalin RYKO-tiedon. Jokaiselle päiväsiilolle luotiin RYKO-tietoa varten tarvittavat tagit. Jokaiselle tagille määritettiin, millä taajuudella tagia päivitetäisiin. Päivitystaajuudeksi määritettiin 1500ms, koska tietoja muutetaan harvoin eikä tieto ole aikakriittistä. Kuvassa 43 on esitetty yhden sekoituskonelinjan päiväsiiloille luodut OPC-tagit.

Tag Name	Address	Data Type	Scan Rate	Scaling	Description
ML16_FS1_Allowed_RYKO	DB501.STRING0[7]	String	1500	None	ML16 Daybin 1601 material RYKO
ML16_FS2_Allowed_RYKO	DB501.STRING10[7]	String	1500	None	ML16 Daybin 1602 material RYKO
ML16_FS3_Allowed_RYKO	DB501.STRING20[7]	String	1500	None	ML16 Daybin 1603 material RYKO
ML16_FS4_Allowed_RYKO	DB501.STRING30[7]	String	1500	None	ML16 Daybin 1604 material RYKO
ML16_FS5_Allowed_RYKO	DB501.STRING40[7]	String	1500	None	ML16 Daybin 1605 material RYKO
ML16_FS6_Allowed_RYKO	DB501.STRING50[7]	String	1500	None	ML16 Daybin 1606 material RYKO

KUVA 43. Sekoituskonelinjan päiväsiilojen OPC-tagit

Pneumaattiselle siirtojärjestelmälle tuotannonohjausjärjestelmä lähettää materiaalien lupataulun. Kuvassa 44 on esitetty pneumaattisen siirtojärjestelmän siilojen lupataululle luodut tagit.

Tag Name	Address	Data Type	Scan Rate	Scaling	Description
RVKO_Table_Init	DB250.DBW1400	Word	100	None	Integer for silo permission init.
RVKO_Mat1	DB250.STRING60[67]	String	1500	None	RVKO + Silo permissions to that material
RVKO_Mat2	DB250.STRING70[67]	String	1500	None	RVKO + Silo permissions to that material
RVKO_Mat3	DB250.STRING140[67]	String	1500	None	RVKO + Silo permissions to that material
RVKO_Mat4	DB250.STRING210[67]	String	1500	None	RVKO + Silo permissions to that material
RVKO_Mat5	DB250.STRING280[67]	String	1500	None	RVKO + Silo permissions to that material
RVKO_Mat6	DB250.STRING350[67]	String	1500	None	RVKO + Silo permissions to that material
RVKO_Mat7	DB250.STRING420[67]	String	1500	None	RVKO + Silo permissions to that material
RVKO_Mat8	DB250.STRING490[67]	String	1500	None	RVKO + Silo permissions to that material
RVKO_Mat9	DB250.STRING560[67]	String	1500	None	RVKO + Silo permissions to that material
RVKO_Mat10	DB250.STRING630[67]	String	1500	None	RVKO + Silo permissions to that material
RVKO_Mat11	DB250.STRING700[67]	String	1500	None	RVKO + Silo permissions to that material
RVKO_Mat12	DB250.STRING770[67]	String	1500	None	RVKO + Silo permissions to that material
RVKO_Mat13	DB250.STRING840[67]	String	1500	None	RVKO + Silo permissions to that material
RVKO_Mat14	DB250.STRING910[67]	String	1500	None	RVKO + Silo permissions to that material
RVKO_Mat15	DB250.STRING980[67]	String	1500	None	RVKO + Silo permissions to that material

KUVA 44. Siilojen lupataulun siirtoon luodut tagit

Pneumaattisen siirtojärjestelmän ohjelmoitava logiikka välittää tuotannonohjausjärjestelmälle tietoa siilojen materiaalien tasosta. Siilojen korkea-, keski- ja alatasoantureiden tietoa välitetään tuotannonohjausjärjestelmään. Tuotannonohjausjärjestelmän siilojen täyttökielto -toiminnallisuutta varten luotiin jokaiselle siilolle omat tagit (kuva 45).

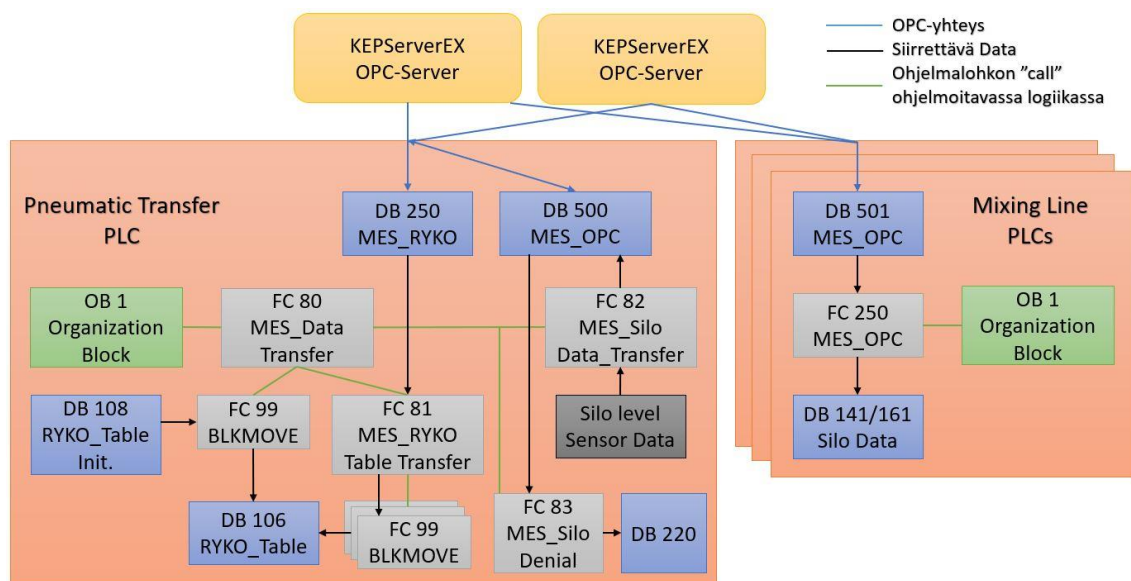
Tag Name	Address	Data Type	Scan Rate	Scaling	Description
ML10_Filler_Silo1_Level_High	DB500.B220.0	Boolean	1500	None	ML10 Daybin 1003 material at high level
ML10_Filler_Silo1_Level_Medium	DB500.B220.1	Boolean	1500	None	ML10 Daybin 1003 material at medium level
ML10_Filler_Silo1_Level_Low	DB500.B220.2	Boolean	1500	None	ML10 Daybin 1003 material at low level
ML10_Filler_Silo1_Usage_Denied	DB500.B220.3	Boolean	1500	None	ML10 Daybin 1003 usage denied
ML10_Filler_Silo2_Level_High	DB500.B221.0	Boolean	1500	None	ML10 Daybin 1002 material at high level
ML10_Filler_Silo2_Level_Medium	DB500.B221.1	Boolean	1500	None	ML10 Daybin 1002 material at medium level
ML10_Filler_Silo2_Level_Low	DB500.B221.2	Boolean	1500	None	ML10 Daybin 1002 material at low level
ML10_Filler_Silo2_Usage_Denied	DB500.B221.3	Boolean	1500	None	ML10 Daybin 1002 usage denied
ML10_Filler_Silo3_Level_High	DB500.B222.0	Boolean	1500	None	ML10 Daybin 1001 material at high level
ML10_Filler_Silo3_Level_Medium	DB500.B222.1	Boolean	1500	None	ML10 Daybin 1001 material at medium level
ML10_Filler_Silo3_Level_Low	DB500.B222.2	Boolean	1500	None	ML10 Daybin 1001 material at low level
ML10_Filler_Silo3_Usage_Denied	DB500.B222.3	Boolean	1500	None	ML10 Daybin 1001 usage denied
ML10_Filler_Silo4_Level_High	DB500.B223.0	Boolean	1500	None	ML10 Daybin 1006 material at high level
ML10_Filler_Silo4_Level_Medium	DB500.B223.1	Boolean	1500	None	ML10 Daybin 1006 material at medium level
ML10_Filler_Silo4_Level_Low	DB500.B223.2	Boolean	1500	None	ML10 Daybin 1006 material at low level
ML10_Filler_Silo4_Usage_Denied	DB500.B223.3	Boolean	1500	None	ML10 Daybin 1006 usage denied

KUVA 45. Siilojen materiaalien rajatiedot

Kun kaikki tarvittavat tagit oli luotu, avattiin tuotannonohjausjärjestelmän ja OPC-palvelimen välinen OPC UA -yhteys. Tarkoituksena oli tehdä alustavaa yhteyksien testausta tuotannonohjaustestiympäristön, OPC-palvelimen ja ohjelmoitavien logiikoiden välillä.

6.2 PLC-ohjelma

PLC-ohjelman kehittämistä varten käytettiin testilogiikkaa. Testilogiikalla oli tarkoitus testata pneumaattisen siirtojärjestelmän PLC-ohjelmaa. Testilogiikka yhdistettiin tuotannon lähiverkkoon ja testilogiikalle avattiin OPC-yhteys. PLC-ohjelmat kehitettiin pneumaattisen siirtojärjestelmän ohjelmoitavalle logiikalle sekä jokaiselle sekoituskonelinjan ohjelmoitavalle logiikalle. Sekoituskonelinjojen PLC-ohjelma suunniteltiin niin, että se olisi käytettävissä jokaisella tuotantolinjalla. Kuvassa 46 on esitetty PLC-ohjelmien rakenne.



KUVA 46. PLC-ohjelmien rakenne

Kaikille tuotannonohjausjärjestelmään kehitettäville toiminnallisuuksille luotiin omat PLC-ohjelmat. PLC-ohjelmia olivat:

- Siilojen lupataulun hallinta (Pneumaattinen siirtojärjestelmä FC80, FC81 ja sekoituskonelinjat FC250).
- Siilojen anturitietojen siirtäminen tuotannonohjausjärjestelmään (FC82).
- Päiväsiilojen täyttökielto (FC83).

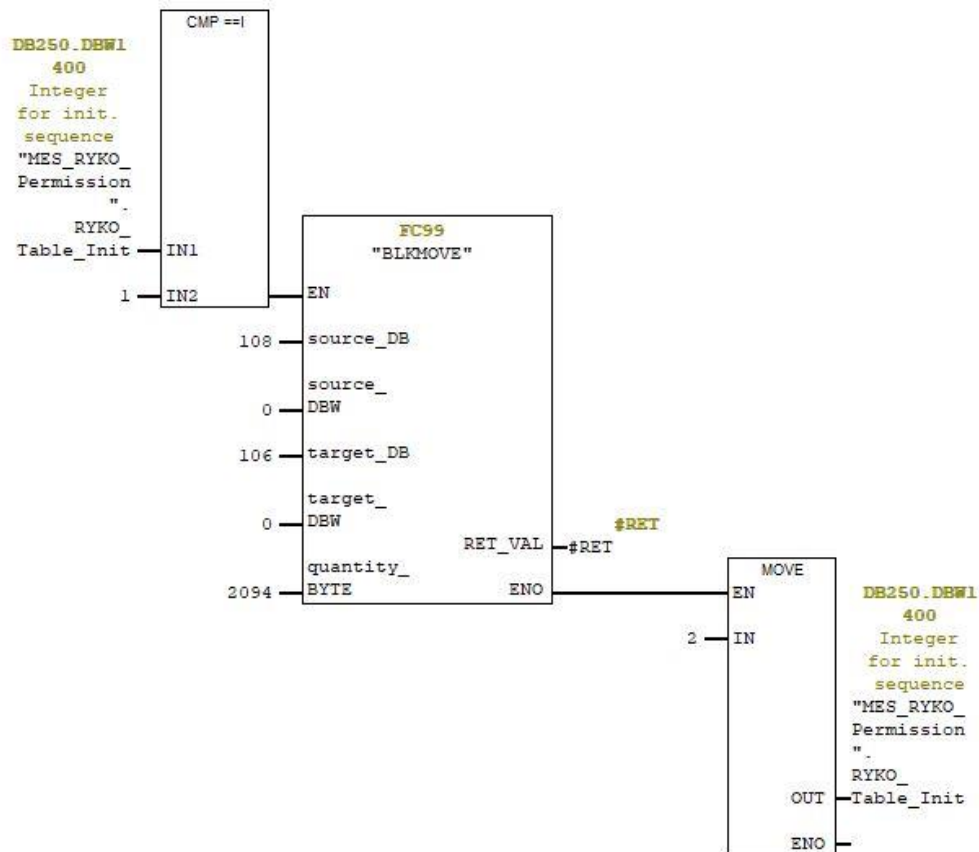
Organization Block 1:ssä kutsutaan (*engl. call*) ohjelmia FC80 & FC 250. OB1 on Siemensin organisaatiolohko, jossa kutsutaan kaikkia ohjelmia sekä aliohjelmia, mikäli halutaan, että ne suoritetaan jokaisella ohjelmakierrolla. Funktiossa FC80 kutsuttiin muita aliohjelmia (FC81, FC82 FC83) sekä toteutettiin siilojen lupataulun alustus (FC99 ja DB108).

6.2.1 Siilojen lupataulu ja siilokartta

Jotta tuotannonohjausjärjestelmä voisi hallita siilojen täyttölupia, tuli pneumaattiseen siirtojärjestelmään siirtää siilojen lupataulu. Tämän lisäksi siilokartta tuli siirtää jokaisen sekoituskonelinjan ohjelmoitavaan logiikkaan. Siilojen lupataulun siirtoa varten OPC-palvelimille oli luotu RYKO_Mat -tagit. RYKO_Mat -tagit on linkitetty DB 250 Data Blockiin, jonne siilojen lupataulu on tallennettu. Jotta pneumaattinen siirtojärjestelmä voisi hyödyntää tuotannonohjausjärjestelmän lähettämää lupataulua, tuli lupataulu vielä siirtää DB 250 Data Blockista DB 106 Data Blockiin.

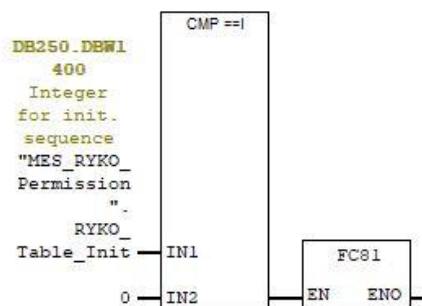
Käyttäjän tehdessä muutoksia tuotannonohjausjärjestelmän käyttöliittymästä siilojen sisältämiin materiaaleihin, tuotannonohjausjärjestelmä kirjoittaa OPC-palvelimelle RYKO_Table_Init-tagiin arvon 1. Tällöin pneumaattisen siirtojärjestelmä ohjelmoitava logiikka alustaa Data Blockin 106 siilon lupataulun siirtoa varten. Tämän jälkeen pneumaattisen siirtojärjestelmän ohjelmoitava logiikka asettaa RYKO_Table_Init-tagiin linkitettyyn osoitteeseen arvon 2, jotta tuotannonohjausjärjestelmä voi aloittaa uuden lupataulun siirtämisen (kuva 47). Siilon lupataulun alustus suoritetaan aina ennen uuden lupataulun lähettämistä. Näin voidaan olla varmoja, ettei käytettävään Data Blockiin jää vanhoja siilon täyttölupia käyttöön. Lupataulu siirretään aina kokonaan, vaikka käyttäjä tekisi muutoksia vain yhteen siiloon tuotannonohjausjärjestelmän käyttöliittymästä.

□ Network 1 : Initialize DB106 before transferring RYKO table



□ Network 2 : Transfer RYKO table from DB250 to DB 106

Transfer RYKO table from DB250 to DB 106 when MES has done changes to OPC



KUVA 47. Siilojen lupataulun siirtosekvenssi

Kun tuotannonohjausjärjestelmä on siirtänyt uuden lupataulun RYKO_Mat -tageihin, se kirjoittaa RYKO_Table_Init-tageihin arvon 0. Tämän jälkeen pneumaattisen siirtojärjestelmän ohjelmoitava logiikka suorittaa FC81-funktion. FC81-funktio siirtää RYKO_Mat -tageihin linkitetyn DB250 Data Blockin arvot DB106 Data Blockiin pneumaattisen siirtojärjestelmän käytettäväksi (kuva 47-48).

FC81 : RYKO-data + Silo permission

```
Transfer 67 char string from MES OPC DB250 to DB106
First 7chars are RYKO, after that all silo permissions
```

Network 1: Material 1. RYKO data transfer

L	"MES_RYKO_Permission".RYKO_Mat_1[1]	DB250.DBB2	-- RYKO+Silo permissions
T	"Bar_Code list from PC".material_01.material_01	DB106.DBB4	-- all value in 'char'
L	"MES_RYKO_Permission".RYKO_Mat_1[2]	DB250.DBB3	-- RYKO+Silo permissions
T	"Bar_Code list from PC".material_01.material_02	DB106.DBB5	
L	"MES_RYKO_Permission".RYKO_Mat_1[3]	DB250.DBB4	-- RYKO+Silo permissions
T	"Bar_Code list from PC".material_01.material_03	DB106.DBB6	
L	"MES_RYKO_Permission".RYKO_Mat_1[4]	DB250.DBB5	-- RYKO+Silo permissions
T	"Bar_Code list from PC".material_01.material_04	DB106.DBB7	
L	"MES_RYKO_Permission".RYKO_Mat_1[5]	DB250.DBB6	-- RYKO+Silo permissions
T	"Bar_Code list from PC".material_01.material_05	DB106.DBB8	
L	"MES_RYKO_Permission".RYKO_Mat_1[6]	DB250.DBB7	-- RYKO+Silo permissions
T	"Bar_Code list from PC".material_01.material_06	DB106.DBB9	
L	"MES_RYKO_Permission".RYKO_Mat_1[7]	DB250.DBB8	-- RYKO+Silo permissions
T	"Bar_Code list from PC".material_01.material_07	DB106.DBB10	

Network 2: RYKO material 1. Silo permissions (targets 1-15)

```
CALL "BLKMOVE"          FC99
source_DB      :=250
source_DBW    :=9
target_DB     :=106
target_DBW   :=18
quantity_BYTE:=60
RET_VAL      :=#RET          #RET
```

KUVA 48. Tiedonsiirtäminen DB250 ja DB106 Data Blockien välillä

RYKO_Mat-tagin on 67-merkkiä pitkä string, jossa ensimmäiset seitsemän merkkiä sisältävät RYKO-tiedon. RYKO-tiedon jälkeiset merkit sisältävät kunkin sallitun siilon siilotunnuksen neljän merkin ryhmässä.

Samalla, kun tuotannonohjausjärjestelmä on lähettänyt siilon lupataulun pneumaattiselle siirtojärjestelmälle, täytyy sen lähettää tehty muutos myös sekoituskonelinjan siilokarttaan. Siilokartta on tallennettu sekoituskonelinjan ohjelmoitavan logiikan DB141 ja DB161 Data Blockeihin. Siilokarttaa varten OPC-palvelimelle on luotu tagit (MLXX_FSX_Allowed_RYKO), jotka on linkitetty sekoituskonelinjan DB501 Data Blockiin. Sallittujen materiaalien RYKO-tiedot siirretään DB141 ja DB 161 Data Blockeihin sekoituskonelinjan käytettäväksi (kuva 49).

FC250 : RYKO-data transfer

Comment:

Network 1 : Filler Silo 1 material RYKO from MES

Transfer RYKO-data from MES OPC data block to production (DB501->DB141)

L	"MES_OPC".Filler_Silo1_RYKO[1]	DB501.DBB2	-- Filler Silo 1 material RYKO
T	"SCALE_1 SiloMatXL (1-5)".Scale_1_Silo_XL_1_to_5[1].Group_Code.ALT_CODE[0]	DB141.DBB182	-- Alternate Code for Mix32
L	"MES_OPC".Filler_Silo1_RYKO[2]	DB501.DBB3	-- Filler Silo 1 material RYKO
T	"SCALE_1 SiloMatXL (1-5)".Scale_1_Silo_XL_1_to_5[1].Group_Code.ALT_CODE[1]	DB141.DBB183	-- Alternate Code for Mix32
L	"MES_OPC".Filler_Silo1_RYKO[3]	DB501.DBB4	-- Filler Silo 1 material RYKO
T	"SCALE_1 SiloMatXL (1-5)".Scale_1_Silo_XL_1_to_5[1].Group_Code.ALT_CODE[2]	DB141.DBB184	-- Alternate Code for Mix32
L	"MES_OPC".Filler_Silo1_RYKO[4]	DB501.DBB5	-- Filler Silo 1 material RYKO
T	"SCALE_1 SiloMatXL (1-5)".Scale_1_Silo_XL_1_to_5[1].Group_Code.ALT_CODE[3]	DB141.DBB185	-- Alternate Code for Mix32
L	"MES_OPC".Filler_Silo1_RYKO[5]	DB501.DBB6	-- Filler Silo 1 material RYKO
T	"SCALE_1 SiloMatXL (1-5)".Scale_1_Silo_XL_1_to_5[1].Group_Code.ALT_CODE[4]	DB141.DBB186	-- Alternate Code for Mix32
L	"MES_OPC".Filler_Silo1_RYKO[6]	DB501.DBB7	-- Filler Silo 1 material RYKO
T	"SCALE_1 SiloMatXL (1-5)".Scale_1_Silo_XL_1_to_5[1].Group_Code.ALT_CODE[5]	DB141.DBB187	-- Alternate Code for Mix32
L	"MES_OPC".Filler_Silo1_RYKO[7]	DB501.DBB8	-- Filler Silo 1 material RYKO
T	"SCALE_1 SiloMatXL (1-5)".Scale_1_Silo_XL_1_to_5[1].Group_Code.ALT_CODE[6]	DB141.DBB188	-- Alternate Code for Mix32

KUVA 49. Siiloihin sallittujen materiaalien RYKO-tietojen siirtämien

FC250-funktio kutsutaan OB1:ssä jokaisella ohjelmakierrolla. FC250 siirtää tuotannon-ohjausjärjestelmän tekemät RYKO-muutokset sekoituskonelinjan käytettäväksi.

6.2.2 Siilojen rajatiedot

Jokaiseen ulko- ja päiväsiiloon on asennettuna kolme anturia, joita käytetään valvomo-ohjelmiston visualisointiin sekä siilon täyttämiseen. Jatkossa visualisointi haluttiin toteuttaa myös tuotannonohjausjärjestelmässä. Tämän takia anturitiedot siirretään OPC-palvelimelle tuotannonohjausjärjestelmän luettavaksi.

Siilojen antureita ovat ala-, pyynti- ja yläraja:

- Alaraja ilmaisee, kun siilo on lähes tyhjä.
- Pyyntiraja aktivoi pneumaattisen siirtojärjestelmän työlistalle työn siilon täyttöä varten.
- Yläraja ilmaisee, kun siilo on täynnä sekä pysäyttää siilon täyttämisen.

Ohjelma luotiin STL-kielellä. STL-kielellä ohjelma oli nopeasti toteutettavissa sekä helposti luettavissa. Jokaisen siilon anturitiedot kirjoitettiin DB500 Data Blockiin. Kuvassa 50 on esitetty esimerkki siilojen rajatieto -ohjelmasta.

FC82 : Silo level

Transfer silo level sensor data to OPC data block for MES to use

```

Network 1: Storage silo (outside silo) material level data to MES
// Storage silo 1
AN  "LIA12"                I8.0          -- silo 1 Level ind. silo not max (0 = mat.)
=   "MES_OPC_Pneuma".Silo_Line[0].Silo_1_High  DB500.DBX0.0

AN  "LIA19"                I8.1          -- silo 1 Level ind. silo not med. (0 = mat.)
=   "MES_OPC_Pneuma".Silo_Line[0].Silo_1_Medium DB500.DBX0.1

AN  "LIA26"                I8.2          -- silo 1 Level ind. silo min (0 = mat.)
=   "MES_OPC_Pneuma".Silo_Line[0].Silo_1_Low   DB500.DBX0.2

// Storage silo 2
AN  "LIA13"                I8.6          -- silo 2 Level ind. silo not max (0 = mat.)
=   "MES_OPC_Pneuma".Silo_Line[0].Silo_2_High  DB500.DBX1.0

AN  "LIA20"                I8.7          -- silo 2 Level ind. silo not med. (0 = mat.)
=   "MES_OPC_Pneuma".Silo_Line[0].Silo_2_Medium DB500.DBX1.1

AN  "LIA27"                I9.0          -- silo 2 Level ind. silo min (0 = mat.)
=   "MES_OPC_Pneuma".Silo_Line[0].Silo_2_Low   DB500.DBX1.2

```

KUVA 50. FC82 Siilojen anturitietojen kirjoittaminen

FC82-funktio päivittää joka ohjelmakerrolla antureiden tilatiedon DB500 Data Blockiin, josta OPC-palvelin päivittää sen OPC-tagiin. Tuotannonohjausjärjestelmän OPC UA-asiakas lukee tagien arvoja ja päivittää siilojen antureiden tilatietoja käyttöliittymään käyttäjän nähtäväksi.

6.2.3 Päiväsiilojen täyttökielto

Pneumaattisen siirtojärjestelmän päiväsiilojen täyttökielto -toiminnallisuutta hallittiin valvomo-ohjelmistosta ja toiminnallisuus haluttiin saada myös tuotannonohjausjärjestelmään. Kun päiväsiilo asetetaan täyttökieltoon, ei pneumaattinen siirtojärjestelmä siirrä materiaalia ulkosiiloista kyseiseen päiväsiiloon.

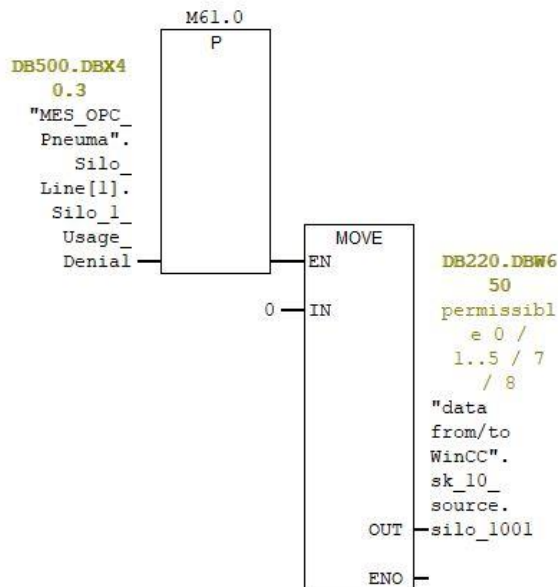
Kun siilo asetetaan täyttökieltoon, pneumaattisen siirtojärjestelmän valvomo-ohjelmisto nollaa skriptillä päiväsiilolle asetetun siirtoluvan. Nollattu siirtolupa tallennetaan DB220 Data Blockiin. Siilon täyttö vapautetaan valitsemalla valvomo-ohjelmistosta päiväsiilolle sallittu ulkosiilo, jolloin ulkosiilon tunnus siirretään aiemmin nollatun siirtoluvan tilalle.

Jotta toiminnallisuutta voitaisiin hallita myös tuotannonohjausjärjestelmästä, oli jokaiselle päiväsiilolle luotu bool-tyyppiset tagit. Käyttäjän valittua päiväsiilo täyttökieltoon, tuotannonohjausjärjestelmä asettaa MLXX_Filler_SiloX_Usage_Denied-tagin tilaan tosi. Käytettävät tagit oli linkitetty pneumaattisen siirtojärjestelmän ohjelmoitavan logiikan DB500 Data Blockiin. Ohjelman havaitessa osoitteessa tilamuutos nollassa ykköseen (nouseva reuna), kirjoitetaan täyttökieltoon asetettavalle päiväsiilolle varattuun integer-tyyppiseen osoitteeseen arvo 0 (kuva 51).

FC83 : MES Silo fill denial control

Comment:

Network 1: SK10 Filler Silo 1 Denial

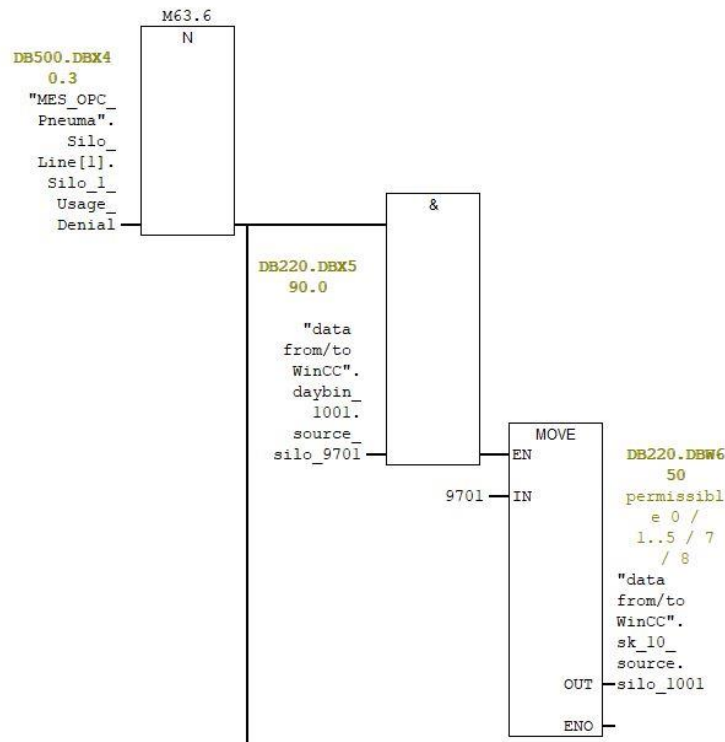


KUVA 51. Päiväsiilolle asetetun siirtoluvan nollaus

Jokaiselle päiväsiilolle toteutettiin oma täyttöluvan nollaus siilokohtaisen osoitteen perusteella. Ohjelma suoritetaan jokaisella PLC:n ohjelmakierrolla. Osoitteen nollaus suoritetaan vain osoitteen nousevan reunan perusteella.

Käyttäjän vapauttaessa päiväsiilo pois kiellosta, tuotannonohjausjärjestelmä asettaa MLXX_Filler_SiloX_Usage_Denied-tagin tilan arvoon epätosi (*engl. false*). Ohjelman havaitessa tagiin linkitetyn Data Blockin osoitteessa muutos ykkösestä nolnaan (laskeva reuna), suorittaa ohjelma lupatauluvertailun. Vertailun tuloksen perusteella vapautettavan päiväsiilon integer-tyyppiseen osoitteeseen kirjoitetaan sallittavan ulkosiilon tunnus (kuva 52).

□ Network 23: SK10 Silo 1 permission recover

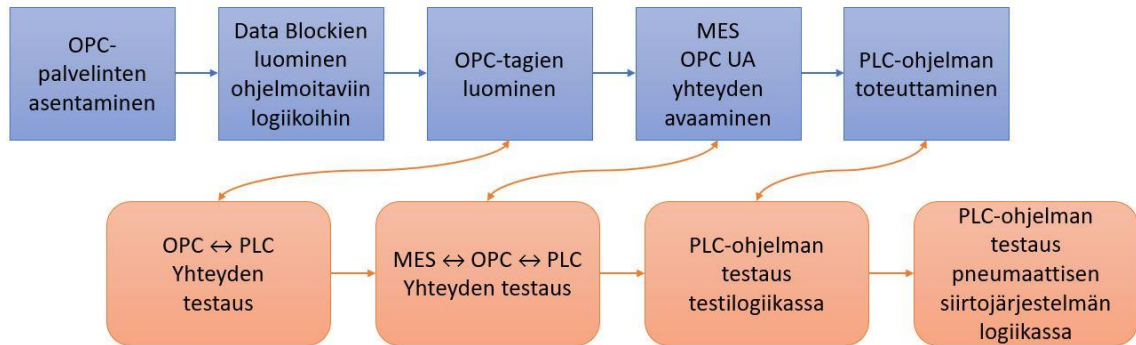


Kuva 52. Siilon täyttöluvan palauttaminen

Lupavertailu suoritetaan jokaiselle päiväsiilolle niin, että jokainen ulkosiilo vertaillaan siilokohtaisesti. Esimerkiksi, jos vapautettavalle siilolle on sallittu lupa siirtää ulkosiilosta 1, kirjoitettaisiin päiväsiilon integer-tyyppiseen osoitteeseen ulkosiilon 1 tunnus 9701. Ohjelma suoritetaan jokaisella ohjelmakierrolla. Päiväsiilolle sallittu ulkosiilon tunnus kirjoitetaan kuitenkin vain silloin, kun käyttäjä vapauttaa päiväsiilon tuotannonohjausjärjestelmän käyttöliittymästä.

7 TESTAUS JA KÄYTTÖÖNOTTOSUUNNITELMA

Integroidun järjestelmän testausta suoritettiin projektin edetessä vaiheittain. Kuvassa 53 on esitetty järjestelmään liittyvän testauksen vaiheet.



KUVA 53. Pneumaattisen siirtojärjestelmän testauksen eri vaiheet

Testaus aloitettiin, kun tarvittavat Data Blockit oli ladattu ohjelmoitaviin logiikoihin ja tagit oli luotu OPC-palvelimelle. OPC-palvelimen ja ohjelmoitavien logiikoiden välistä yhteyttä testattiin kirjoittamalla OPC-tageihin arvoja suoraan OPC-palvelimelta, jolloin arvojen tuli päivittyä myös ohjelmoitavan logiikan Data Blockiin.

Kun OPC UA -yhteys oli avattu tuotannonohjausjärjestelmän ja OPC-palvelimen välille, voitiin tuotannonohjausjärjestelmän testiympäristöstä kirjoittaa arvoja OPC-palvelimen tageihin. Tageista tieto siirtyi ohjelmoitavan logiikan Data Blockeihin.

Pneumaattisen siirtojärjestelmän ohjelmoitavan logiikan ohjelmaa kehitettiin ja testattiin erillisellä testilogiikalla. Testiympäristössä Data Blockeihin kirjoitettiin suoraan ohjelmoitavasta logiikasta samaa dataa, mitä tuotannonohjausjärjestelmä kirjoittaisi. PLC-ohjelma oli mahdollista testata kattavasti testilogiikassa, mikä vähensi ongelmia käyttöönotossa.

Sekoituskonelinjoille ei ollut erillisiä testilogiikoita, vaan OPC-palvelimen tagit oli linkitetty suoraan tuotannossa olevien sekoituskonelinjojen ohjelmoitaviin logiikoihin. Tästä ei aiheutunut haittaa tuotannolle, koska tageihin linkitetyn DB501 Data Blockin tietoja ei siirretty sekoituskonelinjan siilokarttaan (DB141 ja DB161) käytettäväksi.

Kun tuotannonohjausjärjestelmän toimittaja oli saanut testiympäristön toiminnallisuudet valmiiksi, voitiin tuotannonohjausjärjestelmän testiympäristön käyttöliittymästä tehdä muutoksia siilojen materiaaleihin. Muutoksien perusteella tuotannonohjausjärjestelmän tuli lähettää oikeaa tietoa pneumaattisen siirtojärjestelmän sekä sekoituskonelinjojen OPC-tageihin. Kun testaus oli tehty jokaisen siilon osalta ja todettu, että siiloluvat sekä siilokartta päivittyivät ohjelmoitaviin logiikoihin oikein, oli käyttöönotto mahdollista aloittaa.

7.1 Käyttöönoton riskianalyysi

Ennen käyttöönoton aloittamista oli tärkeää pohtia siihen liittyviä riskejä. Riskejä pyrittiin kartoittamaan nelikenttäanalyysin avulla. Kuvassa 54 on esitetty käyttöönottoa käsittelevä SWOT-analyysi.

<p>S Strengths</p> <ul style="list-style-type: none"> •Mahdollista testata ja käyttöönottaa osittain, vaikka tuotanto olisi käynnissä •Ammattitaitoinen projektiryhmä 	<p>W Weaknesses</p> <ul style="list-style-type: none"> •Osaston tuotannon kannalta kriittinen järjestelmä •Resurssit vähäiset projektihenkilöstön suhteen •Henkilöstön koulutus vaatii resursseja •Järjestelmän toiminta vain muutamien henkilöiden tiedossa •Tiukka aikataulu, osaston muut projektit huomioon ottaen
<p>O Opportunities</p> <ul style="list-style-type: none"> •Skaalattavissa useisiin tuotantolaitoksiin •Säästöt turhan työn vähenemisen myötä •Tuotteiden laadun parantuminen •Tuotannon jatkuva kehittäminen •Käytettävyyden parantuminen •Tiedon saatavuus ja läpinäkyvyys •Järjestelmä helpottaa tuotannon hallittavuutta 	<p>T Threats</p> <ul style="list-style-type: none"> •Mahdolliset tuotannon menetykset, mikäli käyttöönotto ei toteudu suunnitellusti •Väärän materiaalin syöttäminen tuotantolinjalle ja sitä kautta ajettaviin tuotteisiin •Tuotannon menetykset järjestelmän vikatilanteissa •Puutteellisen käyttäjäkoulutuksen aiheuttamat vikatilanteet

KUVA 54. Käyttöönottoa varten tehty SWOT-analyysi

Pneumaattinen siirtojärjestelmä on osaston tuotannon kannalta tärkeä järjestelmä. Mikäli pneumaattisen siirtojärjestelmän toiminta pysähtyisi riittävän pitkäksi aikaa, ei osastolta valmistuisi materiaalia muille tehtaan osastoille. Lopulta tehdas ei kykenisi valmistamaan tuotteitaan.

Tuotantomenetyksien lisäksi riskiksi havaittiin osaston tuotteen laatupoikkeamat. Laatu-poikkeamat johtuisivat vikatilanteesta, jossa järjestelmä syöttäisi tuotannossa oleviin resepteihin väärää materiaaleja. Pahimmassa tapauksessa laatu-poikkeamat huomattaisiin vasta tehtaan muilla osastoilla, jolloin virheellisiä tuotteita olisi tuotettu jo suuria määriä.

Käyttöönnotossa on mahdollista ilmetä ongelmia liittyen pneumaattisen siirtojärjestelmän PLC-ohjelmaan. Koska testilogiikkaan voitiin ladata vain osa koko järjestelmän ohjelmalohkoista, ei kaikkien ohjelmalohkojen yhteentoimivuutta voitu testata ennakkoon. Näin ollen ei voitu olla varmoja, miten uusi PLC-ohjelma tulisi toimimaan kaikkien pneumaattisen siirtojärjestelmän PLC-ohjelmien kanssa. Riski liittyisi myös siihen, miten siilojen lupataulu -ohjelma tulisi toimimaan siilon täyttökielto -ohjelman kanssa. Ohjelmien toiminnallisuudet liittyvät yhteen. Vanhassa järjestelmässä siilojen täyttökieltoa hallittiin vain valvomo-ohjelmistosta.

Riskien vähentämiseksi testaukseen ja käyttöönottoon tulisi varata riittävästi aikaa ja resursseja. Ennen varsinaista tuotannon pysäyttämistä, testausta ja käyttöönottoa tulisi suorittaa niin pitkälle kuin mahdollista. Käyttöönotto tulisi suorittaa osissa, mikäli se olisi mahdollista.

7.2 Käyttöönottosuunnitelma

Ennen käyttöönoton aloittamista tulee pneumaattisen siirtojärjestelmän sekä sekoituskonelinjojen ohjelmoitavien logiikoiden ohjelmasta ottaa Online-varmuuskopiot. Online-varmuuskopiossa Data Blockien sen hetkiset arvot siirtyvät ohjelman varmuuskopion mukana. Näin mahdollisessa ongelmatilanteessa Data Blokkeissa käytettyjen arvojen palautus olisi mahdollista.

Pneumaattisen siirtojärjestelmien ohjelmien käyttöönotto olisi suositeltavaa aloittaa siilojen anturien tilatietojen lähettämällä tuotannonohjausjärjestelmään. Siilojen rajatieto-ohjelma olisi mahdollista käyttöönottaa ilman tuotannon pysäyttämistä. Rajatieto -ohjelman toimivuus olisi helppoa ja nopea todeta vertaamalla tuotannonohjausjärjestelmän käyttöliittymän visualisointeja sen hetkisiin anturitietoihin.

Seuraavaksi tulisi käyttöönottaa pneumaattiseen siirtojärjestelmään siilolupien hallintaan tarkoitettu ohjelma. Ohjelman käyttöönottoa varten tuotanto-osastolta varataan aikaa, jolloin pneumaattinen siirtojärjestelmä voisi olla pois käytöstä. Käyttöönotosta voi aiheutua tilanne, jossa osalla sekoituskonelinjoista ei voida ajaa tuotantoa puuttuvien materiaalien takia.

Kun siilolupien hallintaan tarkoitettu ohjelma on ladattu pneumaattisen siirtojärjestelmän logiikkaan ja asetettu ohjelmakiertoon, tulee järjestelmälle suorittaa samanlaista testausta kuin testilogiikassa. Tuotannonohjausjärjestelmän käyttöliittymästä muutetaan siilojen materiaaleja ja tarkistetaan, että tieto siirtyy oikein pneumaattisen siirtojärjestelmän PLC:n siilolupatauluun (DB106) sekä sekoituskonelinjojen OPC-tageihin. Riittävän testaamisen jälkeen voidaan järjestelmä luovuttaa tuotantoon. Sen toimintaa seurataan aina siirto kerrallaan.

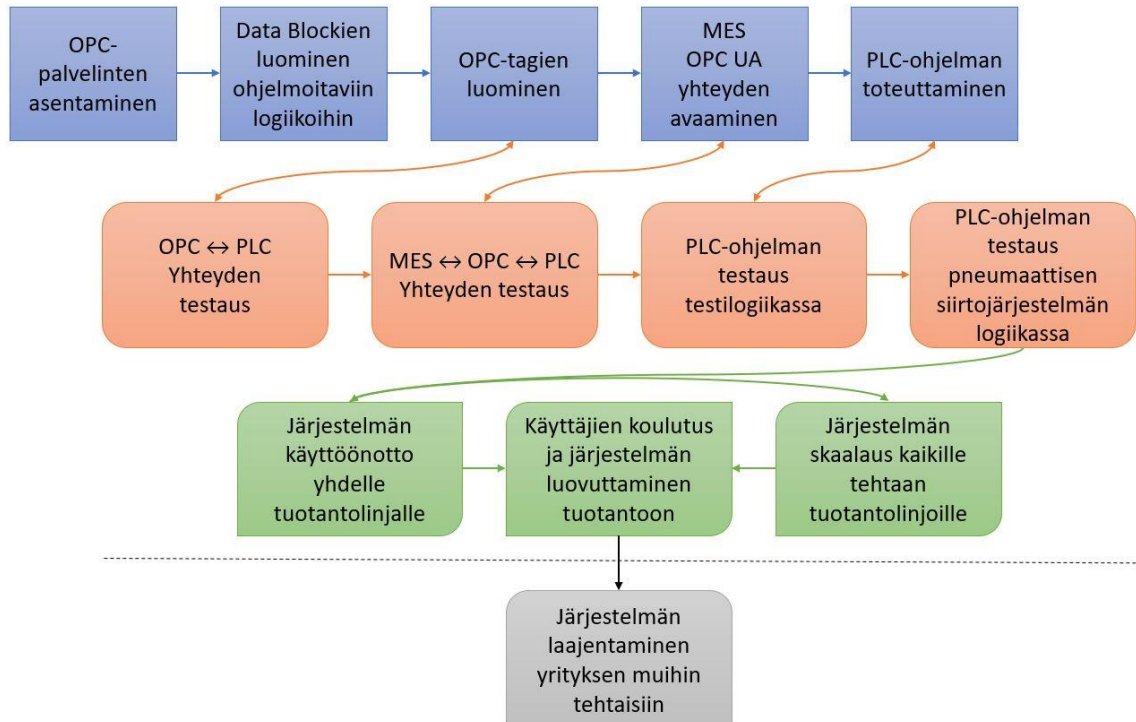
Suurin riski tässä vaiheessa käyttöönottoa liittyy ulkosiilojen täyttämiseen. Koska pneumaattinen siirtojärjestelmä luottaa lupataulussa olevaan tietoon, voi täytettävään siiloon mennä väärää materiaalia. Mikäli tuotannonohjausjärjestelmän lähettämä tieto olisi väärää tai pneumaattisen siirtojärjestelmän logiikka ei käsittelee tietoa oikealla tavalla, ulkosiiloon siirrettäisiin väärää materiaalia.

Siilokarttaa ei vielä alkuvaiheessa siirretä OPC-tageista sekoituskonelinjojen käyttöön (DB141 ja DB161). Näin voidaan välttää väärän materiaalin siirtäminen ulkosiilosta päiviisiiloon tilanteessa, jossa tuotannonohjausjärjestelmä lähettäisi virheellistä tietoa sekä pneumaattiseen siirtojärjestelmään että sekoituskonelinjalle.

Kun testausta on suoritettu riittävä määrä ja järjestelmä on toimintavarma, voidaan siilokartan tietoja alkaa siirtää OPC-tageista myös sekoituskonelinjojen siilokarttaan (DB141 ja DB161).

Kun siilolupien ja siilokartan ohjelma on käyttöönotettu sekä pneumaattiseen siirtojärjestelmään että sekoituskonelinjoille, voidaan käyttöönottaa siilojen täyttökielto -ohjelma. Testauksessa tuotannonohjausjärjestelmästä tulisi asettaa jokainen päiviisiilo yksi kerrallaan täyttökieltoon. Samalla todennetaan pneumaattisen siirtojärjestelmän logiikasta, että siilolle ei ole sallittu täytettävää materiaalia. Tämän jälkeen päiviisiilot tulee vapauttaa

yksi kerrallaan samalla monitoroiden, että siilo saa materiaalin täyttöluvan. Testi pitää tehdä vielä niin, että oikeaa materiaalia syötetään käsiajolla kiellossa olevaan siiloon. Näin voidaan havaita, että järjestelmä toimii myös käytännössä. Kun järjestelmän kaikki toiminnallisuudet on testattu, voidaan järjestelmä luovuttaa lopullisesti tuotannon käyttöön, samalla käyttäjiä kouluttaen (kuva 55).



KUVA 55. Käyttöönoton toteutus

Kuvassa 55 on esitetty pneumaattisen siirtojärjestelmän integraation vaiheittainen toteutus, testaus ja käyttöönotto tuotannonohjausjärjestelmään. Tämän suunnitelman pohjalta pneumaattisen siirtojärjestelmän integraatio voidaan toteuttaa myös kohdeyrityksen muiden tehtaiden tuotannonohjausjärjestelmiin.

8 POHDINTA

Opinnäytetyön tavoitteena oli toteuttaa pneumaattisen siirtojärjestelmän integraatio tuotannonohjausjärjestelmään kirjallisuustutkimusta hyödyntäen. Kirjallisuustutkimusta tehtiin järjestelmän jokaiselta osa-alueelta, mikä selkeytti kokonaiskuvaa järjestelmän rakenteesta ja toiminnasta.

Järjestelmän integraatiotyö toteutettiin niin, että se on hyvin pienin muutoksin laajennettavissa yrityksen Venäjällä sijaitsevaan tuotantolaitokseen. On kuitenkin mahdollista, että yrityksen uuden USA:ssa sijaitsevan tehtaan pneumaattinen siirtojärjestelmä poikkeaa ratkaisuiltaan työssä esitetystä järjestelmästä. Tällöin PLC-ohjelmien koodi ole kopioitavissa sellaisenaan uuden tehtaan järjestelmään. Vaikka järjestelmät poikkeaisivat toisistaan, hyödynnettävissä olisivat käytettävät tiedonsiirtotekniikat sekä materiaalinvarmistukseen liittyvät toimintaperiaatteet.

Pneumaattisen siirtojärjestelmän integraation testaaminen saatiin toteutetuksi testilogiikan ja tuotannonohjausjärjestelmän testiympäristön avulla. Opinnäytetyössä integroidun järjestelmän toiminnan kannalta ei ole merkitystä, tuleeko tuotannonohjausjärjestelmän lähettämä tieto OPC-palvelimelle testi- vai tuotantoympäristöstä. Jotta opinnäytetyössä esitetyt ratkaisut toimisivat myös tuotantoympäristössä edellyttää se sitä, että tuotannonohjausjärjestelmän toimittaja on toteuttanut testi- ja tuotantoympäristön toiminnallisuudet samalla tavalla. Jos testi- ja tuotantoympäristön toiminnallisuudet poikkeavat toisistaan, eivät opinnäytetyössä toteutetut ratkaisut toimi sellaisenaan.

Aikataulusta johtuen opinnäytetyössä ei käsitelty järjestelmän lopullista käyttöönottoa. Tulen todennäköisesti suorittamaan järjestelmän lopullisen käyttöönoton, koska minulla on tietämys järjestelmän rakenteesta ja toiminnasta. On todennäköistä, että käyttöönotossa ilmenee jotakin sellaisia ongelmia, mitä ei huomattu tai ei voitu havaita, kun järjestelmää testattiin testiympäristön avulla.

Suurimpina haasteina opinnäytetyössä oli pneumaattisen siirtojärjestelmän alkuperäisen toiminnan selvittäminen. Tämä johtui siitä, että ihmiset, joilla oli tuntemus järjestelmän toiminnasta, eivät olleet enää yrityksen palveluksessa. Opinnäytetyötä tehdessä uusina asioina tulivat myös OPC-palvelimen toiminta, asentaminen ja konfiguroiminen. OPC-

palvelimen kirjallisuustutkimus auttoi hahmottamaan, mitä ominaisuuksia ja rajoitteita OPC-palvelimilla on. Kollegan apua hyödyntäen OPC-palvelin saatiin asennettua ja konfiguroitua käyttötarkoitukseen sopivaksi.

Opinnäytetyön kirjallisuustutkimuksen lähdemateriaaleista suurin osa on englanninkielisistä lähteistä. Kielikäännökset heijastuvat paikoin tekstistä tökerönä suomenkielenä, koska osalle vierasperäisistä sanoista ei ole suomenkielistä vastinetta. Toisinaan tuntui, että koko lähdemateriaaleista kerätty teksti olisi ollut helpompi toteuttaa englanninkielellä.

Opinnäytetyölle asetetut tavoitteet saavutettiin, mutta olisin halunnut käsitellä opinnäytetyössäni myös pneumaattisen siirtojärjestelmän integraation lopullista käyttöönottoa. Tämä olisi ollut mahdollista, mikäli projekti olisi toteutettu aikataulullisesti aikaisemmassa vaiheessa.

8.1 Järjestelmän jatkokehitys

Järjestelmän jatkokehityksen kannalta on tärkeää pohtia, mihin tarpeisiin kehityksen tulisi vastata. Suomen ja Venäjän tehtailla järjestelmä on toteutettu Siemensin Simatic S7-300 ja S7-400 -sarjan ohjelmoitavilla logiikoilla. Yhdysvaltoihin rakennettavan tehtaan järjestelmä toteutetaan Siemensin uusilla S7-1500 -sarjan ohjelmoitavilla logiikoilla. Kohdeyrityksessä automaatiojärjestelmät on pyritty toteuttamaan samanlaisiksi tehtaiden välillä. Tällöin olisi järkevää päivittää vanhat S7-300 ja S7-400 -sarjan ohjelmoitavat logiikat uusiin S7-1500 ohjelmoitaviin logiikoihin. Näin järjestelmät olisivat yhtenevät tehtaiden välillä sekä tulevaisuudessa varaosien saatavuus olisi turvattu.

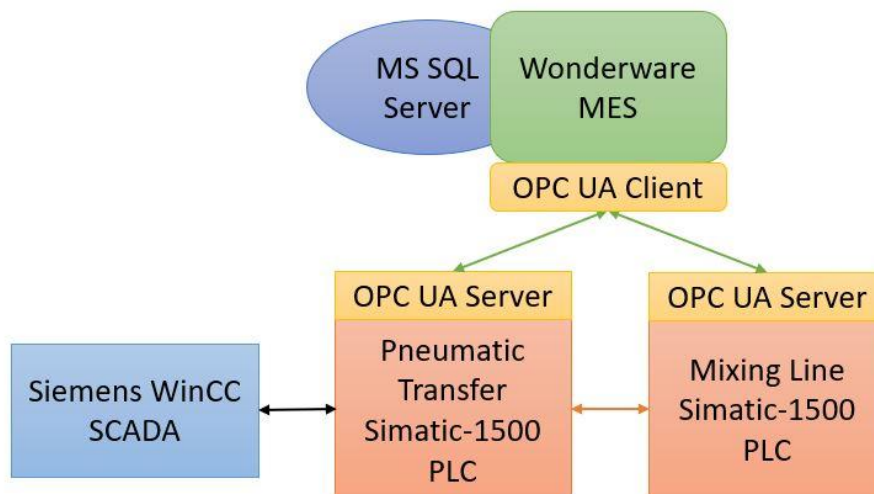
Nokian tehtaalla oleva pneumaattisen siirtojärjestelmä valvomo-ohjelmiston palvelin on toteutettu fyysisellä tietokoneella. Valvomo-ohjelmiston voisi toteuttaa virtuaalikoneella. Näin toteutus olisi yhtenevä Venäjän tehtaan ja tulevan USA:n tehtaan toteutuksen kanssa. Virtuaalikoneelle toteutetulla palvelimella järjestelmän hallittavuus ja ylläpito helpottuisi sekä järjestelmä ei olisi haavoittuvainen tietokoneen rikkoontumiselle.

Päiväsiilojen täyttökielto -toiminnallisuus

Pneumaattisen siirtojärjestelmän päiväsiilojen täyttökiellon -toiminnallisuutta olisi mahdollista muuttaa niin, että koko toiminnallisuus olisi hallittavissa vain tuotannonohjausjärjestelmästä. Tämä vaatisi, että kun siilo laitetaan täyttökieltoon, lähettäisi tuotannonohjausjärjestelmä uuden siilolupataulun pneumaattiseen siirtojärjestelmään. Lähetetty siilolupataulu ei sisältäisi täyttökiellossa olevan siilon siilotunnusta. Käyttöön otettavassa järjestelmässä on se etu, että siilo on mahdollista laittaa täyttökieltoon sekä tuotannonohjausjärjestelmästä että pneumaattisen siirtojärjestelmän valvomo-ohjelmistosta.

Ohjelmoitavan logiikan ja tuotannonohjausjärjestelmän välinen kommunikointi

Ohjelmoitavien logiikkojen ja tuotannonohjausjärjestelmän välinen OPC UA -kommunikointi olisi mahdollista toteuttaa S7-1500 ohjelmoitavan logiikan omalla OPC UA -palvelimella (kuva 56).

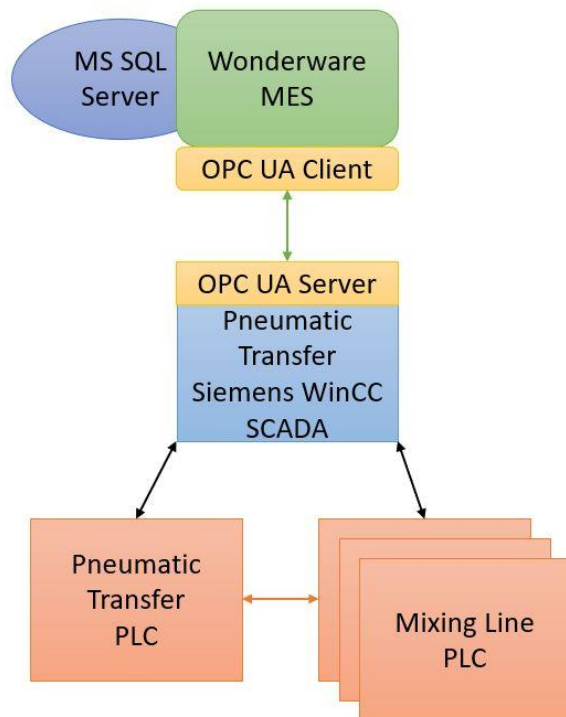


KUVA 56. OPC UA -kommunikointi suoraan ohjelmoitavasta logiikasta. Esimerkkiratkaisu toteutettuna pneumaattiseen siirtojärjestelmään

OPC UA -palvelimen käyttäminen ohjelmoitavassa logiikassa poistaisi Kepwaren OPC-palvelimen ohjelmoitavan logiikan ja tuotannonohjausjärjestelmän väliltä. Siemensin ohjelmoitavassa logiikassa toteutettava OPC UA -palvelin vaatii erillisen lisenssin. Lisenssien hinnat vaihtelevat 100-300€ palvelinta kohti riippuen ohjelmoitavan logiikan mallista. Lisenssi on käytännössä sertifikaatti siitä, että toiminnallisuutta saa käyttää ohjelmoitavassa logiikassa. Kyseessä ei siis ole Siemensille tyypillinen lisenssitiedosto, joka

siirretään käytettävään laitteeseen. Tulevaisuudessa voi olla mahdollista, että Siemens ei vaadi ohjelmoitavassa logiikassa käytettävälle OPC UA -palvelimelle erillistä lisenssi-sertifikaattia. Siemensin ohjelmoitavan logiikan oma OPC UA -palvelin voi hyvin soveltua joihinkin käyttökohteisiin, mutta itse en näkisi perusteltuna korvata nykyistä Kepwaren OPC-palvelinta sillä.

Toinen vaihtoehto kommunikoinnin toteuttamiselle olisi hyödyntää Siemensin valvomo-ohjelmiston OPC-rajapintaa (kuva 57). Kommunikointi olisi toteutettu tässäkin tapauksessa OPC UA -yhteydellä. Tällä toteutuksella Kepwaren OPC-palvelin poistuisi käytöstä.



KUVA 57. OPC UA -kommunikointi valvomo-ohjelmiston avulla. Esimerkkiratkaisu toteutettuna pneumaattiseen siirtojärjestelmään

Järjestelmän toteutus riippuu siitä, halutaanko tuotannonohjausjärjestelmän ja ohjelmoitavien logiikkojen väliset tagit hallita yhdellä OPC-palvelimella vai järjestelmäkohtaisesti. Nykyisessä Kepwaren OPC-palvelimessä on hyvät ominaisuudet tagien käsittelyyn ja tiedonsiirto on varmistettu kahdella toisistaan erillisillä palvelimilla.

LÄHTEET

ABB Oy. 2000-07. TTT-käsikirja. Automaation tietoliikennetekniikka. Luettu 31.03.2018

http://www.oamk.fi/~kurki/automaatiolabrat/TTT/05_0_Automaation%20tietoliikenne.pdf

Almada-Lobo, F. 2015. Consulting-Specifying Engineer: Define, integrate, implement MES with controls, ERP. Luettu 24.08.2018

<https://www.csemag.com/single-article/define-integrate-implement-mes-with-controls-erp.html?print=1>

Asmala, H., Koskinen, K., Koskela, M., Mätäsniemi, T., Soini, A., Strömman, M., Tommila, T., Valkonen, J. 2005. Automaatiosovellusten ohjelmistokehitys. Suomen automaation tuki Oy.

Bolton, W. 2006. Programmable Logic Controllers. Oxford: Elsevier Newnes

Brandl, D. 2008. What is ISA-95? Industrial Best Practices of Manufacturing Information Technologies With ISA-95 Models. Luettu 24.08.2018

http://www.apsom.org/docs/T061_isa95-04.pdf

Digia. Tuotannonohjaus. Luettu 21.08.2018

<https://digia.com/palvelumme/tuotannonohjaus>

Hackworth, J. R. & Hackworth Jr. F. D. 2004. Programmable Logic Controllers: Programming Methods and Applications. New Jersey: Pearson Education, Inc.

Hanssen, D. 2015. Programmable Logic Controllers. A Practical Approach To IEC 61131-3 Using CODESYS. 1. painos. Englanti: John Wiley & Sons, Ltd.

Hovi, A., Huotari, J. & Lahdenmäki, T. 2005. Tietokantojen suunnittelu ja indeksointi. Jyväskylä: Docendo Finland Oy.

Härkönen, P., Mikkola, J., Piikkilä, V., Sahala, A., Sahlsten, T., Sanström, B., Sirviö, A., Spangar, T., Sulku, J. 2012. Rakennusautomaatiojärjestelmät. Tietotekniset järjestelmät. Espoo: Sähkötieto Oy

IEC 61131-3: a standard programming resource. Verkkodokumentti PLCopen.

http://www.plcopen.org/pages/tc1_standards/downloads/intro_iec.pdf Luettu 17.03.2018

Iwanitz, F. & Lange, J. 2002, 2nd Edition. OPC Fundamentals, Implementation and Application. Heidelberg: Hüthig GmbH.

Keinänen, T., Kärkkäinen, P., Lähetkangas, M. & Sumujärvi, M. 2007. Automaatiojärjestelmien logiikat ja ohjaustekniikat. Helsinki: WSOYpro Oy.

Kletti, J. 2007. Manufacturing Execution System – MES

Mahnke, W., Leitner, S.-H. & Damm M. 2009. OPC Unified Architecture. Berlin: Springer-Verlag GmbH.

Mahnke, W. & Leitner, S.-H. OPC Unified Architecture: The future standard for communication and information modeling in automation. Luettu 02.01.2018.
https://library.e.abb.com/public/75d70c47268d78bfc125762d00481f78/56-61%203M903_ENG72dpi.pdf

Mehta, B. R. & Reddy, Y. J. 2015. Industrial Process Automation Systems, Design and Implementation. Amsterdam: Elsevier Inc.

OPC Datahub. Cogent Real-Time Systems Inc. What is OPC? Luettu 13.06.2017.
<http://www.opcdatahub.com/WhatIsOPC.html>

OPC Foundation, What is OPC UA. Luettu 04.03.2018.
<https://opcfoundation.org/about/opc-technologies/opc-ua/>

Paine, T. 2013. Understanding OPC: Open Connectivity via Open Standards. Luettu 12.06.2017.
https://www.automation.com/pdf_articles/Understanding OPC_Kepware_eBook.pdf

Pathak, N. 2008. Database Management System. Mumbai: Himalaya Publishing House

Watson, R. T. 2006. Data Management: Databases and Organizations. 5. painos: John Wiley & Sons, Inc

What is OPC? Luettu 12.06.2017.
<https://opcfoundation.org/about/what-is-opc/>

Unified Automation, OPC UA Specifications. Luettu 11.03.2018
<http://documentation.unified-automation.com/uasdkcpp/1.4.3/html/L2OpcUaSpecifications.html>

Yli-Knuuttila, E. 2014. Siemens Simatic -oppimisympäristön rakentaminen. Automaatiotekniikan koulutusohjelma. Seinäjoen ammattikorkeakoulu. Opinnäytetyö