

Antti Salminen

Raportointialustojen tekninen vertailu kerättävälle prosessitiedolle

Opinnäytetyö

Tekniikan ammattikorkeakoulututkinto

Energiatekniikan koulutus

2022



**Kaakkois-Suomen
ammattikorkeakoulu**

| | |
|----------------|--|
| Tutkintonimike | Insinööri (AMK) |
| Tekijä | Antti Salminen |
| Työn nimi | Raportointialustojen tekninen vertailu kerättävälle prosessitiedolle |
| Toimeksiantaja | Ontec Oy |
| Vuosi | 2022 |
| Sivut | 122 sivua |
| Työn ohjaajat | Merja Mäkelä Xamk; Esa Käyhty Ontec Oy |

TIIVISTELMÄ

Opinnäytetyön tavoitteena oli kartoittaa tiedonkeruun ja raportoinnin mahdollisuudet Siemens WinCC Unified -valvomoympäristössä. Tarkasteltavina raportointialustoina olivat Siemensin Process Management -lisäosakokonaisuuteen kuuluvat PM-QUALITY ja PM-ANALYZE sekä WinCC Unified -valvomosovelluksen lisenssiin sisältyvä raportointilisäosa WinCC Unified Report Execution. Lisäksi työssä tarkasteltiin Siemensin omia rajapintoja Unified Runtime Openness ja Open Pipe sekä yleistä Open Database Connect -tietokantarajapintaa.

Opinnäytetyöraportti koostuu kolmesta osiosta. Kirjallisuusosiossa pohjustetaan lukijalle, mitä tieto ja tiedon jalostaminen ovat sekä miten tiedonkeruu tapahtuu järjestelmäintegraatioilla prosessiteollisuudessa. Lisäksi luodaan katsaus, miten yritykset voivat tarjota raportointia palveluna sekä miten tiedonkeruu, tiedon jalostus ja laitekannan kehitys yhdistyvät teollistumisen neljännen aikakauden tavoitteissa. Raportin toisessa osassa, joka on työn varsinainen pääsisältö, kartoitetaan alustojen ominaisuuksia lähdemateriaalin pohjalta sekä testaamalla alustoja, jotka sisältyvät valvomosovelluksen lisenssiin. Kartoitus on tehty toimeksiantajan määrittelemän ominaisuuslistan pohjalta, minkä keskeisimmät kriteerit ovat, miten alustat soveltuvat eri mittakaavan tilanteisiin sekä miten niillä voidaan toteuttaa ajanjaksoon tai tiettyyn valmistuserään sidottuja raportteja. Raportin viimeisessä osassa esitellään itsenäisesti tehdyt johtopäätökset samaan listaukseen verraten.

Opinnäytetyön johtopäätöksinä todetaan, että PM-QUALITY-sovellus on alustoista kattavin raportoinnin sekä raportoitavan tiedon jalostamisen osalta. Sillä voidaan toteuttaa raportointia isoissakin tuotantolaitoksissa, joissa on useita tuotantoyksiköitä ja valvomoasemia. PM-ANALYZE-sovelluksen vahvuus on analysoinnissa, mutta raportoinnin näkökulmasta siinä ei ole ominaisuuksia, joilla saavutettaisiin etua PM-QUALITY-sovellukseen nähden. Rajapintoja on mahdollista hyödyntää itse toteutetuilla tai kolmannen osapuolen raportointisovelluksilla. Rajapinnoilla on pääsy saman raakatietoon kuin PM-QUALITY-sovelluksella, joten määrittäväksi tekijäksi jää rajapintaa hyödyntävän raportointisovellus. Yksinkertaisen ja taulukkomuotoisen raportoinnin toteuttaminen on suoraviivaisinta Report Execution -raportointilisäosalla, jolla raportoinnin voi toteuttaa itsenäisesti kokenut loppukäyttäjänkin.

Opinnäytetyö valmistui aikataulussa ja sen konkreettiset tulokset esiteltiin toimeksiantajalle jo projektin puolivälissä. Toimeksiantaja oli tyytyväinen tekijän omatoimisuuteen sekä kartoitustyön dokumentoinnin tarkkuuteen.

Asiasanat: tiedon jalostaminen, järjestelmäintegraatio, prosessiautomaatio, raportointi

| | |
|-----------------|--|
| Degree title | Bachelor of Engineering |
| Author | Antti Salminen |
| Thesis title | Technical comparison of reporting platforms for sampled process data |
| Commissioned by | Ontec Oy |
| Time | 2022 |
| Pages | 122 pages |
| Supervisors | Merja Mäkelä Xamk; Esa Käyhty Ontec Oy |

ABSTRACT

The objective of the thesis was to explore the data collection and reporting possibilities in Siemens WinCC Unified operator control and monitoring system. The platforms that were examined comprised PM-QUALITY and PM-ANALYZE add-ons from Siemens' Process Management tools for WinCC and reporting add-on WinCC Unified Report Execution which is included in the WinCC Unified licence. Under examination were also Siemens' own interfaces Unified Runtime Openness and Open Pipe and a universal database interface Open Database Connect.

The thesis consists of three parts. First information and information processing and management of data collection with system integrations in the process industry were examined. In addition, it was examined how companies could provide reporting as a service, and how data collection, data processing and equipment development combine in the goals of Industry 4.0. The second part is the core of the thesis, and it explored the features of reporting platforms and included a testing of functionalities of reporting platforms included in WinCC Unified licence. The study was based on a list of features defined by the commissioner. The main points of interest were the suitability of platforms for projects of different sizes, and accomplishment of reports tied to a time frame or production batch. The same list of features was used as a basis for conclusions made in the final part of the thesis.

The thesis concludes that PM-QUALITY is the most comprehensive in terms of reporting and processing reportable information. It can be used in projects conducted in large production facilities with multiple production units and control stations. The strength of PM-ANALYZE is in analysis, but it lacks the features that would bring major advantage over PM-QUALITY from a reporting perspective. Interfaces can be utilized with self-developed or third-party reporting applications and have access to the same raw data as PM-QUALITY, so the reporting application remains the main determining factor. Simple and tabular reporting can be created in the most straightforward manner with the Report Execution add-on which allows experienced end-users to independently perform reporting tasks.

Keywords: data refinement, systems integration, process automation, reporting

SISÄLLYS

| | | |
|-------|--|-----|
| 1 | JOHDANTO | 5 |
| 2 | TIEDON HIERARKIA JA JÄRJESTELMÄINTEGRAATIOT | 6 |
| 2.1 | Tiedon hierarkia | 7 |
| 2.2 | Järjestelmäintegraatiot ja tiedonsiirto..... | 16 |
| 2.2.1 | Syntaktisen ja teknisen tason järjestelmäintegraatiot | 17 |
| 2.2.2 | Järjestelmäintegraatiotasojen yhdistyminen OSI-mallin avulla | 22 |
| 2.2.3 | Yleisiä tiedonsiirtomenetelmiä | 28 |
| 2.3 | Prosessitiedon arkistointi | 38 |
| 3 | TARKASTELTAVA OHJAUSJÄRJESTELMÄYMPÄRISTÖ..... | 41 |
| 3.1 | Siemens Simatic -tuotteet..... | 43 |
| 3.2 | Kehittämistarve ohjausjärjestelmäympäristössä | 54 |
| 4 | RAPORTOINTIALUSTOJEN OMINAISUUSKARTOITUS | 56 |
| 4.1 | Process Management -lisäosat WinCC-valvomosovellukselle | 57 |
| 4.1.1 | Tuotantoerätyökalu PM-QUALITY | 70 |
| 4.1.2 | Prosessitietotyökalu PM-ANALYZE | 83 |
| 4.1.3 | PM-lisäosien Microsoft Excel -liitännäinen..... | 85 |
| 4.2 | Raportoinnin toteuttaminen rajapintojen välityksellä | 88 |
| 4.2.1 | Siemens WinCC Unified Runtime Openness -rajapinta | 89 |
| 4.2.2 | WinCC Unified Open Pipe -rajapinta | 91 |
| 4.2.3 | Open Database Connectivity -rajapinta | 93 |
| 4.3 | Raportointilisäosa WinCC Unified Report Execution | 95 |
| 5 | RAPORTOINTIALUSTOJEN SOVELTUVUUSTARKASTELU | 103 |
| 6 | YHTEENVETO | 108 |
| | LÄHTEET..... | 110 |
| | KUVALUETTELO | |

1 JOHDANTO

Tämän opinnäytetyön tavoitteena on tehdä kartoitus Siemens WinCC Unified -valvomoympäristön tiedonkeruusta ja raportoinnista. Työssä on tarkoitus tutustua käytettävään ohjausjärjestelmäympäristöön ja siihen soveltuviin raportointialustoihin. Alustat voivat olla valmiita kaupallisia tuotteita tai rajapintojen avulla toteutettuja omia työkaluja. Työn lopputavoitteena ei ole valita yhtä raportointialustaa tiettyyn käyttötapaukseen vaan tarkastella, miten alustat ominaisuuksiltaan soveltuvat eri mittakaavan tiedonkeruu- ja raportointitilanteisiin.

Opinnäytetyön tilaaja on kotkalainen vuonna 2002 perustettu teknologiayritys Ontec Oy. Yrityksen toimialana ovat kone- ja prosessisuunnittelu sekä IT-palvelut ja -konsultointi. Ontec on erikoistunut vaativien nestellogistiikan ohjausjärjestelmien kokonaistoimituksiin. Ne täyttävät sertifiointin Euroopan Unionin mittauslaitedirektiiville (MID) sekä räjähdysvaarallisten alueiden ja tilojen ohjausjärjestelmille (ATEX), kun käsiteltävät aineet ovat syttymis- tai räjähdysherkkiä. Ontecin tuotteet ovat moduulipohjaisia, joiden pohjalta voidaan rakentaa sertifiointin täyttävä ja tarpeisiin skaalattu kokonaisuus ilman järjestelmäintegraatio-ongelmia. (Ontec 2022a.)

Raportointi on jalostettua prosessitietoa. Sen vaatimuksia ovat viiveettömyys ja korkea jalostusaste, mikäli sitä hyödynnetään tehokkaan päätöksenteon tukena. Raportoinnin toteuttaminen vaatii automaatioympäristössä teknisiä ratkaisuja ja niiden yhteensovittamista. Opinnäytetyöraportin kirjallisuusosiossa käsitellään raportointiin liittyvää tiedon olemusta, sen luomia mahdollisuuksia palveluntarjonnan näkökulmasta sekä tulevaisuuden näkymiä tiedon jalostuksen ja laitteiden yhteen nivoutumisesta. Lisäksi kirjallisuusosiossa käsitellään teknisiä ratkaisuja tiedon kulun ja raportoinnin näkökulmista automaatiojärjestelmissä. Luodaan yleiskatsaus järjestelmäintegraatioihin, tiedonsiirron ja arkistoinnin toimintaan ja taustoihin. Opinnäytetyön pääsisältönä on raportointialustojen ominaisuuksien kartoittaminen ja vertailu. Tämän taustoja, terminologiaa ja teknologioita selitetään kirjallisuusosion avulla.

2 TIEDON HIERARKIA JA JÄRJESTELMÄINTEGRAATIOT

Moderni yritysmaailma ja valmistusprosessit asettavat korkeat vaatimukset tiedon hyödyntämiselle sekä saatavuudelle. Yritysmaailman toimintaympäristöt eivät ole enää lineaarisia ja ennustettavia kuten aiemmin. Yrityksien kykyä menestyä määrittävät nopea mukautumiskyky muuttuvaan kilpailuympäristöön sekä kilpailuedun saavuttaminen tehokkaalla päätöksenteolla, jota tuetaan tiedolla ja informaatiolla. Valmistusprosessien pitää vastaavasti tuottaa mahdollisimman korkealaatuisia sekä personoituja tuotteita ja laitoksien seisonta-aikojen tulee olla minimaaliset. Tämä vaatii avointa ja toisiinsa integroitavissa olevaa ohjelmisto- ja verkkoinfrastruktuuria, joka ulottuu mitta- ja toimilaitteilta yritysjohtoon tasolle asti. Tiedon tehokas hyödyntäminen ja järjestelmien saumattomuus eivät ole ainoastaan yritysmaailman sanelema vaatimus selviytymiselle nykyajan kiihtyvässä kilpailussa, vaan ne ovat myös tavoitteita käynnissä olevalle teollistumisen neljännelle aikakaudelle. (Awad & Ghaziri 2003, 25–26; Berge 2005, 3; Dey & Sen 2022, 2; Gobinath 2020, 3; Åkerman 2018, 3.)

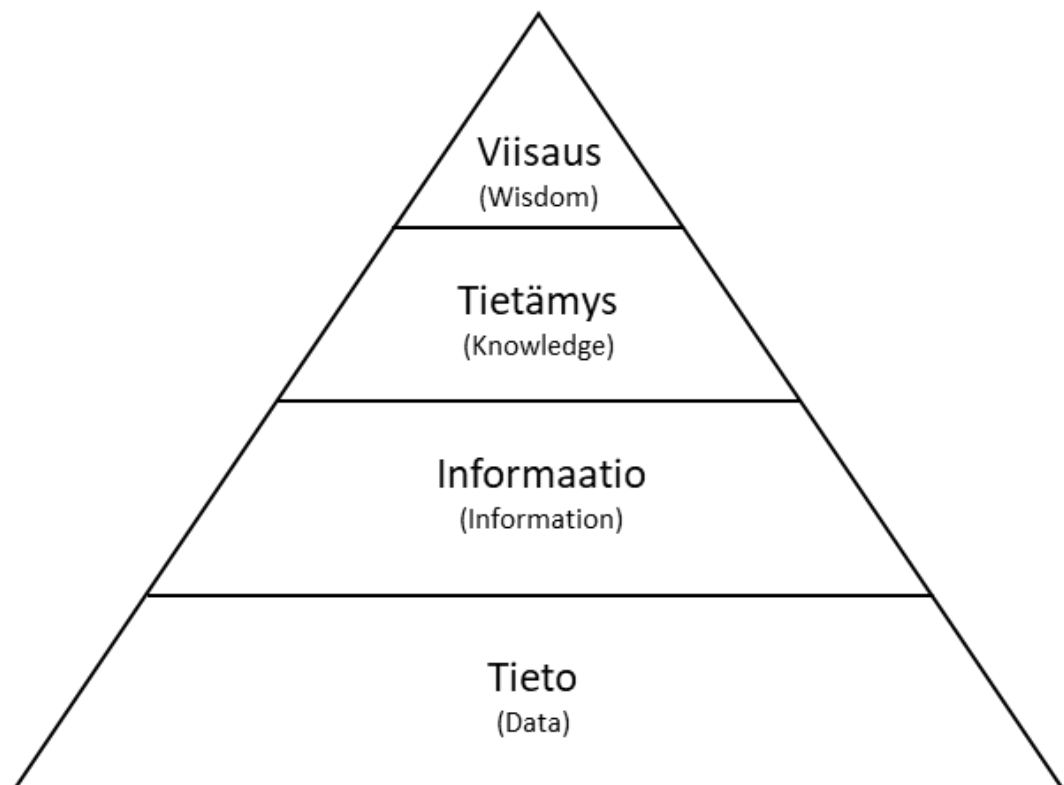
Teollistumisen alkuaikoina prosessilaitosten eri järjestelmät, yksiköt ja koneet toimivat omina saarekkeinaan. Prosessitietoa luettiin mittareista, valotauluista sekä venttiilien asennoista. Yhteistoiminta ja tiedonkeruu perustui ihmisiin ja ihmisten väliseen koordinointiin. Mikroprosessorin julkaiseminen 1970-luvun alussa mahdollisti teknisten laitteiden hyödyntämisen tiedon keräämisessä ja tämä käynnisti samalla tietojenkäsittelyn ensimmäisen aikakauden. Tätä seurasivat 1980-luvun informaation aikakausi sekä 1990-luvun tietämyksen aikakausi. Kukin aikakausista on ollut kehitysaskel kyvyssä jalostaa kerättyä tietoa joko laitteiden suorituskyvyn kasvaessa tai tiedonjalostusmenetelmien kehittyessä. (Awad & Ghaziri 2003, 25–26; Berge 2005, 1)

Tiedon kerääminen on perustoimintaa kaikissa nykypäivän teollisuusyrityksissä, mutta sen valtava määrä ja hajautuminen eri toimijoiden välille hankaloittaa edelleen sen hyödyntämistä. Yrityksillä ei myöskään aina ole tarvittavaa teknistä järjestelmätuntemusta tiedon keräämisen, arkistoinnin ja jalostamisen toteuttamista varten. Järjestelmät ja laitteet halutaan ostaa kokonaistoimituksena ja käyttövalmiina. Tämä mahdollistaa eritasoista yhteistyötä laitetoimittajien ja tilaajayrityksien välillä järjestelmien tiedon keruun osalta. Mainittuun toi-

mituskokonaisuuteen voivat kuulua esimerkiksi kriteerit täyttävän viranomaisraportoinnin automatisointi ja kunnossapitoraportointi, jotka tarjoavat suorat toimintavaihtoehdot heti raja-arvojen ylittyessä. (Ahonen ym. 2017, 16; Kärri ym. 2017, 127–128; Käyhty 2022.)

2.1 Tiedon hierarkia

Puhekielessä tiedon käsitteitä käytetään hyvin löyhästi ja eri jalostusasteet tai niiden sisäiset käsitteet sekoitetaan toisiinsa. Myös tiedon määrällinen kasvu, yhdessä aiemmin mainittujen ongelmien kanssa, aiheuttaa nykyaikana monille haasteita kyvyssä tulkita tietoa ja tilastoja (Bowne-Anderson 2018). Tämän takia tiedeyhteisössä on luotu hierarkiamalli, joka määrittää tiedon eri jalostusasteet, niiden välisiä suhteita ja vakioi terminologiaa tieteellisessä kontekstissa. Kuvassa 1 on esitetty tiedon hierarkian perusmalli, jonka juuret johtavat tilastotieteilijä Ackoffin vuoden 1989 julkaisuun *From Data to Wisdom*. Malli tunnetaan myös nimellä DIKW-hierarkia tai DIKW-pyramidi. Lyhenne tulee englannin kielen sanoista data, information, knowledge ja wisdom.



Kuva 1. Tiedon hierarkian perusmalli (mukaillen Rowley 2006, 164)

Tieto

Hierarkian alimmalla tasolla on data, josta käytetään tämän työn yhteydessä nimitystä tieto. Tiedolle on lukuisia peruskäsitteitä, kuten yksikkö, alkio, mittaus, havainto ja lähde. Tietoa kerätään informaation lähteeksi, mikä tapahtuu mittauksin ilmiöistä tai havainnoin tapahtumista ja asioista. Mittauksen tai havainnoinnin kohteena on aina yksikkö, joka on osa jotakin kokonaisuutta. Yksiköllä on ominaisuuksia tai muuttujia, joita kutsutaan alkioiksi. Alimman tason tieto on siis mittauksin ja havainnoin kerättyjä arvoja ilmiöistä, jotka merkitään yksikön alkioille. (Australian Bureau of Statistics s.a.)

Ilmiöiden ja asioiden arvot voivat olla kvantitatiivisia tai kvalitatiivisia. Kvantitatiivisia eli määrällisiä arvoja ovat esimerkiksi kohteiden paineet tai lämpötilat. Kvalitatiivisia eli laadullisia arvoja ovat esimerkiksi kohteiden nimet. Mittaukseen ja havainnointiin liittyy aina virheen käsite, eli kerätyn arvon ja todellisen arvon välinen ero. (Australian Bureau of Statistics s.a.) Kuvassa 2 on havainnollistettu peruskäsitteitä taulukkomuotoisessa rakenteessa, joka sisältää prosessitietoa lämpötilan mittaus- ja säätöpiiristä.

| Loop | \$Date | \$Time | Mode | TEMPERATURE | OUTPUTPERCENT | SETPOINT e(t) | |
|--------|----------|----------|------|-------------|---------------|---------------|----------|
| TIC-12 | 09/16/21 | 11.46.00 | Auto | 59,99349 | 22,53483 | 60 | 0,00651 |
| TIC-12 | 09/16/21 | 11.46.00 | Auto | 59,99349 | 22,55486 | 60 | 0,00651 |
| TIC-12 | 09/16/21 | 11.46.01 | Auto | 59,99349 | 22,5749 | 60 | 0,00651 |
| TIC-12 | 09/16/21 | 11.46.01 | Auto | 59,99349 | 22,5749 | 60 | 0,00651 |
| TIC-12 | 09/16/21 | 11.46.02 | Auto | 59,99349 | 22,59493 | 60 | -0,04051 |
| TIC-12 | 09/16/21 | 11.46.02 | Auto | 59,99349 | 22,61492 | 60 | -0,04051 |
| TIC-12 | 09/16/21 | 11.46.02 | Auto | 59,99349 | 22,63496 | 60 | 0,00651 |
| TIC-12 | 09/16/21 | 11.46.03 | Auto | 59,99349 | 22,65499 | 60 | 0,00651 |

→ Alkiot

→ Yksikkö

→ Määrällinen muuttuja

→ Laadullinen muuttuja

Kuva 2. Tiedon käsitteet taulukkomuotoisessa rakenteessa (mukailien Australian Bureau of Statistics s.a.)

Tosielämän havainnot ja ilmiöt ovat hyvin usein kvalitatiivisia, mutta prosessiteollisuudessa mittaukset ja ohjausmallit perustuvat kvantitatiiviseen tietoon, jos yksittäisten sumean logiikan sovellutuksien pyrkimyksiä ei huomioida (Ibrahim ym. 2008, 2630). Tiedon alkulähteet ovat joko suoria tai epäsuoria. Suora lähde tarkoittaa, että tieto on kerätty kyseessä olevaa käyttötarkoitusta varten ja epäsuora, että hyödynnettävä tieto ei ole kerätty kyseessä olevaa käyttötarkoitusta varten (Australian Bureau of Statistics s.a.).

Informaatio

Toisin kuin tieto, informaatio on jäsenneiltyä ja käsiteltyä. Informaatio esittää asioiden välisiä suhteita vastaamalla kysymyksiin, kuten kuka, mikä, missä, milloin ja kuinka monta. Informaation muodostamisessa sallitaan kerätyn tiedon muokkaaminen, esimerkiksi ilmiselvät virrehavainnot sallitaan poistettavaksi jalostamisen yhteydessä. Informaatiokäsitteeseen liittyy myös olennaisesti sen esitysmuoto, jonka pitäisi auttaa havainnollistamaan esitettävien asioiden suhteita tai kokoluokkaa. Esitysmuotoina voivat yksinkertaisimmillaan olla tilastollisesti jäsenneiltyä tietoa, kuten trendikäyrät, keskiarvot tai muut mielekkäät tunnusluvut (Ahonen ym. 2017, 19). Awad & Ghaziri (2003, 61) esittävät analogian informaation hyvästä esitysmuodosta vertaamalla sitä puhelimeen. He esittävät, että puhelin ei yksistään varmista merkityksellistä keskustelua, mutta joidenkin puhelimien avulla viesti välittyy selkeämmin.

Faktaan perustuvan päätöksenteon toteuttaminen on helpompaa informaation kuin pelkän tiedon pohjalta. Hyvä informaatio luo sisältöä raporttiin tai välitettävään sanomaan, mutta sen määrä ei kuitenkaan ole laadun tae. Informaation ominaisuudet ovat siis enemmän kvalitatiivisia kuin kvantitatiivisia. Tästä informaation määrästä Ackoff (1989, 3) toteaa, että on yritysjohdon kriittinen tarve ei ole saada suinkaan lisää relevanttia informaatiota, vaan vähemmän epärelevanttia informaatiota.

Tietämys

Tietämys on johdettu informaatiosta, kuten informaatio on johdettu tiedosta. Tietämys vastaa epävarmuuden ja kompleksisuuden asettamiin haasteisiin ja on ollut ihmiskunnan kehityksen kannalta välttämätön osatekijä. Tietämyksen määritelmä riippuu usein asiayhteydestä, mutta ihmisillä se on yleisesti ymmärrystä, joka on saavutettu kokemuksen ja oppimisen kautta. Lyhennettynä puhutaan tietotaidosta (engl. know-how). Tietämyksen ongelmana, että sitä itessään ei voida suoraan siirtää asiayhteydestä toiseen. Awad & Ghaziri (2003, 57) esittävät tästä esimerkin, että sydänkirurgi osaa tietotaidollansa operoida eteis-kammio-äppänen, mutta samalla tietotaidolla hän ei osaa korjata ajoneuvon vaihdelaatikkovikaa.

Informaatioon verrattuna tietämys sisältää havainnointikykyä, taitoa, koulutusta ja kokemusta, jotka ovat kaiken päätöksenteon peruspilareita. Näitä lueteltuja ominaisuuksia käyttäen voidaan informaatiosta jalostaa tietämystä tai ainakin arvioida informaation soveltuvuutta sen pohjaksi. Esimerkiksi prosessisuureiden ja muiden mittauksien perusteella voidaan tehdä kunnossapidollinen johtopäätös toimilaitteen ikääntymisestä. (Ahonen ym. 2017, 19; Awad & Ghaziri 2003, 57, 61–63.) Ackoffin (1989, 4) mukaan tietämys mahdollistaa, että voimme käyttämisen sijasta hallita järjestelmiä ja saada ne toimimaan tehokkaasti.

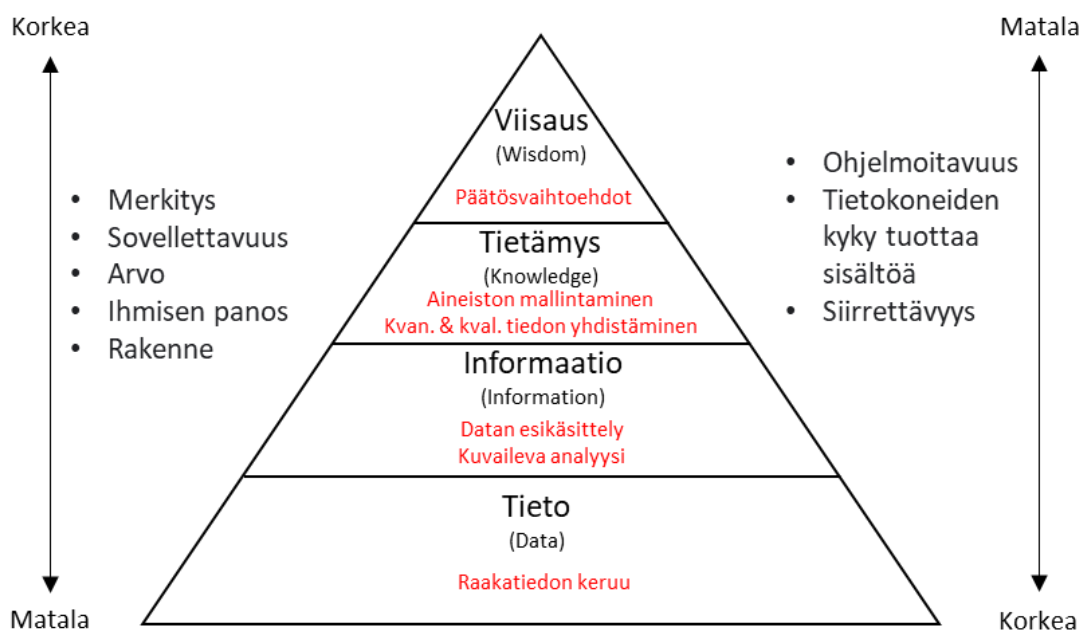
Viisaus

Viisaus on korkein aste kyvyssä käsitellä abstraktiutta. Sitä voidaan kuvailla ihmisillä pitkän asiantuntijauran kokemuksena. Tietämykseen verrattuna viisaus sisältää ominaisuuksia, kuten visio, kyky ennakoida pidemmällä aikavälillä, yhdistellä asioita, muodostaa vaihtoehtoja sekä arvioida niiden hyviä ja huonoja puolia. Viisauteen on liitetty myös subjektiivisia ominaisuuksia, kuten vakaumus ja arvomaailma. Ahonen ym. (2016, 20) esittävät, että esimerkiksi viisautta kunnossapidon kehityksessä on vikaantuneen kohteen tunnistamisen jälkeen arvioida, että olisiko komponentti parempi vaihtaa toisen malliseen, pitäisikö koko kunnossapito-ohjelmaa muutettava vai vaatikko ongelman ratkaiseminen kattavampaa uudelleensuunnittelua. Informaatio, kuten uutinen, vanhenee nopeasti, mutta viisauden elinkaari on huomattavasti pidempi, joskin sekin väistämättä vanhentuu. (Ackoff 1989, 9; Awad & Ghaziri 2003, 64.)

Viisaus on tiedon hierarkian korkein taso. Sen katsotaan olevan ihmisten, muttei vielä koneiden ja laitteiden tavoitettavissa. Viisauden tieteellisen tulkinnan katsotaan olevan haastavaa myös ihmisten osalta, koska kirjallisuudessa sitä käsitellään huomattavasti lyhyemmin ja vaihtelevammin kuin muita hierarkian tasoja (Rowley 2006, 174–176). Tietojärjestelmien ja niihin liittyvän tiedonhallinnan koetaan olevan vielä tietämystasolla, joskin viisaustason saavuttaminen on asetettu käynnissä olevan teollistumisen neljännen aikakauden tavoitteeksi. Tason saavuttaminen vaatii koneilta kognitiivisia kykyjä eli koneoppimista ja tekoälyä (Åkerman 2018, 20).

Hierarkian soveltaminen eri käyttötarkoituksiin

Aiemmin esiteltyä tiedon hierarkian perusmallia on laajennettu sisällyttämällä siihen päätelmiä ja nykyaikaan soveltuvia uusia käsitteitä. Mallia voidaan täten käyttää teoreettisena pohjana moniin eri sovelluksiin ja tieteellisiin tarkasteluihin. (Rowley 2006, 174.) Kuvassa 3 on pyramidin sisälle tuotu tiivistettynä kunkin tason tärkeimmät ominaisuudet. Pyramidin lomaan on tuotu eri käsitteitä ja niiden ominaisuuksia liikuttaessa eri tasojen välillä.



Kuva 3. Tiedon hierarkiamallin laajennus (mukaillen Rowley 2006, 176)

Uusien käsitteiden tuominen malliin on luonut mallista aikaisempaa tulkinnanvaraisemman sekä mustavalkoisemman. Tätä kuvastaa, että Rowley (2006, 176) oli sijoittanut siirrettävyyden pyramidin vasemmalle sivustalle, mutta Awad & Ghaziri (2003, 57) tulkinnan mukaan siirrettävyys heikkenee, mitä korkeammalle hierarkiassa mennään. Lähde Awad & Ghaziri sijoittaa siirrettävyyden siis pyramidin oikealle sivustalle. Täten kuvassa 3 siirrettävyys on sijoitettu myös oikealle, jotta se on linjassa aiemmin esitetyn sydänkirurgin tietotaito-analogian kanssa. Laajennuksestakin havaitaan, että tiedon ja informaation kerääminen ovat pääsääntöisesti teknisiä kysymyksiä, mutta hierarkian ylemmät tasot vaativat laajempaa ymmärrystä kohdeympäristöstä ja asiantuntemusta.

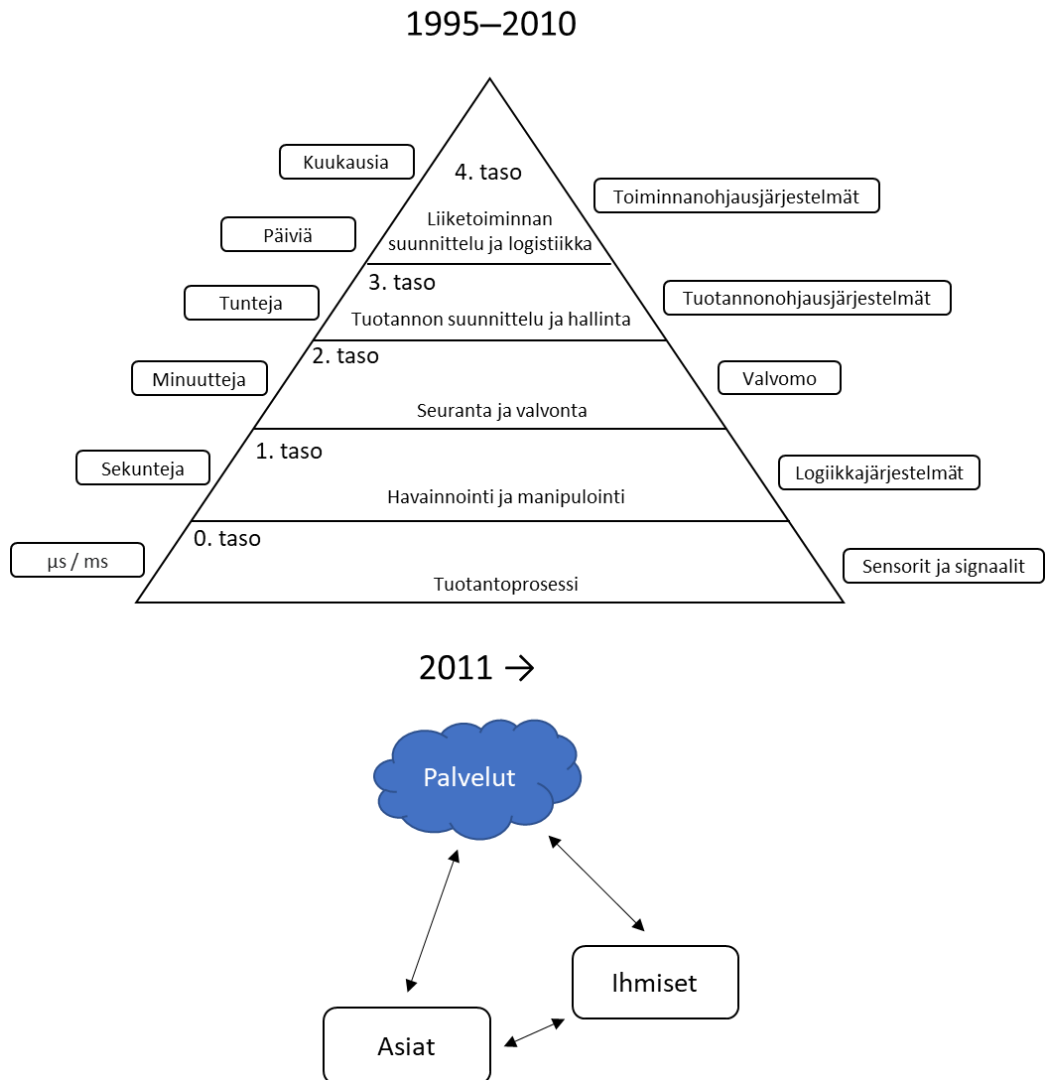
DIKW-hierarkiamallia soveltamalla on luotu malli kuvaamaan yritysten välistä yhteistyötä. Kuten mainittu, yrityksillä ei ole aina tarvittavaa osaamista tiedon keräämiseksi ja jalostamiseksi, joten tämä mahdollistaa eri yhteistyötasoja ti-laajayritysten ja järjestelmätoimittajien välillä. Tämä ei välttämättä aina tarkoita suoraa konsultointia tai erillisten asiantuntijapalvelujen tuottamista, vaan se voidaan myös mieltää tiedonkeruun ja raportoinnin toteuttamisena automaatiojärjestelmään.

Yhteistyötasomallin alimmalla tasolla laitetoimittaja mahdollistaa tiedon keräämisen ja asiakas itse vastaa sen jalostamisesta informaatioksi. Vaativimmalla tasolla palveluntarjoaja vastaa tiedon keruusta, jalostamisesta sekä johtopäätöksistä, tarjoten asiakkaalle toimintavaihtoehtoja. Tätä yhteistyötasoja kuvaa hierarkiaa kutsutaan Datasta liiketoimintaosaamiseen -malliksi, joka esiteltiin teknologiayrityksistä ja tutkimusorganisaatioista koostuvan DIMECC Oy:n raportissa vuonna 2017. Mallissa tiedosta käytetään nimitystä data, joten se on jätetty ennalleen seuraavissa kuvauksissa. (Ahonen ym. 2017, 16; Kärri ym. 2017, 127–128.)

Datasta liiketoimintaosaamiseen -mallin yhteistyötasot ja niissä tarjottavat palvelut ovat seuraavat:

- Data palveluna (engl. Data as a Service). Datan kerääminen ja tallentaminen ovat alimman tason peruspalveluna. Tarjotaan relevantit välineet mittauksien keräämiseen, siirtämiseen ja tallentamiseen.
- Informaatio palveluna (engl. Information as a Service). Kerätystä datasta jalostetaan informaatiota. Toteutetaan datasta visualisointeja ja lasketaan tunnuslukuja. Palveluntarjoaja tarjoaa analysointivälineet, jolla asiakas itse tai palveluntarjoaja tuottaa asiakkaalle valmiit raportit.
- Tietämys palveluna (engl. Knowledge as a Service). Tason saavuttaminen vaatii palveluntarjoajalta käsitystä asiakkaan yrityksestä, tarpeista ja toimintaympäristöstä, jotta informaatiosta voidaan tehdä merkityksellisiä johtopäätöksiä.

Kuvassa 5 esitellään ensin perinteinen ISA 95 -automaatiopyramidi vastaikoineen, järjestelmätasoinen ja niiden päätehtävineen. Sen alapuolella esitellään kyberfyysisen tuotantojärjestelmän mallia, joka kuvaa hierarkian vähyttä, kommunikaation suoraviivaisuutta, tuotannon skaalautuvuutta ja joustavuutta. Ihmiset ja palvelut osallistuvat suoraan toimintaan, jolloin tuotetaan personoituja tuotteita ja osallistutetaan asiakkaat syvälle suunnitteluvaiheisiin.



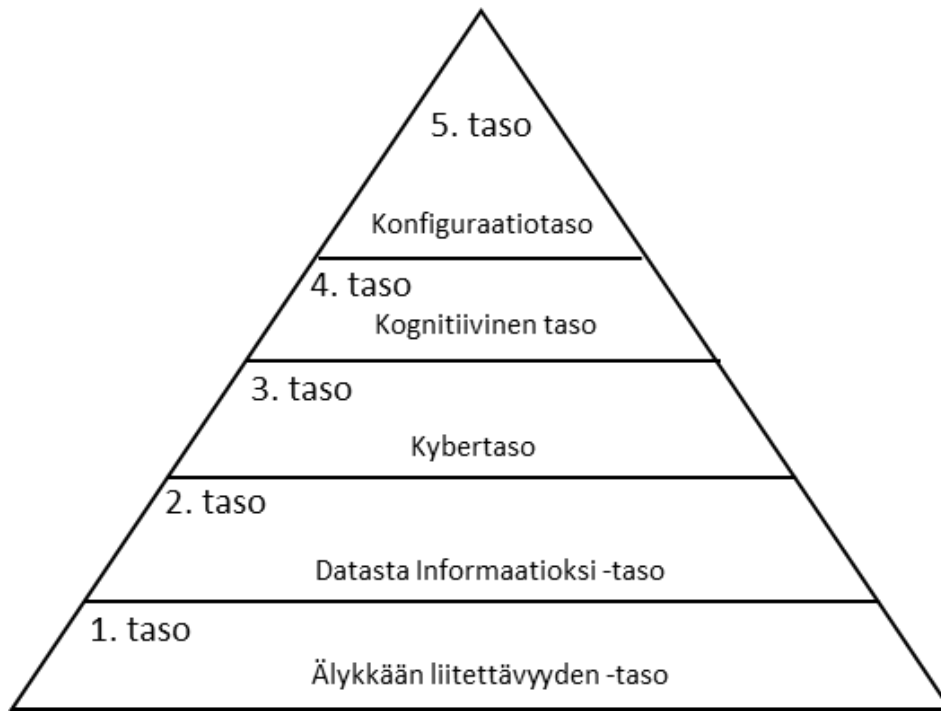
Kuva 5. ISA 95 -automaatiopyramidi ja kyberfyysinen tuotantojärjestelmä (mukaillen Åkerman 2018)

Kyberfyysisestä järjestelmästä käytetään DIKW-hierarkian laajennukseen jäljitettävissä olevaa mallia, jossa on lisäksi vastaavia ominaisuuksia verrattuna Datasta liiketoimintaosaamiseen -malliin. Käytetty 5C-malli on viisiportainen ja lyhenne tulee englanninkielisistä termeistä: connection, conversion, cyber, cognition ja configuration. Mallin neljällä ensimmäisellä tasolla on hyvin läheinen yhteys DIKW-hierarkian tasoihin.

5C-mallin eri kerroksien tehtävät ovat seuraavat:

- Ensimmäinen taso on Älykäs liitettävyys. Tasolla tehdään tiedon keräystä ohjaimien ja antureiden avulla. Tasoon liittyviä termejä ovat Plug and Play, langattomuus ja sensoriverkko.
- Toinen taso on Datasta informaatioksi. Nimensä mukaisesti tasolla jalostetaan ensimmäisellä tasolla kerättyä tietoa informaatioksi. Tasolla kone myös tarkkailee omaa suorituskykyään ja komponenttien kuntoa.
- Kolmas taso on Kybertaso. Tasolla koneet verkottuvat ja yhdistelevät itse keräämäänsä informaatiota muualta hankittuun tietoon ja informaatioon. Näistä vertaillaan yhtäläisyyksiä ja poikkeamia sekä tehdään niihin tarvittaessa korjauksia. Tämän informaation pohjalta jalostetaan tietämystä.
- Neljäs taso on Kognitiivinen-taso. Tällä tasolla tapahtuu vasta ensimmäinen koneen ja ihmisen välinen vuorovaikutus. Kone tuo jalostamiin ehdotuksia ihmisten päätöksenteon tueksi. Tähän koneet tarvitsevat viisautta, joka muodostuu koneoppimisen ja tekoälyn tukemana.
- Viides taso on järjestelmän Itsekonfigurointi. Tasolla järjestelmä itsenäisesti optimoi ja ylläpitää omaa toimintakykyään.

Kuvassa 6 on 5C-mallin pyramidimallinen esitysmuoto, joka vastaa semanttikaltaan DIKW-mallia. (Bagheri ym. 2014, 19; Gobinath 2020, 2–3.)



Kuva 6. Kyberfyysinen järjestelmän 5C-malli (mukaillen Bagheri ym. 2014)

2.2 Järjestelmäintegraatiot ja tiedonsiirto

Järjestelmien yhteentoimivuus tarkoittaa kahden järjestelmän kykyä ymmärtää ja käyttää toistensa toiminnallisuuksia (Chen ym. 2008, Åkerman 2018, 13 mukaan). Tästä yhteentoimivuudesta voidaan käyttää myös sanaa järjestelmänintegraatio, joka tarkoittaa tiedon välittämistä järjestelmien ja sovelluksien välillä. Pelkästään kahden sovelluksen asentaminen samaan tietokoneeseen ei ole varsinaista järjestelmäintegraatiota, jollei ohjelmien välillä ole mitään yhteyttä (Berge 2005, 44, 51). Järjestelmäintegraatiot voidaan jakaa neljään eri teoreettiseen ja hierarkkiseen tasoon. Alhaalta ylöspäin luettuna ne ovat tekninen, syntaktinen, semanttinen ja organisatorinen. Tekninen taso käsittelee järjestelmien laitteisto- ja ohjelmistonäkökohtia. Syntaktinen taso on tietojen siirtoa, jota yleensä hallinnoidaan standardoiduilla viestintäprotokollilla. Semanttisella tasolla tarkastellaan, että siirrettyä tietoa käytetään sovittujen määritelmien mukaisesti. Organisatorinen taso mittaa miten hyvin kokonaiset organisaatiot voivat vaihtaa tietoja. Jokaisen tason saavuttaminen edellyttää alla olevien tasojen onnistunutta toteutusta. Tiedonvaihdon tarpeet vaihtelevat aina käyttöympäristönsä mukaan, vaikka kyseessä olisikin tyypiltään samanlaiset järjestelmät. (Rezaei ym. 2014, Åkerman 2018, 13 mukaan.)

Kuten teollistumisen neljännen aikakauden tavoitteiden asettelu määritti, on automaatioverkoissa tarve järjestelmien, laitteiden ja instrumenttien väliselle kommunikaatiolle. Nykyisellään tiedonvaihto automaatioverkoissa tapahtuu avoimien tai laitevalmistajien omien protokollien ja rajapintojen välityksellä, mutta automaatiojärjestelmiäkin vaivasi pitkään kansainvälisten standardien puuttuminen, kuten yleisesti muutakin informaatioteknologiaa. (Dey & Sen 2022, 98; Berge 2005, 35.) Teknologisen kehityksen alkuaikoina oli yleisesti käytössä ajatusmalli, että omaa markkinaosuutta voidaan vakiinnuttaa ja kasvattaa käyttämällä omia suljettuja menetelmiä ja viestintäprotokollia. Tyypillään tällaisia suljettuja ratkaisuja nimitetään valmistajakohtaisiksi (engl. proprietary). Tämän todettiin itsessään kuitenkin hidastavan laitekannan kehitystä ja olevan yleisestikin epäedullista järjestelmäratkaisujen sekä koko toimialan kokonaiskehitykselle. Myöhemmin havaittiin, että on tärkeää huomioida muutkin automaatiojärjestelmän elinkaarivaiheet kuin laitehankintoihin liittyvät. (Berge 2005, 53, 62; Dey & Sen 2022, 203.) Tässä työssä tarkastellaan järjestelmäintegraatioita vain teknisellä ja syntaktisella tasolla.

2.2.1 Syntaktisen ja teknisen tason järjestelmäintegraatiot

Syntaktinen taso tarkoittaa järjestelmäintegraatioissa tietojen siirtoa eri järjestelmien välillä. Tiedonsiirtomenetelmä automaatiojärjestelmissä alkoi pneumaattisilla ja sähköisillä, jännite- ja virtatasoihin perustuvilla signaaleilla. Tämä ”analogiseksi kenttäväyläksi” kutsuttu tiedonsiirtomenetelmä vaatii isoissa tuotantolaitoksissa hyvin suuria määriä kaapelointia, mikä kasvattaa asennuskustannuksia ja vaikeuttaa vianetsintää. Teknologian kehittyminen loi myöhemmin kyvyn siirtää informaatiota digitaalisessa muodossa, jonka tuomia etuja ovat kaapeloinnin vähäisempi määrä, kytkentöjen yksinkertaisuus sekä tiedonsiirron kaksisuuntaisuus. Kaksisuuntaisuudella tarkoitetaan, että tietoa voidaan vastaanottamisen lisäksi myös lähettää mitta- tai toimilaitteille. Nämä digitaaliset kenttäväylät mahdollistivat entistä paremman ja kustannustehokamman prosessinhallinnan. (Dey & Sen 2022, 203–205.)

Alussa laitevalmistajat loivat omiin käyttötarkoituksiin soveltuvia digitaalisia kenttäväyläratkaisuja. Tämä johti heikkoon integroitavuuteen, kun kyseessä oli valmistajien välisten järjestelmien kommunikaatio. Kehittyi uudelleen auto-

maatiosaarekkeitä, kuten teollistumisen alkuaikoina. Integrointiongelmien suuruus johti lopulta ensimmäisen standardoidun kenttäväyläkonseptin luonnokselle vuonna 1985, jolla luotiin pohjaa avoimuudelle ja standardeille. Samalla valmistajakohtaisten ratkaisujen määrää väheni huomattavasti, mutta markkinoille jäi silti kirjava joukko eri liitännöitä ja viestintäprotokollia, esimerkiksi jo hankittujen järjestelmien muodossa. (Dey & Sen 2022, 203.)

Suurempi muutos teolliseen tiedonsiirtoon tapahtui 2000-luvun taitteessa, kun Ethernet-siirtotekniikka yleistyi teollisuusympäristössä. Tämä muutos auttoi merkittävästi liitännöiden ja kaapeloinnin yhtenäistämässä. Samalla tapahtui askel kohti yleisiä informaatioteknologian verkkostandardeja ja yhteensopivuutta, koska Ethernet-siirtotekniikan kanssa on luonnollista hyödyntää IP-pohjaista verkkotekniikkaa. Ethernet-siirtotekniikka ei ole ratkaissut kaikkia kenttäväyliin liittyviä ongelmia. Automaatiopyramidin kenttälaitetasolla vaaditaan syklistä ja erittäin pieniviiveistä tiedonsiirtoa, jota yleinen Ethernet-siirtotekniikka ei tue. Tämän vaatimuksen takia digitaaliset kenttäväylät poikkeavat edelleen yleisistä toimistoverkoista. (Goller 2019, 1; Jasperneite ym. 2017, 18; Åkerman 2018, 16.)

Kenttäväylät ovat tällä hetkellä standardisoitu kahdella IEC-standardilla. Standardien tarkoituksena on edistää avoimuutta ja valmistajariippumattomuutta teollisuuden automaatioverkoissa. IEC 61158 -standardi on yleiskuvaus kenttäväylyistä. IEC 61784 -standardia voidaan kuvailla soveltamisohjeeksi.

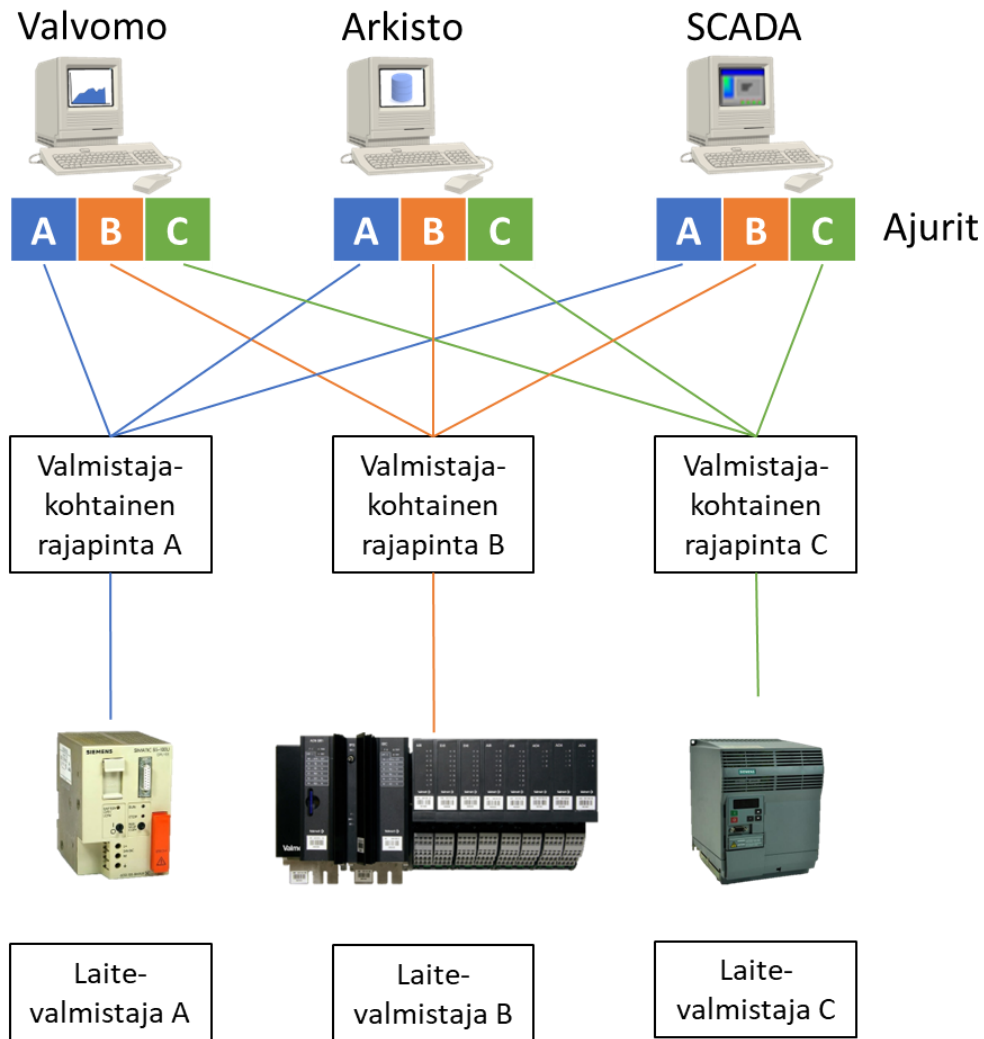
Tekninen integraatiotaso käsittelee järjestelmien laitteisto- ja ohjelmistonäkökohtia. Pääosiltaan ohjelmistot ovat aina valmistajien tekijänoikeussuojattuja tuotteita, joiden toimintaperiaatteet ja toteutustapa eivät ole avoimia esimerkiksi lähdekoodin muodossa. Tästä lähtökohdasta ohjelmat eivät myöskään vaihtaisi tietoja keskenään, ellei jotakin osaa ohjelman toteutuksesta olisi tarkoituksenmukaisesti avattu ja dokumentoitu ulkopuolisten käyttöön (Berge 2005, 44).

Teknisen tason integrointiin päävaihtoehtoja ovat laiteajuri (engl. driver) ja rajapinta (engl. interface). Ajuri tarkoittaa yhden laitteen ja yhden sovelluksen yhdistämiseen tarkoitettua alemman tason sovellusta, jolla käytetään suoraan

laitteen ominaisuuksia. Rajapinta on yhtä lailla suunniteltu ohjelmistojen ja laitteiden väliseen tiedonvaihtoon, mutta siinä kohde tarjoaa toimintojaan käytettäväksi yksinkertaisin komennoin. Esimerkiksi laitteen tapauksessa, rajapinnan alla toimii laitevalmistajan itse teettämä ajuri, joka vastaa laitteen ominaisuuksien suuremmasta käytöstä. (Berge 2005, 45; Difference between an API... 2015).

Terminologia rajapintojen suhteen on kirjava: ohjelmointirajapinta, ohjelmistorajapinta, tiedonsiirtorajapinta, API jne. Sanastokeskus (2014) määrittelee rajapinnan siten, että kahden laitteen välillä sen voi muodostaa fyysinen liitäntä tai tiedonsiirron määritelmät, ohjelman ja käyttäjän välillä sen muodostaa käyttöliittymä ja kahden ohjelman välillä sen muodostaa ohjelmointirajapinta eli API (engl. application program interface). Tämän perusteella rajapinta jakautuu terminologisesti kahdeksi, mutta kummastakin voi hyvin käyttää yksilöimättömää termiä rajapinta.

Kuten järjestelmäintegraatioissa yleisestikin, ohjelmointintegraatioiden toteuttamista hidasti standardien puuttuminen. Mikäli laitevalmistajalla ylipäättään oli toteutus ajureiden ja rajapintojen tukemiseksi, niiden käyttämiseksi tarvittiin hyvin usein tapauskohtaisesti räätälöityjä ajuri- tai rajapintatoteutuksia. Dokumentaatiot olivat usein puutteellisia, tai valmistajat saattoivat vaatia hyvin kalliita maksuja dokumentaatiosta ja rajapintojen avaamisesta. Toimintaa varjosti siis vahvasti aikakauden ajatus markkinaosuuksien tavoittelusta, kuten kenttäväyläpuolellakin. Tämä ei ainoastaan hankaloittanut järjestelmien rakentamista, vaan yhtä lailla ylläpitoa ja päivittämistä. (Berge 2005, 35.) Kuvassa 7 on havainnollistettu ennen standardointia olevaa tilannetta, jossa eri laitevalmistajat joutuivat luomaan laitekohtaisia ajureita valmistajakohtaisia rajapintoja vasten.



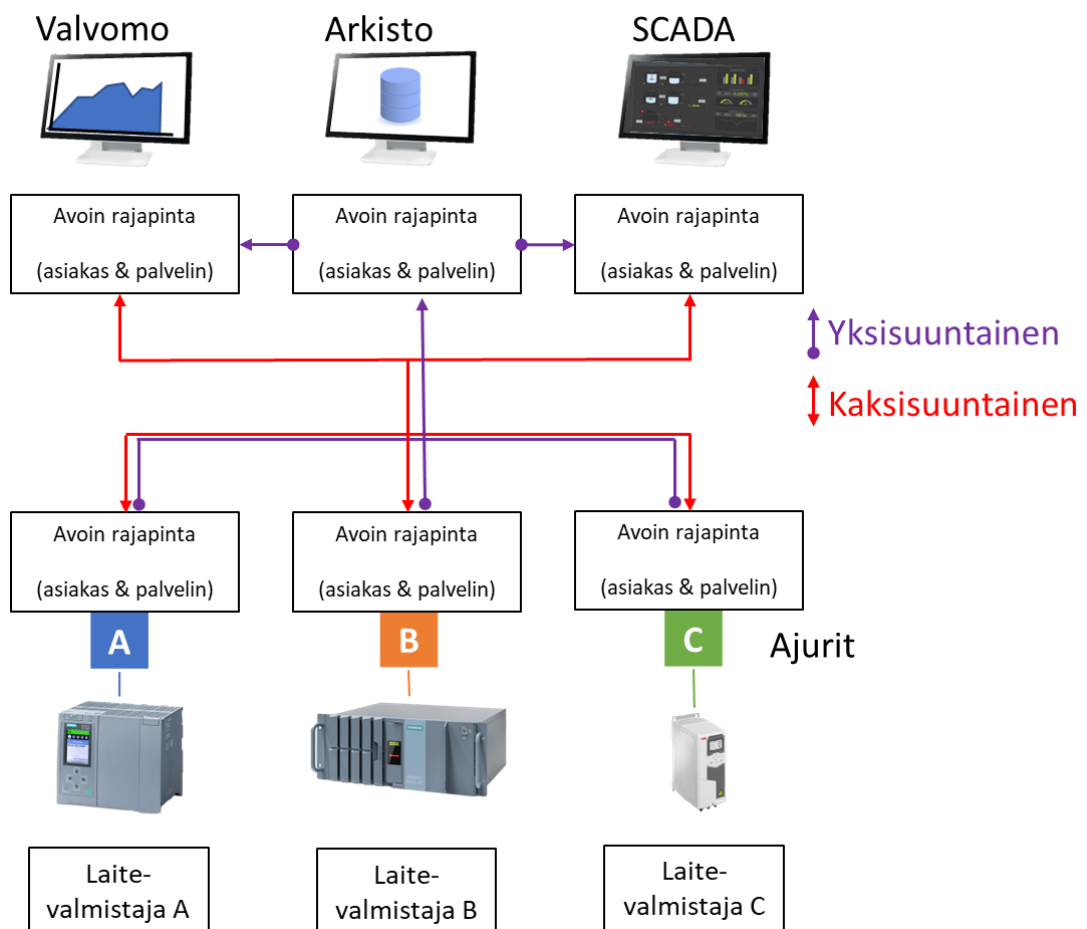
Kuva 7. Standardoimattomat ja valmistajakohtaiset rajapinnat (mukaillen Natri 2019, 32)

Avoimuuden havaittiin lopulta parantavan yleistilannetta sekä laitevalmistajien että tuotantolaitoksien näkökulmasta, koska avoimet rajapinnat helpottavat eri laitevalmistajien tuotteiden yhteiskäyttöä, mahdollistavat ns. vaakasuuntaisen integraation eri laitteiden välillä sekä helpottavat ylläpitoa. Avointen rajapintojen toteuttaminen ei vaadi kuitenkaan sellaista avoimuutta, että ohjelmien toteuttamisesta tulisi epäedullista eri valmistajille. Rajapinta avaa vain korkeamman tason rakenteen ohjelmasta kolmansille osapuolille. Ohjelman ydintoiminnot eivät paljastu rajapinnan käyttäjälle. Puhutaan ns. mustan laatikon -mallista. (Berge 2005, 46; Natri 2019, 33; Pitkänen 2016, 10, 18.)

Rajapinnat voivat olla yksi tai kaksisuuntaisia, avoimia, suljettuja tai puoliavoimia. Yksisuuntaisessa rajapinnassa toinen osapuoli lähettää kutsuja, joihin toinen vastaa tiedolla tai toimenpiteillä. Kaksisuuntaisessa rajapinnassa molemmat osapuolet voivat tehdä näitä kutsuja. Kaksisuuntaiset rajapinnat ovat

usein yksityisiä, esimerkiksi tarkkaan valvottuja yritysten välisiä tiedonvaihtokanavia. Avoin rajapinta on kaikkien vapaasti käytettävissä oleva rajapinta. Se ei kuitenkaan poikkea toiminnaltaan em. mustan laatikon mallista ja nykyään julkaisijat asettavat käytölle yleensä hyvin tarkat rajoitteet. Suljetut rajapinnat ovat yritysten ja sen tuotteiden sisäiseen tiedonsiirtoon tarkoitettuja. Puoliavoimet ovat näiden kahden väliltä, jolloin rajapinnassa on esimerkiksi erikseen lisensoitavia osia. (Pitkänen 2016, 7–8.)

Kuvassa 8 esitetään rajapintatoteutusta kuvitteellisessa ympäristössä, jossa arkistointijärjestelmä kerää tietoa yksisuuntaisella rajapintatoteutuksella ja muissa toiminnoissa hyödynnetään kaksisuuntaisia rajapintoja. Kuvassa näkyy myös aiemmin viitattu vaakasuuntainen integraatio, jonka avulla ylätason järjestelmät ovat integroitavissa toisiinsa ilman varsinaista lisätyötä.



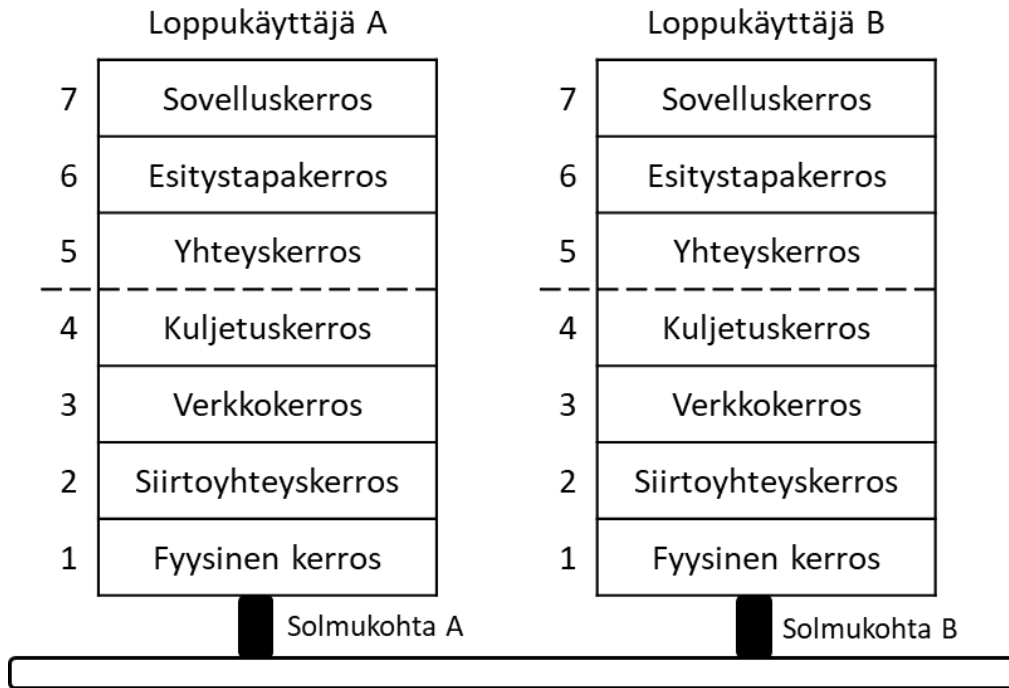
Kuva 8. Avoimet rajapinnat (mukaillen Natri 2019, 33)

Nykyiset automaatiojärjestelmät ovat erittäin riippuvaisia eri ohjelmistojen ominaisuuksista ja toiminnoista. Lisäksi niiltä vaaditaan ympäristöriippumattomuutta, uudelleenkäytettävyyttä ja siirrettävyyttä. Täten ohjelmistopuolta on standardisoitu kahdella IEC-standardilla. IEC 61512 on suunnattu erityisesti eräprosessien ohjausjärjestelmille, mutta soveltuu myös jatkuvien prosessien ohjausjärjestelmille. IEC 62264 määrittelee toiminnanohjaus- ja tuotannonohjausjärjestelmien välistä integraatiota rajapintojen avulla. (Dey & Sen 2022, 86)

2.2.2 Järjestelmäintegraatiotasojen yhdistyminen OSI-mallin avulla

Järjestelmien välinen tiedonsiirtotapahtuma rakentuu aiemmin esiteltyjen teknisen- ja syntaktisen tasojen ominaisuuksien yhteen nivoutumiselle. Tätä kokonaisuutta varten on luotu seitsenportainen Open Systems Interconnection (OSI) -referenssimalli, joka on kirjattu IEC 7498-1 -standardiin. Se kuvaa pino-tyyppisesti, miten eri tiedonsiirron vaiheita yhdistetään toisiinsa ja mikä niiden toimintajärjestys on. Täydellinen seitsenportainen malli kuvaa tiedonsiirron vaiheet loppukäyttäjältä toiselle fyysisen siirtotien kautta. Malli itsessään on yleistetty eikä kuvaa erityisiä laitteita tai menetelmiä. Varsinaiset tosimaailman toteutukset, kuten TCP/IP tai teollisuuden kenttäväylät, eivät myöskään noudata täysin OSI-mallia, mutta niiden toimintaa voidaan silti kuvata mallin avulla. (Thompson 2008, 27, 29–30.)

Kuvassa 9 on täydellinen seitsenportainen OSI-malli, jossa kuvataan kahden loppukäyttäjän välistä yhteyttä. Teoreettisesti, kun loppukäyttäjä A haluaa olla yhteydessä loppukäyttäjään B:hen, on A:lta lähtevän viestin kuljettava ensin pino alas solmukohtaan A ja takaisin ylös solmukohdalta B.



Kuva 9. OSI-malli (mukaillen Thompson 2008, 30)

OSI-mallin eri kerroksien tehtävät ja toimintaperiaatteet ovat seuraavat:

- Fyysinen kerros tarkoittaa menetelmiä, joilla digitaalinen tieto liikkuu ns. välittäjässä (media), kuten kuparijohto tai valokaapeli. Digitaalinen tieto on bittejä ja ne usein kuvataan sarjana symboleita 1 ja 0. Välittäjä ja käytettävät liittimet ovat keskeisimmät fyysisen kerroksen ominaisuudet.
- Siirtoyhteyskerros toimii välittäjänä fyysisen kerroksen ja verkkokerroksen välillä. Fyysisen kerroksen suuntaan se varmistaa, että lähtevä liikenne ei mene päällekkäin muiden lähetyksien kanssa ja vastaanotettu liikenne on virhevapaata ja otettu täydellisenä vastaan. Tarvittaessa se pyytää jonkin tiedon uudelleen lähetystä. Siirtoyhteyskerros paketoit fyysiseltä kerrokselta saapuvan bittitiedon kehyksiin, jotka lopuksi siirtään ylemmän verkkokerroksen käyttöön. Siirtoyhteyskerros liittää kehukseen mukaan solmukohdalle yksilöllisen fyysisen osoitteen, jota käytetään ohjaamaan paketteja ylemmillä tasoilla.

- Verkkokerroksen tehtävänä on reitittää kehyksiin pakattua tietoa siirtoyhteys- ja kuljetuskerroksien välillä. Tämä voi tapahtua monisäikeisissäkin verkoissa, kuten isot toimistotalot tai palvelinkeskukset. Tätä varten verkkokerros luo oman loogisen osoitejärjestelmänsä ja sitoo siirtoyhteyskerroksen fyysiset osoitteet näihin loogisiin osoitteisiin. Looginen osoite liitetään mukaan siirtoyhteyskerrokselta saatuun kehykseen, jonka jälkeen kokonaisuutta nimitetään paketiksi. Verkkokerroksessa käytetty tekniikka on yksi ratkaisevimmista tekijöistä yleisen integroituuden kannalta.
- Kuljetuskerroksessa liikkuvia tiedon kokonaisuuksia kutsutaan segmenteiksi. Kuljetuskerros vastaa verkkokerrokselta saatujen pakettien purkamisesta ja muuntamisesta muotoon, jossa ylemmät kerrokset voivat hyödyntää sitä. Vastaavasti se myös pakkaa ylemmiltä kerroksilta saadun tiedon biteiksi segmenttien kehysosiin. Kerroksen protokollia on kahta tyyppiä: yhteydellisiä ja yhteydettömiä. Yhteydellinen protokolla varmistaa, että yhteys on luotu ylemmän kerroksen vastaanottajan kanssa ennen tiedon lähettämistä. Yhteydetön protokolla ei varmista yhteyttä, vaan lähettää sen kaikissa tapauksissa. Kuljetuskerros on myös taso, jossa OSI-malli yleisesti jaetaan ylempiin ja alempiin kerroksiin.
- Yhteyskerros huolehtii sovellus- ja kuljetuskerroksen välisen yhteyden todentamisesta, salauksesta, muodostuksesta, sulkemisesta ja uudelleen käynnistämisestä katkostatilanteissa.
- Esitystapakerros toimii yhteys- ja sovelluskerroksen välissä. Se vastaa, että kommunikointi tapahtuu käytännössä yhteisellä kielellä ja halutussa muodossa. Se muuntaa asioita formaateista toiseen standardikoodauksien mukaisesti.

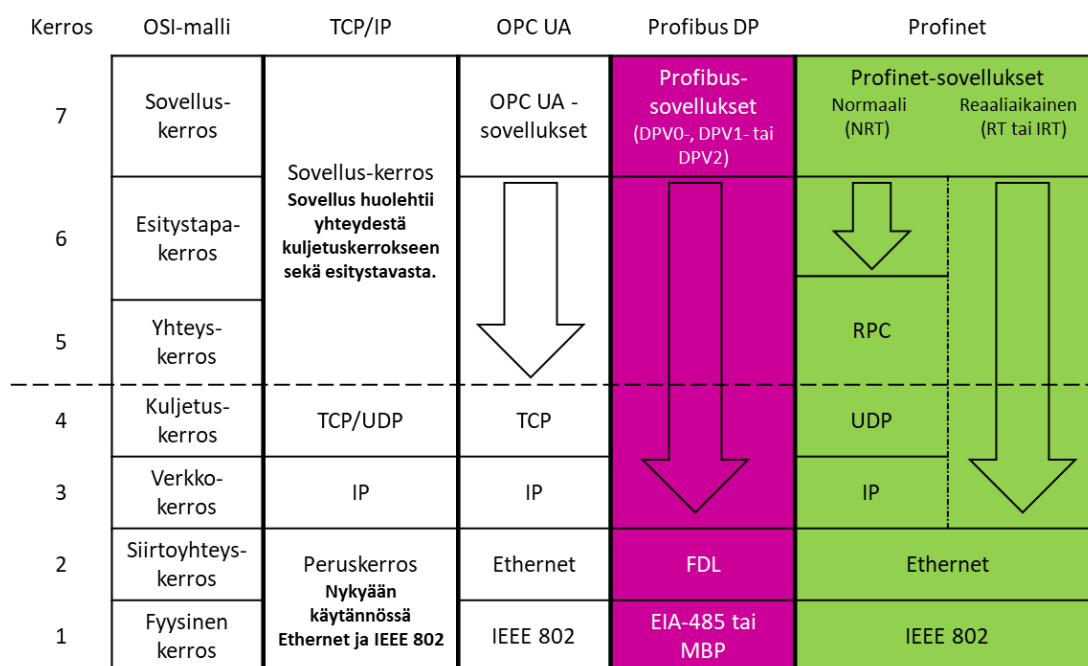
- Ylin kerros on sovelluskerros. Loppukäyttäjät näkevät yleensä vain tämän kerroksen käyttäessään järjestelmiä. Sovellus voi olla mikä tahansa prosessinohjausjärjestelmästä tietokantasovellukseen. Sovellus viestii verkon suuntaan jollakin sovelluskerrokselle tyypillisellä protokollalla, kuten HTTP, IMAP tai TELNET. (Fall & Stevens 2012, 9–10; Thompson 2008, 30–31.)

Voidaan havaita, että teknisen ja syntaktisen integraatitasojen väliraja on varsin häilyvä OSI-mallin pohjalta, mutta se riippuu myös hyvin paljon käyttöympäristöstä. Kenttäväylissä syntaktinen taso voi ulottua teoriassa OSI-mallin ylempiinkin kerroksiin, kun internet-ympäristössä se rajoittuu OSI-mallin alempiin kerroksiin. Kuvassa 10 on verrattu oleellisia päämalleja keskenään. Internet-mallista on yhdistetty useita kerroksia ja tasoilla 1 ja 2 käytettäviin tekniikoihin ei oteta kantaa. Kenttäväyliä varten on oma karsittu mallinsa. Sitä käytetään myös laajojen ohjausjärjestelmien eri osien väliseen tiedonsiirtoon, jolloin siitä käytetään myös nimitystä Enhanced Performance Architecture (EPA) -malli (Dey & Sen 2022, 182).

| Kerros | Tyyppi | OSI-malli | Internet-malli | Kenttäväylä-malli |
|--------|-----------|--------------------|----------------|--------------------|
| 7 | Data | Sovelluskerros | Sovelluskerros | Sovelluskerros |
| 6 | Data | Esitystapakerros | | |
| 5 | Data | Yhteyskerros | | |
| 4 | Segmentti | Kuljetuskerros | Kuljetuskerros | |
| 3 | Paketti | Verkkokerros | Verkkokerros | |
| 2 | Kehys | Siirtoyhteyskerros | Peruskerros | Siirtoyhteyskerros |
| 1 | Bitti | Fyysinen kerros | | Fyysinen kerros |

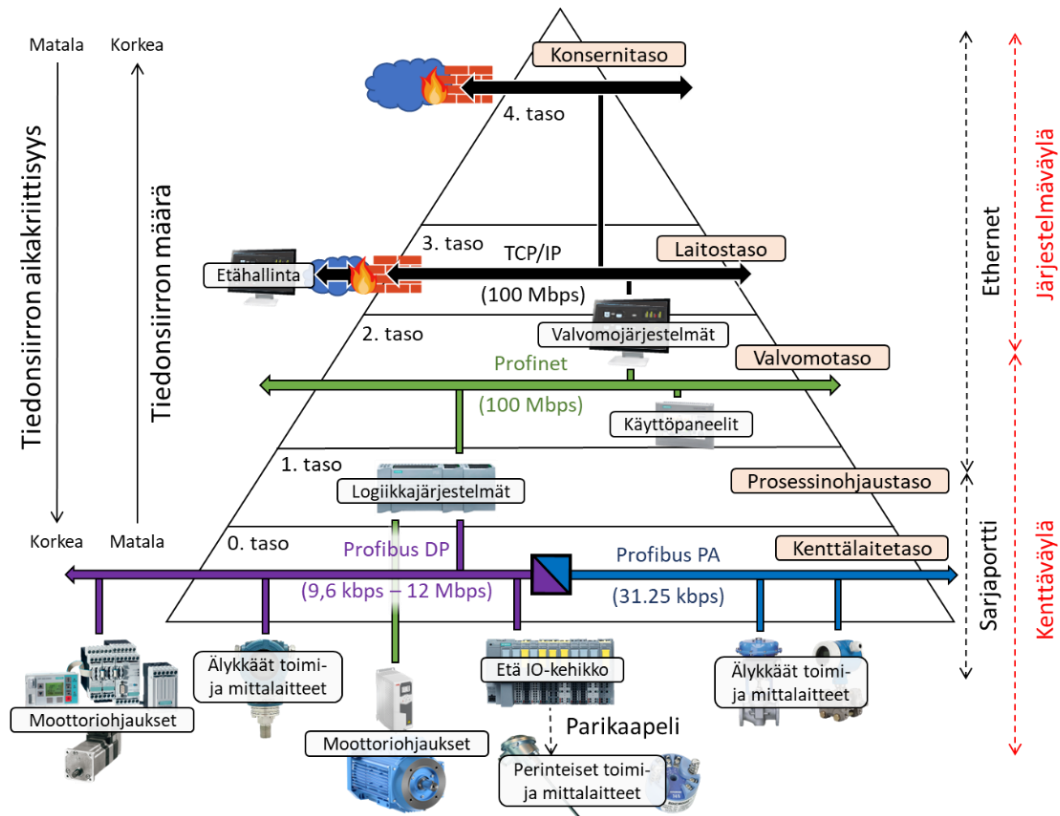
Kuva 10. OSI-, Internet- ja Kenttäväylä-mallit (mukaillen Goller 2019, 2)

Pinotyypin kerrosrakenteen takia voimme siis käyttää mitä tahansa tekniikkaa, kunhan se on yhteensopiva OSI-mallissa ylemmän ja alemman kerroksen kanssa. Kuvassa 11 on havainnollistettu, miten teollisuusympäristössä yleisesti käytetyt tiedonsiirtomenetelmät sijoittuvat OSI-mallissa. Profibus DP edustaa perinteistä kenttäväyläratkaisua. Sen fyysisen kerroksen liittimet ovat RS485-tyyliset sarjaliikenneväyläliittimet. Kehittyneempi Profinet-kenttäväyläratkaisu sisältää ominaisuuksia kenttäväylä- sekä Internet-mallista. Sen siirtoyhteys- ja verkkokerrokset Ethernet- ja IP-pohjaisina mahdollistavat kenttäväylien verkkoliikenteen ajamisen samassa verkossa toimistoverkkojen TCP-protokolla kanssa, mutta se tukee myös kenttälaitteiden kanssa vaadittua syklistä ja pieniviiveistä tiedonsiirtoa. OPC UA on avoin tiedonsiirtostandardi, jota käytetään järjestelmäintegraatioihin automaatiopyramidin ylätasolla. OPC UA sijoittuu TCP/IP-protokollapinon päälle ja on täten siirtoystävällinen menetelmä. (Natri 2019, 35; Profinet University s.a.)



Kuva 11. OSI-malli ja yleisiä teollisuusympäristön tiedonsiirtomenetelmiä (mukaillen Goller 2019, 2; OPC UA s.a.; Profinet University s.a.)

Työssä jo aiemmin viitattiin, että pyrkimys siirtoyhteys- ja verkkokerroksien yhtenäistämiseksi Ethernet- ja IP-pohjaisiksi on hyvin ratkaiseva järjestelmäintegraatioiden kannalta. Täten pystytään yhtenäistämään kaapelointia sekä pakettien reitittämiseen tarvittava laitteisto, vaikka siitä ylemmät protokollat poik-



Kuva 12. Väylätasot ja tiedonsiirron periaatteet automaatiopyramidissa (mukailten Dey & Sen 2022, 223)

2.2.3 Yleisiä tiedonsiirtomenetelmiä

Prosessiteollisuudessa tai Internetin eri palveluiden välisessä tiedonsiirrossa käytetään monia yleisiä tiedonsiirtotekniikoita. Raportoinnin kontekstissa prosessiteollisuudessa esiintyvät tiedonsiirtomenetelmät sijoittuvat prosessinohjaustasolta ylöspäin automaatiopyramidissa. Tiedonsiirtomäärät ovat suurempia kuin laitetasolla, mutta vastaavasti tiedonsiirron aikakriittisyys on matala. Tiedonsiirtomenetelmät voivat käsittää useamman tason OSI-mallista, jolloin siitä voidaan puhua arkkitehtuurina tai standardina.

Integrointistandardit OPC Data Access ja OPC Unified Architecture

Open Platform Communications (OPC) -standardi on laitetoimittajien, loppukäyttäjien ja ohjelmistokehittäjien integrointitavoitteiden lopputuotos. Sillä on ollut keskeinen rooli teollisuusautomaatiosovellusten tiedonvaihdon standardoinnissa sekä aiemmin esitettyjen integraatiohaasteiden ratkaisussa. Toimijoiden tavoitteena oli saada käyttöön yhteisiä menetelmiä valvomo- ja prosessinohjaustasojen asiakas-palvelin- ja palvelin-palvelin-yhteysarkkitehtuureille.

Vaatimuksena oli yhteinen rajapinta prosessitiedon, hälytysten ja tapahtumien seurannalle sekä arkistoinnille. Standardin ensimmäinen versio julkaistiin 1990-luvun puolivälissä ja sen ylläpitämiseksi perustettiin voittoa tavoittelematon säätiö. Standardin kattavuutta ja yhtenäisyyttä tuki, että säätiön jäseniksi liittyivät lähes kaikki teollisuusautomaation tärkeimmät toimijat. Tuotteiden yhteensopivuutta valvotaan sertifiointilla, jossa testaaminen perustuu järjestelmätoimittajien omiin sekä kolmannen osapuolen toteuttamaan testaukseen, jotka OPC-säätiön lopuksi varmentaa. (Damm ym. 2009, 1–8; Dey & Sen 2022, 86.)

OPC Data Access (DA) on standardin ensimmäinen versio, joka julkaistiin vuonna 1996. Siitä käytetään myös nimitystä OPC Classic. Ensimmäisen versio on ominaisuuksiltaan hyvin rajattu, mikä toisaalta vauhditti sen julkaisua. Sen sovelluskerroksen protokolla on sidottu Microsoft Windowsin ohjelmistokomponenttimalleihin Component Object Model (COM) ja Distributed Component Object Model (DCOM). Sitoutumisella vähennettiin ohjelmistojen määrittelyyn ja toteuttamiseen kuluva aikaa, vaikka se rajoittikin ohjelmistokehittäjien vaihtoehtoja. Alkuperäisen OPC DA:n sisältämät tekniikat ovat nykymitapuulla vanhentuneita, koska COM- ja DCOM-ohjelmistokomponentin käyttö sitoo sovellukset Microsoft Windows käyttöjärjestelmään ja tekniikka ei täytä nykyaikaisia tietoturva-vaatimuksia. Myös yhteyskerroksessa käytetty Remote Procedure Call (RPC) -tekniikka aiheuttaa vaikeuksia toteuttaa turvallisia etäyhteyksiä palomuurien lävitse. (Damm ym. 2009, 3–4; Natri 2019, 34.) Kuvassa 13 on esitelty OPC DA:n pinorakennetta OSI-mallin rinnalla. OPC DA rakentuu TCP- tai UDP-protokollan päälle, mikä tarkoittaa yhteensopivuutta Ethernet-siirtotekniikan kanssa.

| Kerros | OSI-malli | OPC DA |
|--------|--------------------|-------------------------------|
| 7 | Sovelluskerros | OPC DA -rajapinta COM/DCOM |
| 6 | Esitystapakerros | ↓ |
| 5 | Yhteyskerros | RPC |
| 4 | Kuljetuskerros | TCP/UDP |
| 3 | Verkkokerros | IP |
| 2 | Siirtoyhteyskerros | Ethernet |
| 1 | Fyysinen kerros | IEEE 802 |

Kuva 13. OPC DA ja OSI-malli (mukaihen OPC CLASSIC s.a.)

OPC DA:n sovelluskerroksen rajapinta mahdollistaa prosessidataa sisältävien muuttujien valvonnan, kirjoittamisen ja lukemisen. Valvonta on suora käänös lähdemateriaalista ja sillä tarkoitetaan päivitysvälien hallintaa. Kirjoittamisella ja lukemisella tarkoitetaan varsinaista muuttujien käsittelyä. Tietoina välittyvät varsinainen prosessimuuttujan arvo, aikaleima sekä erillinen verkkoyhteyden laatumuuttuja. (Damm ym. 2009, 4–5.)

OPC DA:n rinnalle on luotu useita laajennuksia. Niillä pyritään lisäämään toimintoja, jotka jäivät alussa OPC DA:n ulkopuolelle. OPC Alarm & Events (A&E) mahdollistaa tapahtumailmoitusten ja hälytysten vastaanottamisen. OPC Historical Data Access (HDA) mahdollistaa prosessitiedon arkistojen lukemisen ja päivittämisen. OPC XML-DA oli ensimmäinen yritys luopua rajoitteiksi tulleista COM-, DCOM- ja RPC-tekniikoista. Siinä yhteys- ja sovelluskerrokset hyödyntävät SOAP-rajapinta-arkkitehtuuria, jossa sovelluskerroksessa käytetään HTTP-protokollaa ja esitystapakerroksessa XML-formaatin viestejä. Mainittujen lisäksi on lukuisia muita laajennuksia, kuten OPC Security, OPC Complex Data, OPC Batch, OPC Data eXchange ja OPC Commands. (Damm ym. 2009, 5–8.)

OPC Unified Architecture (UA) -standardin synty rakentuu OPC DA:n rajoitteiden ja yleisten integroimisvaatimuksien pohjalle. OPC UA on täysin uuden sukupolven tiedonsiirtomenetelmä, jolla tavoitellaan alustariippumattomuutta, tukea kehittyneempien tiedonmallinnusmenetelmien tiedonsiirrolle yritystason järjestelmissä, tiedonsiirtoa hajautettujen ohjausjärjestelmien välillä sekä jo OPC DA:sta tuttua prosessitiedonsiirtoa. OPC UA on kirjattu IEC 62541 -standardiksi. Käytännössä OPC UA on modernisoitu OPC DA, johon on sisällytetty OPC DA:han tehdyt laajennukset. Teknisenä päätavoitteena suunnittelussa oli luopua COM-pohjaisesta sovelluskerroksesta ja tehdä tiedonvaihdosta reititettävää palomuurien lävitse. OPC DA-XML hahmotteli jo tätä tavoitetta, mutta oli suorituskyvyltään liian heikko. Se myös sisälsi valmistajien välisiä yhteensopivuusongelmia sovellus- ja esitystapakerroksessa, minkä modernisointia sillä juuri tavoiteltiin. (Damm ym. 2009, 8–11.)

OPC UA:n tuki yritystason tietojärjestelmille on myös kehittyneempi verrattuna OPC DA:n ominaisuuksiin. OPC DA:n laajennukset, kuten Complex Data, eivät tarjoa tukea kuin hyvin pelkistetylle tiedolle, kuten mitatun muuttujan arvon tai mittalaitteen pelkistetty yleistyypin. OPC UA mahdollistaa itsessään jo huomattavasti monipuolisemman tiedonvälityksen ja mallinnuksen, kuten mittalaitteen tarkan sensorityypin välittämisen oheistietona. Tämä mahdollistaa huomattavasti monipuolisemman tiedon jalostamisen. (Damm ym. 2009, 19–20.)

OPC UA käyttää vastaavaa asiakas-palvelin-yhteysarkkitehtuuria kuten OPC DA. Tiedonsiirto tapahtuu joko OPC-standardissa määritellyn binäärimuotoisella tai SOAP-rajapinta-arkkitehtuuriin perustuvalla tiedonsiirrolla. Tiedonsiirtotavan valinta riippuu käytettävästä yhteysvälistä. Binäärimuotoinen tiedonsiirto on huomattavasti suorituskykyisempää ja soveltuu kahden OPC-järjestelmän välille. SOAP-rajapinta-arkkitehtuuriin perustuvalla tiedonsiirrolla voidaan keskustella minkä tahansa arkkitehtuuria tukevan sovellutuksen kanssa. (Damm ym. 2009, 13–14, 192–194, 198.) Kuvassa 14 on OPC UA:n pinorakenne OSI-mallin rinnalla. Modifioitu TCP-protokolla OPC UA TCP liittyy määritettyyn binäärimuotoiseen tiedonsiirtoon.

| Kerros | OSI-malli | OPC UA Binary | OPC UA SOAP |
|--------|-------------------------|----------------------|--------------------------------|
| 7 | Sovellus- kerros | OPC UA -rajapinta | OPC UA -rajapinta & HTTP |
| 6 | Esitystapa- kerros | OPC UA -binääri | XML |
| 5 | Yhteys- kerros | OPC UA TCP | |
| 4 | Kuljetus- kerros | TCP | TCP |
| 3 | Verkko- kerros | IP | IP |
| 2 | Siirtoyhteys- kerros | Ethernet | Ethernet |
| 1 | Fyysinen kerros | IEEE 802 | IEEE 802 |

Kuva 14. OPC UA ja OSI-malli (mukaillen Damm ym. 2009, 192, 198–200; OPC UA s.a.)

Selkokieiset tiedonsiirtoformaatit

Tiedonsiirto ei välttämättä aina voi tapahtua binäärimuodossa, vaikkakin tiedon esitystapamuunnokset eli serialisointi voitaisiin standardisoida binäärimuotoiseksi. Joissakin käyttötapauksissa tiedonsiirtoformaatin pitää olla kompromissi tiedon tehokkaan koneluettavuuden ja ihmisluettavuuden välillä. Tällöin käytetään ns. selkokiekisiä tiedonsiirtoformaatteja. Menetelmiä on useita, mutta yleensä tiedonsiirtotapa ja käyttöympäristön ominaisuudet määrittävät käytettävän formaatin. Yleisimmät käytössä olevat formaatit ovat CSV, XML ja JSON. Niitä voidaan käyttää konkreettisina tiedostoina tai OSI-mallin esitystapakerroksen tietomuotona. (Manoochehri 2013, 14, 22.)

Kappaleen esimerkeissä käytettävän tieto on poimittu laboratoriotyön mittauksista, jotka liittyvät Kaakkois-Suomen ammattikorkeakoulun mittaus- ja säätötekniikka -kurssiin. Mittauksissa esiintyy lämpötilan mittaus- ja säätöpiiri TIC-12 sekä siihen liittyviä arvoja. Alkuperäinen mittaustieto on CSV-formaatissa, josta sitä on muokattu XML- ja JSON-formaattien esimerkkeihin.

Comma-separated value (CSV) -formaatissa alkiot ja muuttujat erotellaan nimensä mukaisesti pilkulla, mutta nykymääritelmässä se voi olla mikä tahansa soveltuva merkki, joka ei aiheuta virheellistä tiedon jäsentelyä. Kuvassa 15 yläpuolella on pilkulla eroteltua mittaustietoa CSV-formaatissa ja alapuolella sama tieto esitettynä taulukkomuodossa. Ensimmäinen rivi CSV-formaatissa sisältää alkiot ja loput rivit alkioihin liittyvät muuttujat.

```
Loop, $Date, $Time, Mode, TEMPERATURE, OUTPUTPERCENT, SETPOINT, e(t)
TIC-12, 2021-09-16, 11:46:00, Auto, 59.99349, 22.53483, 60, 0.00651
TIC-12, 2021-09-16, 11:46:00, Auto, 59.99349, 22.55486, 60, 0.00651
TIC-12, 2021-09-16, 11:46:01, Auto, 59.99349, 22.5749, 60, 0.00651
TIC-12, 2021-09-16, 11:46:01, Auto, 59.99349, 22.5749, 60, 0.00651
TIC-12, 2021-09-16, 11:46:02, Auto, 59.99349, 22.59493, 60, 0.00651
TIC-12, 2021-09-16, 11:46:02, Auto, 59.99349, 22.61492, 60, 0.00651
TIC-12, 2021-09-16, 11:46:02, Auto, 59.99349, 22.63496, 60, 0.00651
TIC-12, 2021-09-16, 11:46:03, Auto, 59.99349, 22.65499, 60, 0.00651
```

| Loop | \$Date | \$Time | Mode | TEMPERATURE | OUTPUTPERCENT | SETPOINT | e(t) |
|--------|------------|----------|------|-------------|---------------|----------|---------|
| TIC-12 | 2021-09-16 | 11:46:00 | Auto | 59.99349 | 22.53483 | 60 | 0.00651 |
| TIC-12 | 2021-09-16 | 11:46:00 | Auto | 59.99349 | 22.55486 | 60 | 0.00651 |
| TIC-12 | 2021-09-16 | 11:46:01 | Auto | 59.99349 | 22.5749 | 60 | 0.00651 |
| TIC-12 | 2021-09-16 | 11:46:01 | Auto | 59.99349 | 22.5749 | 60 | 0.00651 |
| TIC-12 | 2021-09-16 | 11:46:02 | Auto | 59.99349 | 22.59493 | 60 | 0.00651 |
| TIC-12 | 2021-09-16 | 11:46:02 | Auto | 59.99349 | 22.61492 | 60 | 0.00651 |
| TIC-12 | 2021-09-16 | 11:46:02 | Auto | 59.99349 | 22.63496 | 60 | 0.00651 |
| TIC-12 | 2021-09-16 | 11:46:03 | Auto | 59.99349 | 22.65499 | 60 | 0.00651 |

Kuva 15. Prosessitietoa CSV-formaatissa ja taulukkomuodossa

CSV-formaatti on hyvä esittämään rivimäistä ja yksitasoista historiatietoa. Sen rakenne on periaatteessa muuttumaton, mikä tekee siitä luotettavan valinnan integraatiotilanteisiin. Se ei sovellu tilanteisiin, joissa tarkasteltava tai mitattava kokonaisuus on hyvin laaja tai monitahoinen, koska jokainen alkio täytyy olla eritelty, vaikka sen merkitys olisi triviaalinen. Manoochehri (2013, 17) käyttää tästä esimerkkinä poliittisten puolueiden kannatuslukuja, jotka ovat eriteltyinä maantieteellisesti. Jokainen puolue pitää olla yksilöitynä alkioiksi, vaikka se esiintyisi vain yhdellä maantieteellisellä alueella ja sillä olisi kannattajia hyvin marginaalisesti. Jos kuvassa 15 esiintyviä alkioita tarkastellaan tältä pohjalta, huomataan, että TEMPERATURE ei ole hyvä alkioityypiksi. Mikäli mittaustietoa olisi myös ohjauspiiristä, jossa mitattava suure olisi paine, pitäisi alkoiden määrää kasvattaa heti yhdellä. Formaatti ei myöskään tue mahdollisuutta, että sen mukana voitaisiin välittää kyseiseen tapaukseen liittyvää oheistietoa. Tällainen hyödyllinen oheistieto voisi olla esimerkiksi käytetty väli-merkki. (Manoochehri 2013, 16–18.)

Extensible Markup Language (XML) -formaattissa tietosisältö rakennetaan elementtien sisään. Elementit voivat jakautua vielä useampiin alaelementteihin. Elementit erotetaan toisistaan alku- ja päättömerkinnällä. (Carter 2018, 29–35.) Syntaksia on esitelty kuvassa 16, jossa laboratoriotyön mittauksien ensimmäinen rivi on muunneltu XML-formaattiin. Vasemmalla kuvassa on esitetty perusrakenne ja oikealla, miten rakenteessa voitaisiin hyödyntää alaelementtejä.

Esimerkkirivi:

| Loop | \$Date | \$Time | Mode | TEMPERATURE | OUTPUTPERCENT | SETPOINT | e(t) |
|--------|------------|----------|------|-------------|---------------|----------|---------|
| TIC-12 | 2021-09-16 | 11:46:00 | Auto | 59.99349 | 22.53483 | 60 | 0.00651 |

Perusrakenne:

```
<Measurements>
  <Loop>"TIC-12"</Loop>
  <$Date>"2021-09-16"</$Date>
  <$Time>"11:46:00"</$Time>
  <Mode>"Auto"</Mode>
  <TEMPERATURE>"59.99349"</TEMPERATURE>
  <OUTPUTPERCENT>"22.53483"</OUTPUTPERCENT>
  <SETPOINT>"60"</SETPOINT>
  <e(t)>"0.00651"</e(t)>
</Measurements>
```

Alaelementteihin jako:

```
<Measurements>
  <Loop>"TIC-12"</Loop>
  <$Date>
    <year>"2021"</year>
    <month>"09"</month>
    <day>"16"</day>
  </$Date>
  <$Time>
    <hour>"11"</hour>
    <minutes>"46"</minutes>
    <seconds>"00"</seconds>
  </$Time>
  <Mode>"Auto"</Mode>
  <TEMPERATURE>"59.99349"</TEMPERATURE>
  <OUTPUTPERCENT>"22.53483"</OUTPUTPERCENT>
  <SETPOINT>"60"</SETPOINT>
  <e(t)>"0.00651"</e(t)>
</Measurements>
```

Kuva 16. Taulukkomuotoinen rivi yhtenä XML-formaatin elementtinä

Elementit alkumerkintä voi sisältää myös muuttujalle hyödyllisiä attribuutteja. Kuvassa 17 havainnollistettuna piirin nimi ja aikaleima on sisällytetty pääelementin attribuutiksi. (Carter 2018, 29–35.)

```
<Measurements Loop="TIC-12" $Date="2021-09-16" $Time="11:46:00">
  <Mode>"Auto"</Mode>
  <TEMPERATURE>"59.99349"</TEMPERATURE>
  <OUTPUTPERCENT>"22.53483"</OUTPUTPERCENT>
  <SETPOINT>"60"</SETPOINT>
  <e(t)>"0.00651"</e(t)>
</Measurements>
```

Kuva 17. XML-formaatin elementti attribuuteilla

XML-formaatin etu CSV-formaattiin on sen sisäkkäinen rakenne ja alkioden joustavampi käyttö attribuuttien avulla. Formaatti tukee myös oheistiedon välittämistä. Yleisin välitettävä oheistieto on formaattiin liittyvä Schema-standardin mukaisuus, joka tarkoittaa tyyliä, miten tieto on jäsenneltynä kyseiseen tiedostoon. (Manoochehri 2013, 17–18.)

JavaScript Object Notation (JSON) -formaatti muistuttaa hyvin paljon XML-formaattia ja on myös sisäkkäisiin rakenteisiin perustuva formaatti. JSON-formaatin etuna on tiedon ohjelmallisen jäsentelyn tehokkuus verrattuna XML-formaattiin. Kuvassa 18 on laboratoriotyön mittauksien ensimmäinen rivi muunnettuna JSON-formaattiin. Formaatisissa elementit erotellaan pilkulla. Kuvan 18 oikeanpuoleisessa osassa havainnollistetaan, miten alaelementit avataan ja suljetaan aaltosulkeilla.

Esimerkkirivi:

| Loop | \$Date | \$Time | Mode | TEMPERATURE | OUTPUTPERCENT | SETPOINT | e(t) |
|--------|------------|----------|------|-------------|---------------|----------|---------|
| TIC-12 | 2021-09-16 | 11:46:00 | Auto | 59.99349 | 22.53483 | 60 | 0.00651 |

| | |
|---|--|
| <p>Perusrakenne:</p> <pre>"Measurements": { "Loop": "TIC-12", "\$Date": "2021-09-16", "\$Time": "11:46:00", "Mode": "Auto", "TEMPERATURE": 59.99349, "OUTPUTPERCENT": 22.53483, "SETPOINT": 60, "e(t)": 0.00651 }</pre> | <p>Alaelementteihin jako:</p> <pre>"Measurements": { "Loop": "TIC-12", "\$Date": { "year": 2021, "month": 9, "day": 16 }, "\$Time": { "hour": 11, "minutes": 46, "seconds": 0 }, "Mode": "Auto", "TEMPERATURE": 59.99349, "OUTPUTPERCENT": 22.53483, "SETPOINT": 60, "e(t)": 0.00651 }</pre> |
|---|--|

Kuva 18. Taulukkomuotoinen rivi yhtenä JSON-formaatin elementtinä

JSON-formaatti ei tue attribuutteja, mutta sen suurin etu on muuttujien tietotyyppien yksilöiminen numeroksi, merkkijonoksi, Boolean-arvoksi tai tyhjäksi. Tietotyytit yksilöidään eri merkintätapoja käyttäen. Merkkijonossa käytetään lainausmerkkejä ja numero esiintyy ilman niitä. Boolean-arvo on joko true tai false. Tyhjää tietoa merkitään sanalla null. Tiedot voidaan ryhmitellä myös taulukoiksi, sulkemalla ne hakasulkeiden sisään. Tämä nopeuttaa huomattavasti tietojen ohjelmallista käsittelyä. Kuvassa 19 on havainnollistettu null-arvon, Boolean-arvon sekä taulukon käyttöä, ottamatta kantaa olisiko esitetty tapa kovin käytännöllinen. Piirin ajotahoa kuvaava Mode olisi esimerkin kannalta ennalta sovittu, jossa true tarkoittaa automaattista ajotahoa ja false manuaalista ajotahoa. Mittaustarkkuutta on kasvatettu tukemaan millisekunteja, mutta sen arvo on tyhjä. Varsinaiset mittaukset koostuvat kahdesta taulukosta, joista ensimmäisestä voisi lukea alkion ja toisesta muuttujan. (Carter 2018, 181–194; Manoochehri 2013, 18–19.)

```

"Measurements": {
  "Loop": "TIC-12",
  "$Date": {
    "year": 2021,
    "month": 9,
    "day": 16
  },
  "$Time": {
    "hour": 11,
    "minutes": 46,
    "seconds": 0,
    "msec": null
  },
  "Mode": true,
  "Items": ["SETPOINT", "TEMPERATURE", "OUTPUTPERCENT", "e(t)"],
  "Values": [60, 59.99349, 22.53483, 0.00651]
}

```

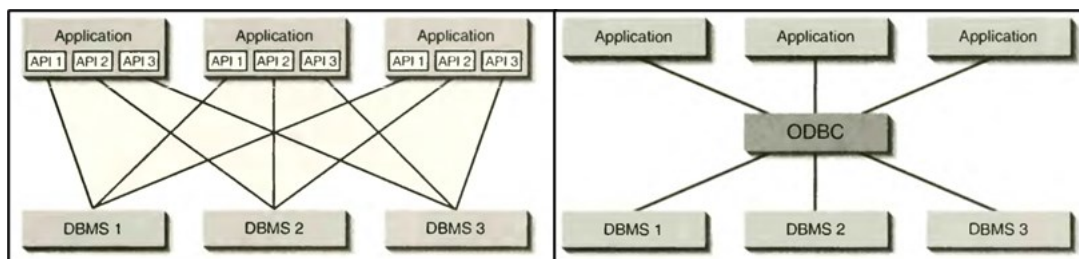
Kuva 19. JSON-formaatti hyödyntäen eri tietotyypppejä

XML- ja JSON-formaatin erot muodostuvat JSON-formaatin tehokkaasta luetavuudesta ohjelmallisesti ja XML-formaatin monipuolisesta tavasta välittää tietoja attribuuttien avulla. CSV-formaatin erona XML- ja JSON-formaattiin on vähäisempi merkkien määrä, pienempi laskentateho käsiteltäessä ja yksinkertainen rakenne.

Edellä mainittujen tiedonesitysformaattien lisäksi löytyy vielä lukuisia muita, kuten YAML, TOML, WDDX jne. Kaikkien formaattien taustalla on sama ihmisluettavuuden lähtökohta ja niiden keskinäisessä vertailussa on helppo havaita tämä ja muita yhtäläisyyksiä. Eri formaattien taustalla on kuitenkin aina tarve painottaa eri osa-alueiden tärkeyttä, kuten koneellista käsittelyn nopeutta, standardisoitavuutta, tukea tietorakenteille tai soveltuvuutta ohjelmointikielille.

Ohjelmointirajapinta Open Database Connectivity

Open Database Connectivity (ODBC) on Microsoftin vuonna 1992 julkaisema ohjelmistorajapinta tietokantojen käsittelyyn. ODBC-rajapinnan tavoitteena oli luoda alustariippumaton ja yhteinen rajapinta, jota voidaan hyödyntää useilla ohjelmointikielillä ja tietokannanhallintajärjestelmästä riippumatta. Rajapinta on kehitetty siis ratkaisemaan ongelman, joka esiteltiin järjestelmäintegraatiokappaleen yhteiset rajapinnat osiossa, mutta se keskittyy vain tietokantojen käsittelyyn. (Geiger 1995, 3–6, 100, 461.) Kuvassa 20 Geiger havainnollistaa (1995, 20–21) jo vuonna 1995 valmistajakohtaisten rajapintojen ongelmaa.

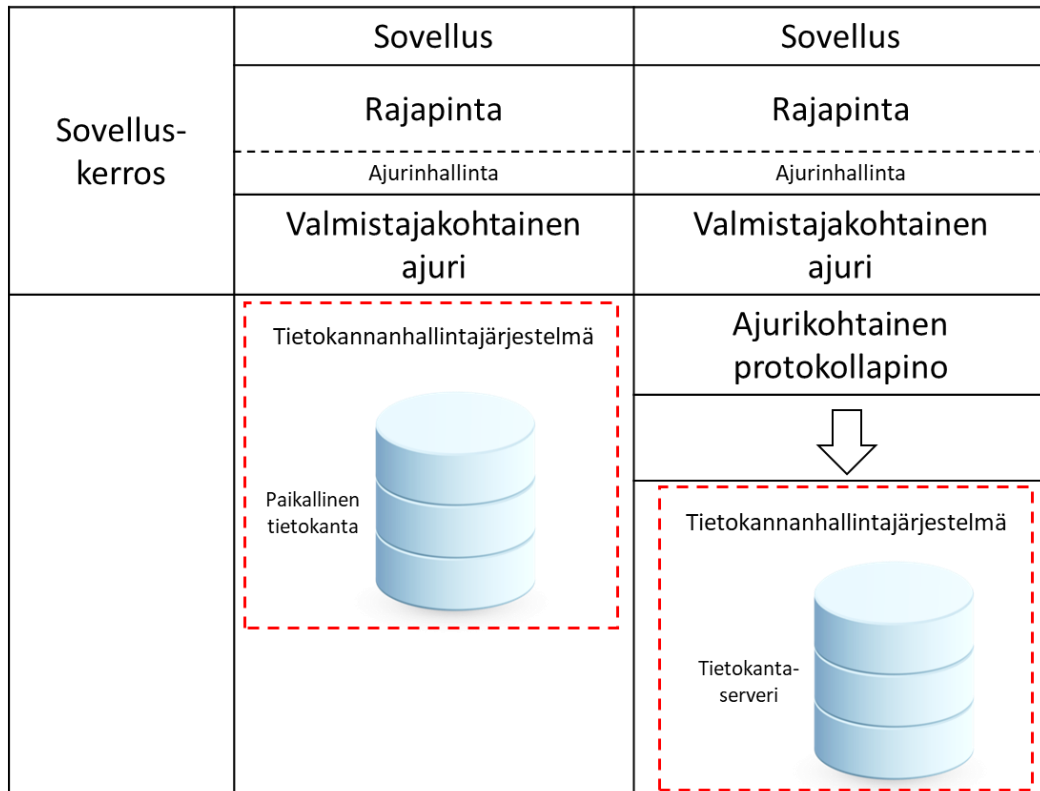


Kuva 20. Valmistajasidonnaisuuden ongelma rajapinnoissa (Geiger 1995, 20–21)

ODBC-rajapinta perustuu SQL Access Groupin (SAG) kehittämään Call Level Interface (CLI) -rajapintaan, joka kehitettiin myös 1990-luvulla vastaamaan käytännössä samaan tarpeeseen. CLI-rajapinta on myös edelleen käytössä ja on määritelmiltäänkin yhtenevä ODBC-rajapinnan kanssa, mutta on suunniteltu hieman eri käyttötilanteisiin. (Geiger 1995, 3–6.)

ODBC-rajapintaa voidaan hyödyntää lukuisilla ohjelmointikielillä, kuten JAVA, R, Python tai C. Rajapinnan kanssa käytettävät komennot perustuvat SQL-ky-selykieleen. Rajapintaan perustuvia lisäosia löytyy myös monista tietomassojen käsittelyyn soveltuvista ohjelmistoista, kuten Microsoft Excel, Microsoft Access, STATA tai SPSS. Kaikilta isoimmilta tietokantojen ohjelmistokehittäjiltä löytyvät ajurit ODBC-rajapinnan hyödyntämiseksi. (Geiger 1995, 16.)

ODBC-rajapinnan toimintaperiaate on vastaava kuin muissakin rajapinnoissa. Rajapinnan alla oleva ajuri määrää yhteyden tietokannanhallintajärjestelmään riippumatta siitä, tapahtuuko se paikallisesti vai verkkoyhteyden ylitse. ODBC-rajapinnan yksi lähtökohdistista on, että ajurien ja verkkomäärityksien kompleksisuus piilotetaan ns. tietolähdenimen (engl. data source name) taakse ja rajapinnan ajurinhallinta huolehtii sopivan valmistajakohtaisen ajurin valinnasta. Protokollapinossa hyödynnetään automaattisesti ajurin vaatimia tiedonsiirto-protokollia, kuten TCP/IP:tä. (Geiger 1995, 85, 102, 123–128.) Kuvassa 21 esitellään ODBC-rajapinnan toimintaa aiempien kuvien OSI-mallin pinorakenteen tyyliin mukailtuna. Vasemmalla kuvassa on havainnollistettu paikallista tietokantayhteyttä ja oikealla verkkoyhteyden ylitse toimivaa yhteyttä.



Kuva 21. ODBC-arkkitehtuuri (mukaillen Geiger 1995, 102)

2.3 Prosessitiedon arkistointi

Automaatiojärjestelmissä kerätty prosessitieto arkistoidaan yleisesti johonkin tietokantaan. Tietoa tallentavaa ja hallinnoivaa ohjelmaa kutsutaan tietokannanhallintajärjestelmäksi tai tietokantamoottoriksi (Thompson 2008, 37). Yleisin tietokantatyyppejä on 1970-luvulta peräisin oleva relaatiomalliin perustuva tietokanta. Siinä tietoa esitetään järjestettynä joukkoina ja ryhmiteltyinä tauluiksi eli relaatioiksi. Jos tauluja on useita, niiden välillä vallitsee jokin looginen yhteys. Vaihtoehtoisia tietokantatyyppejä ovat puutyylinen hierarkiamalli tai asioiden välisiä suhteita kuvaava verkkomalli. (SQL ja relaatiotietokannat s.a.; Tietokantojen perusteet 2019.)

Tämän kappaleen esimerkeissä käytetty tietokanta sisältää WinCC Unified -valvomosovelluksen arkistoimaa tietoa, joka on opinnäytetyötä varten luodusta yksinkertaisesta valvomokäyttöliittymästä. Valvomokäyttöliittymän tarkempi kuvaus on myöhemmin kappaleessa neljä.

Tietokannan taulu on nimensä mukaisesti taulukoitua tietoa. Relaatiotietokantojen yhteydessä käytetty terminologia poikkeaa hieman tiedonhierarkian yhteydessä esitellyistä termeistä. Alkioita kutsutaan attribuuteiksi, muuttujan arvo on tallennettu soluun ja riviä voidaan kutsua myös tietueeksi. Yksi taulun attribuuteista nimetään taulun pääavaimeksi. Sen arvo on yksilöllinen ja täten kukin tietue voidaan yksilöidä. Kuvassa 22 on esitelty nämä taulun rakenneosat sekä taulun pääavain. Pääavaimeksi on määritetty pk_TimeStamp:n ja pk_fk_Id:n yhdistelmä eli aikaleiman ja Id-tunnus sidottuna toisiinsa. Yhdistelmää käytetään, koska kuten kuvastakin voidaan huomata, että Id-tunnuksilla ei voida yksilöidä yhtä riviä. Samoin aikaleima voi olla kahdelle Id-tunnukselle identtinen, vaikka todennäköisyys olisikin hyvin pieni.

| pk_TimeStamp | pk_fk_Id | Quality | Value | Attribuutit |
|--------------------|----------|---------|-------|--------------|
| 132973349080571485 | 1 | 192 | 34.0 | |
| 132973349088686451 | 2 | 192 | 31.0 | Solu |
| 132973349188074327 | 2 | 192 | 20.0 | |
| 132973349263255743 | 1 | 192 | 23.0 | Rivi, tietue |
| 132973349309994864 | 2 | 192 | 36.0 | |
| 132973349941180327 | 1 | 192 | 37.0 | |
| 132973349948271800 | 2 | 192 | 21.0 | |
| 132973350798129630 | 1 | 192 | 24.0 | |
| 132973350805132040 | 2 | 192 | 35.0 | |
| 132973351035594310 | 1 | 192 | 11.0 | |

↓
Sarake

Pääavain:

PRIMARY KEY(pk_TimeStamp, pk_fk_Id)

Kuva 22. Taulun rakenneosat ja pääavaimen määrittely

Tietoja tallennetaan yleensä useampaan tauluun, koska muuten yksittäisestä taulusta tulisi liian massiivinen. Ongelmaa voi verrata CSV-formaatin yhteydessä esitettyihin esimerkkeihin. Useamman taulun välisiä tietoja voidaan yhdistellä, kun taulujen välille luodaan looginen yhteys vierasavaimen avulla. Tietokannan suunnittelijan pitää huomioida, että tämä yhteys on looginen ja yksilöllinen. (SQL ja relaatiotietokannat s.a.) Kuvassa 23 on kaksi taulua, jotka

ovat yhdistettävissä toisiinsa fk_LogID- ja ObjectID-attribuuteilla. Yhdistämällä kahden taulun tietoja, voidaan esimerkiksi tutkia mihin tiedostoon ylemmässä taulussa esiintyvien prosessimuuttujien tiedot tallentuvat.

Table: LoggingTag

| pk_Key | TagSystemId | TagObjectId | TagElementId | LoggingTagId | Version | fk_LogSystemId | fk_LogId |
|--------|-------------|-------------|--------------|--------------|-----------|----------------|----------|
| Filter | Filter | Filter | Filter | Filter | Filter | Filter | Filter |
| 1 | 1 | 1 | 114 | 1 | 687865857 | 0 | 1 |
| 2 | 2 | 1 | 120 | 1 | 687865857 | 0 | 1 |

Ylemmän taulun vierasavain: FOREIGN KEY(fk_LogId) REFERENCES iseg_storage(ObjectId)

Table: iseg_storage

| pk_StorageId | SystemId | ObjectId | Version | Name | ConfiguredName | Location |
|--------------|----------|----------|---------|-----------------------|-------------------------|---|
| Filter | Filter | Filter | Filter | Filter | Filter | Filter |
| 1 | 1 | 1 | 116 | 0 HMI_RT_1-SIM_TLG116 | HMI_RT_1::Data log_test | C:\ProgramData\SCADAProjects\HMI_RT_... |

Kuva 23. Tietojen yhdistäminen relaatiotietokannassa

Tietokannanhallintajärjestelmän tehtävä on toimia välikätenä tietokantaan tallennetun tiedon ja loppukäyttäjän välillä. Käyttäjä hyödyntää tietokantaan tekemällä siihen joko tiedon manipulaatioita tai rakenteen määrittelyjä. Näiden tekemiseen käytetään relaatiotietokantojen yhteydessä Structured Query Language (SQL) -kieltä. Kaikki yleisimmät hallintajärjestelmät, kuten DB2, Microsoft SQL Server, Oracle, Microsoft Access, MySQL ja PostgreSQL ymmärtävät SQL-kieltä. Kielen yleisyydestä reaali-tietokantojen yhteydessä huomataan, että reaali-tietokannoistakin puhutaan usein SQL-kantoina. SQL-kieli on standardoitu, mutta jokaisella hallintajärjestelmällä on siihen yksilöllisiä laajennuksia. (SQL ja relaatiotietokannat s.a.)

Tiedon manipulointi tarkoittaa tietokannan sisältämien tietojen lisäämistä, poistamista, hakua, päivittämistä jne. Hakujen yhteydessä voidaan tehdä laskutoimituksia ja käyttää erilaisia suodatuksia sekä loogisia operaattoreita. Rakenteen määrittelyllä voidaan nimensä mukaisesti suunnitella tietokannan rakennetta, tauluja sekä niiden välisiä loogisia yhteyksiä. SQL-kielillä toteutetun haun tuloksena palautetaan käyttäjälle väliaikaisesti olemassa oleva taulu. Se sisältää haun kriteereinä käytettyjä attribuutteja, vertailuja ja suodatuksia vastaavat tietueet. Yleensä taulu häviää sovelluksen muistista, kun se on saatu tulostettua käyttäjän näkyville, mutta usein käytetyistä hauista voidaan muodostaa pysyviä hakutuloksia eli näkymiä. (SQL ja relaatiotietokannat s.a.)

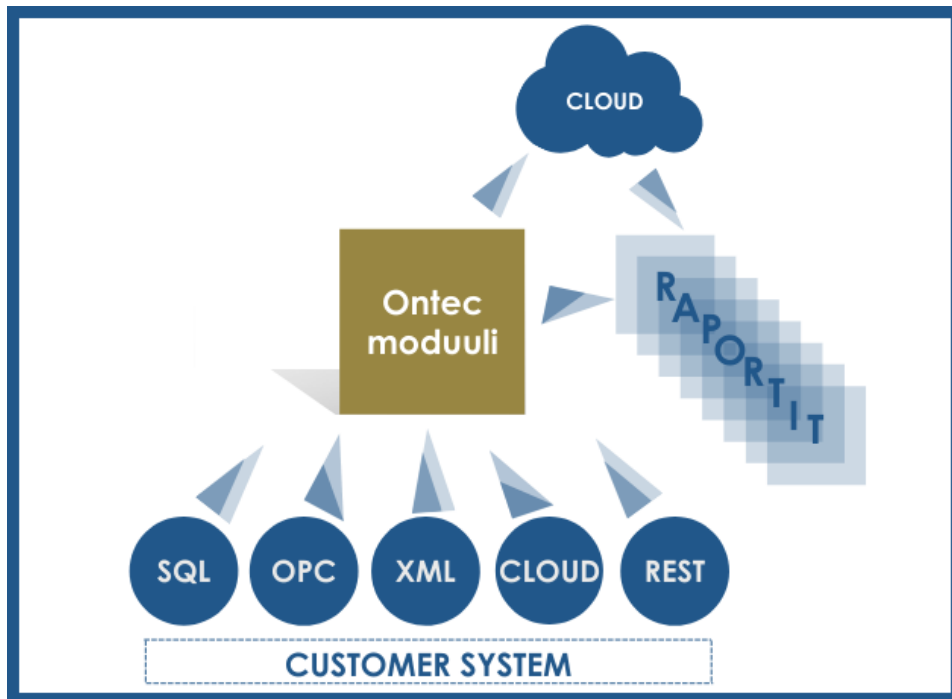
Massiivisten internetpalveluiden yleistyessä on relaatiotietokannan rinnalle syntynyt ns. NoSQL-tietokantoja. NoSQL on yleisnimitys, jolla viitataan, että tiedon organisointi vain poikkeaa perinteisestä relaatiotietokantatyypistä ja pyrkimys on olla joustavampi modernissa käyttöympäristössä. NoSQL-tietokannat eivät pyri samantasoiseen tiedon yhtäpitävyyteen ja reaaliaikaisuuteen, kuten perinteiset relaatiotietokannat, mutta niiden kapasiteettia voidaan lisätä kuormituksen kasvaessa. NoSQL-tietokantatyyppejä ovat avain-arvo-, dokumentti-, sarake- ja verkkotietokannat. Toisin kuin relaatiotietokantojen tapauksessa, niille ei ole standardoituja tai kattavia kyselykieliä. (Manoochehri 2013, 29–30; Tietokantojen perusteet 2019; Åkerman 2018, 19–20.)

3 TARKASTELTAVA OHJAUSJÄRJESTELMÄYMPÄRISTÖ

Ontec Oy tuotteet ovat ohjelmisto- ja järjestelmäratkaisuja öljy-, kaasu-, kemian- ja energiateollisuuden yrityksille. Pääasiallisena kohteena ovat Euroopan Unionin mittauslaitedirektiivi (MID) täyttävät nesteiden käsittely- ja lastausjärjestelmät, mutta myös muut teollisuusprosessit. Näiden ohessa Ontec tarjoaa kulunvalvonta-, palosammutus- ja saattolämmitysjärjestelmiä sekä ylläpito- ja kyberturvallisuuspalveluja. Ontec (2022b) nimeää omiksi vahvuusikseen: asiakkaiden tuotantoprosessien tuntemuksen, kokonaisuuksien hallinnan, osaaminen, ketteryyden ja vasteajat. Näitä ominaisuuksia voidaan peilata tiedon hierarkiaan sekä Datasta liiketoimintaosaamiseen -mallin palvelutasoihin.

Ontec Oy:n tuotteiden lähtökohtana on modulaarisuus, huollettavuus sekä skaalautuvuus. Modulaarisuudella tarkoitetaan, että yksittäiset tuotteet ovat komponentteja, joita voidaan yhdistellä asiakkaan haluamalla tavalla. Tuotteet sisältävät sekä laitteet että ohjelmistot. Ontecin järjestelmät keskustelevat eri rajapintojen ja teknisten ratkaisujen avulla, kuten REST, OPC UA, XML sekä SQL. Huollettavuus on yksi järjestelmien ja laitteiden kunnossapidettävyyden ominaisuuksista, vikojenlöydettävyyden ja korjattavuuden ohella. Kunnossapidettävyyden määrittely on SFS-standardissa kohteen kykyä olla pidettävissä tilassa tai palautettavissa tilaan, jossa se pystyy suorittamaan vaaditun toiminnon (SFS-EN 13306: 2017, 7). Tässä yhteydessä huollettavuudella tarkoitetaan hankittujen laitekomponenttien korvattavuutta ilman, että se vaikuttaa järjestelmän toimivuuteen. Skaalattavuudella tarkoitetaan muutoksia järjestelmän

käsittelmässä tietomäärässä tai käyttäjien lukumäärässä. (Ontec 2022c.) Kuvassa 24 esitellään Ontecin tuotemoduulien tiedonsiirtoa.



Kuva 24. Ontecin periaatemalli tiedonsiirrosta tuotemoduulien kanssa (Ontec 2022c)

Tarkasteltava ohjausjärjestelmäympäristö koostuu ohjelmoitavista logiikkajärjestelmistä, valvomoasemista ja levypalvelimesta. Tiedonsiirto perustuu Ethernet-siirtotekniikkaan ja tiedonsiirtoprotokollina toimivat Profinet ja TCP/IP. Prosessinohjaus sekä mittaussignaalien käsittely on toteutettu Siemensin Simatic tuoteperheen ohjelmoitavilla logiikkajärjestelmillä S7-1500 ja S7-1200. Siemensin WinCC Unified -valvomosovelluksen käyttöliittymä toimii ihmisrajapintana prosessiin ja vastaa prosessitiedon arkistoinnista, johon se tarvittaessa hyödyntää levypalvelimen tallennuskapasiteettiä. Ohjausjärjestelmän automaatioverkko on eriytetty palomuurilla toimistoverkosta, mutta sen valikoiduilla työasemilla on pääsy levypalvelimen palveluihin palomuurin lävitse. Kuvassa 25 oleva ohjausjärjestelmäympäristö on yleisen tason periaatemalli. Se ei liity suoranaisesti mihinkään esimerkkiproessiin ja sitä tarkastellaan vain valvomo- ja prosessinohjaustasolla. Periaatemalli kuitenkin vastaa Ontecin luomaa koejärjestelyä, jossa Siemensin WinCC Unified -valvomosovellus ajetaan jo käytössä olevan prosessin rinnalla (Käyhty 2022).

Ohjelmoitavat logiikkajärjestelmät Siemens Simatic -sarja

Mikroprosessori on yksi teollistumisen kolmannen aikakauden merkittävimmistä innovaatioista. Se on ohjelmoitavien logiikkajärjestelmien (engl. programmable logic controller) keskeisin komponentti ja merkittävin etu verrattaessa ohjelmoitavia logiikkajärjestelmiä niitä edeltäneisiin relelogiikkajärjestelmiin. Relelogiikkajärjestelmät ovat mekaanisesti ohjelmoitavia ohjausjärjestelmäkomponentteja, joiden tapauksessa muutokset prosessinohjauksessa vaativat johdotuksien uudelleen järjestelyä. Ohjelmoitavien logiikkajärjestelmien kanssa nämä muutokset tehdään logiikkajärjestelmän ajamaan sovellusohjelmaan eikä muutoksia kytkentöihin yleensä vaadita. Mikroprosessoripohjaisuus mahdollistaa ohjelmallisen uudelleenohjelmoinnin lisäksi monimutkaistenkin laskutoimituksien suorittamisen loogisten operaatioiden ohella. (Dey & Sen 2022, 13, 15–18.)

Järjestelmän toimintaperiaate perustuu kiertävään ohjelmasykliin. Kuvassa 26 on esitetty yhden ohjelmasyklin kierros. Sykli alkaa tuloliitäntöjen lukemisella. Seuraavaksi ajetaan ohjelmoitu sovellus, jonka syötteinä toimivat tuloliitäntöistä luetut tiedot. Sovelluksen antamat tulosteet päivitetään lähtöliitäntöille, jotka toimivat järjestelmän ohjauslähtöinä. Neljäntenä osana on tiedonsiirto ja järjestelmän oman toiminnan diagnostiikka. (Dey & Sen 2022, 28–29.)

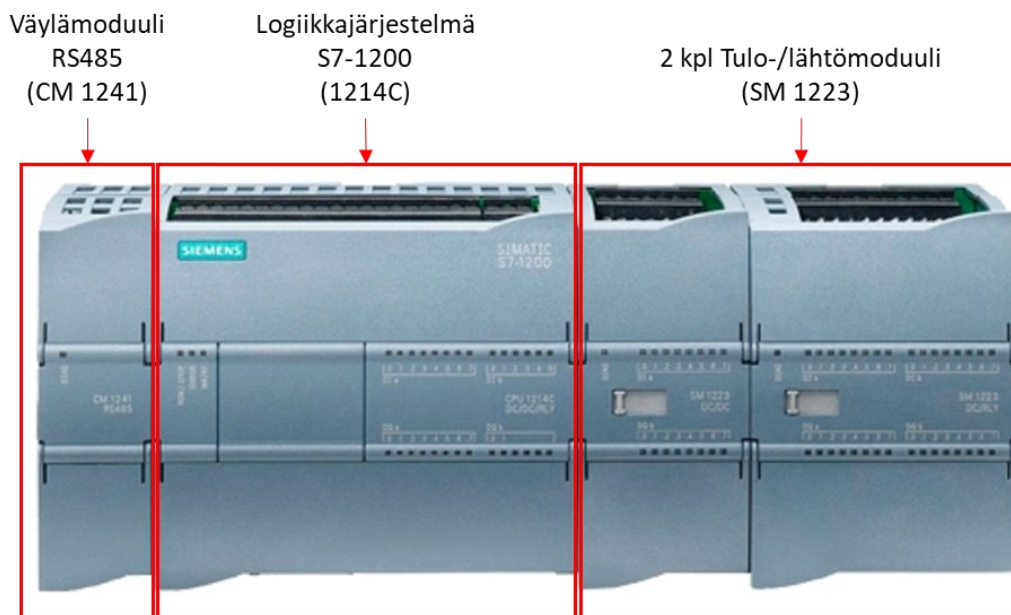


Kuva 26. Logiikkajärjestelmän ohjelmasykli (mukaillen Dey & Sen 2022, 29)

Logiikkaohjelmoinnissa muuttujat ovat yksi sovelluksen peruselementeistä kuten yleisesti ohjelmoinnissa. Ne ovat ohjelman osia, jotka toimivat tai joiden pohjalta toimitaan. Muuttujiin liittyy olennaisesti niiden tietotyyppi eli millaista tietoa muuttuja sisältää. Logiikkaohjelmoinnin perinteisiä tietotyyppejä ovat binäärikoodiksi muunnetut numerot, mutta Siemensin ympäristössä tuetaan myös ohjelmointikielistä tuttuja kokonaislukuja, merkkijonoja, tietorakenteita sekä taulukoita. Muuttujat ovat sidottu laitteen muistiosoitteisiin tai tulo- ja lähtöliitännöiden eli siirrännäisten osoitteisiin. Muuttujien arvot ovat vain hetkellisiä, ellei niitä erikseen arkistoida. (Dey & Sen 2022, 35; Siemens 2021f. 5138–5139, 5271–5275.)

Logiikkaohjelmoinnissa muuttujista käytetään usein nimitystä PLC tag, mutta terminologia on hyvin kirjavaa. Myös kirjallisuudessa sekä manuaaleissa viittaukset logiikkajärjestelmän muuttujiin, ja niistä arkistoituun tietoon, vaihtelevat jopa saman materiaalin sisällä. Tärkein periaatteellinen jako perustuu siihen, onko arvo peräisin suoraan muuttujasta vaiko peräisin arkistotietokannasta. Kaikkiin tilanteisiin sopivia yksittäisiä termejä on kuitenkin hankala muodostaa, niin etteivät ne olisi kielellisesti kömpelöitä. Seuraavissa kappaleissa logiikkajärjestelmän muuttujasta käytetään nimitystä tagi. Tagilla voi olla arvo, joka on yleensä reaaliaikaisin tieto prosessista tai ohjelman tilasta. Yleisellä tasolla voidaan puhua tagitiedosta, kun käsitellään useampaa muuttujaa. Arkistoitu tieto on tagin saamia arvoja eri ajanhetkillä. Tämän tapauksessa pyritään tekstissä painottaamaan sanaa arkisto. Vertailun vuoksi tag-sanalla viitataan prosessiteollisuudessa yleensä automaatioposition, kuten säätö-, mitaus- tai venttiilinohjauspiiriin.

Siemens on yksi isoimmista logiikkajärjestelmien valmistajista yhdessä Rockwell Automationin (Allen-Bradley), Schneider Electricin, Mitsubishi Electricin, Omronin, ABB:n ja GE:n kanssa. Siemensin yleisimmät logiikkajärjestelmät isompiin sovellutuksiin ovat S7-1200 ja S7-1500 ja pienempiin sovellutuksiin S7-300 ja S7-400. Logiikkajärjestelmät ovat moduulipohjaisia, jolloin järjestelmien liitännöitä voidaan laajentaa tarvittavilla tulo- ja lähtöliitännöillä sekä teknologiamoduuleilla, kuten robotiikan korkean tarkkuuden asennoittimilla. (Dey & Sen 2022, 23; Siemens 2014, 23; Siemens 2022e, 2; Siemens 2022f, 2.) Kuvassa 27 on Siemensin Simatic S7-1200 ohjelmitava logiikkajärjestelmä, jota on laajennettu väylämoduulilla ja kahdella tulo- ja lähtömoduulilla.



Kuva 27. Siemens Simatic S7-1200 ja laajennusmoduuleja (Siemens 2022e, 2)

Logiikkajärjestelmät kommunikoivat kentälaitetason kanssa yleisesti sarjaväyläliikenne protokollilla tai perinteisillä virta- ja jänniteviesteillä. Siemensin logiikkajärjestelmissä nämä toteutetaan soveltuvilla lisämoduuleilla tai joissakin logiikkajärjestelmissä kiinteästi olevilla liitännöillä, kuten kuvassa 27 olevassa S7-1214C-logiikkajärjestelmässä. Se sisältää kiinteät liitännät perinteisille jänniteviesteille ja niiden määrää on kasvatettu kahdella lisämoduulilla. Sarjaväyläliikenne kyky on lisätty RS485-väylämoduulilla. Siemensin logiikkajärjestelmistä löytyy kiinteänä Ethernet-siirtotekniikan RJ45-liitin. Sitä voidaan käyttää PROFINET-protokollalla kentälaitetason tiedonsiirtoon esimerkiksi moottorihjauksien kanssa, mutta ennen kaikkea se on tarpeellinen valvomo- ja prosessinohjaustason kommunikointiin. Tähän tarkoitukseen logiikkajärjestelmät tukevat mm. TCP/IP-, PROFINET- ja Modbus TCP -protokollia sekä OPC UA -rajapintaa. (Dey & Sen 2022, 27, 57–62; Siemens 2014, 23, 518; Siemens 2021g, 27–28, 73; Siemens 2022e, 2.)

Logiikkajärjestelmien ohjelmointi tehdään erillisellä ohjelmointityökalulla, joka nykyaikana on yleisimmin asennettuna PC-pohjaiseen suunnitteluasemaan. Ohjelmointi voidaan toteuttaa graafisilla ohjelmointikielillä, kuten Ladder Logic (LAD), joka muistuttaa relelogiikkajärjestelmän sähkökaaviota tai Function Block Diagram (FBD), joka muistuttaa lohkokaavio-ohjelmointia. Ohjelmointi voidaan toteuttaa myös tekstipohjaisilla ohjelmointikielillä, kuten Statement

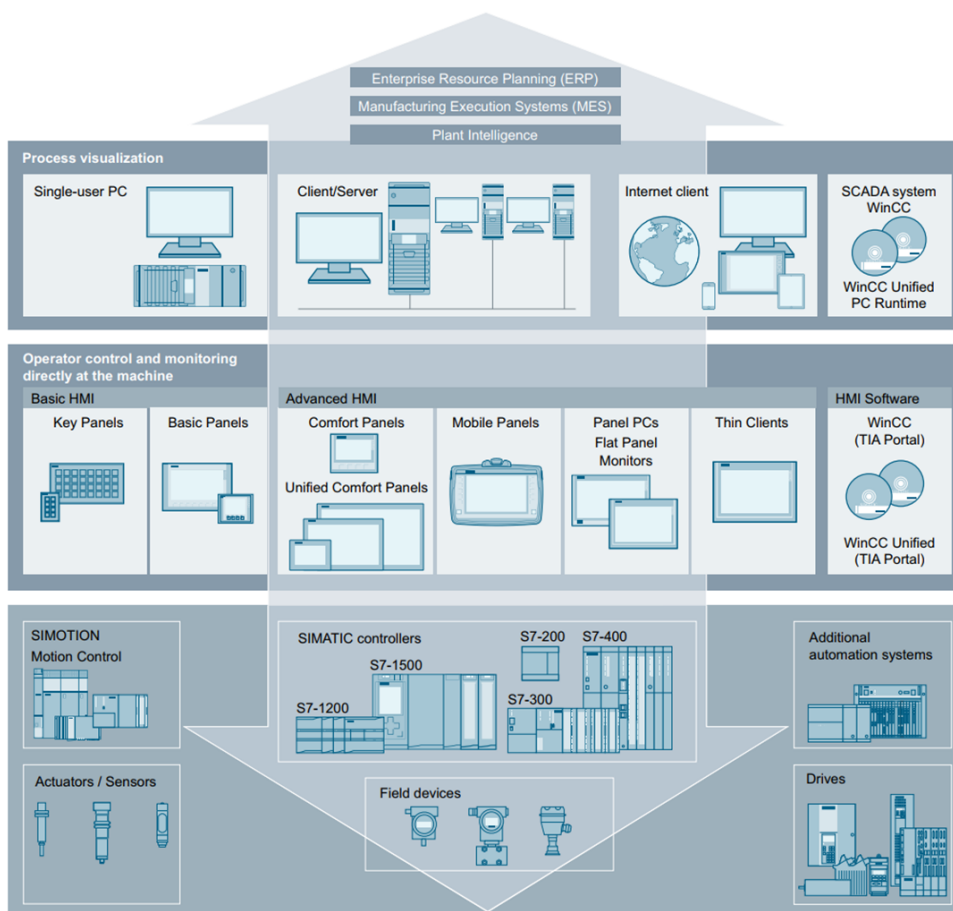
List (STL) tai Structured Control Language (SCL). Ohjelmat ladataan työkalulla logiikkajärjestelmän ohjelmamuistiin, josta ne suoritetaan. Ohjelmointityökalulla voidaan myös valvoa logiikkajärjestelmän ohjelman suorittamista erillisessä monitorointitilassa. Tässä valvontatilassa muuttujien arvoja voidaan sekä tarkkailla että muuttaa. (Dey & Sen 2022, 21–22, 28, 30; Siemens 2021f, 12895, 12957, 13013, 13061.)

Siemensin logiikkajärjestelmien ohjelmointi tapahtuu STEP7-työkalulla. Se on osa Siemensin TIA Portal -suunnitteluohjelmistoa. Siemensin tavoitteena on ollut yhdistää automaation suunnittelun yhden sovelluksen alle, jolloin laitteen hallinta, verkkojen määrittely, sovellusten ohjelmointi ja käyttöliittymien visualisointi voidaan toteuttaa yhtenä projektikonaisuutena. TIA Portal -sovellukseen on integroitu STEP7:n lisäksi mm. SINAMICS- ja SIMOCODE-moottorihjauksien parametrintyökalut sekä WinCC-valvomosovelluksien ohjelmointityökalun. Lisäksi Siemensin simulointityökalut, kuten logiikkajärjestelmiä simuloiva PLCSIM Advanced, ovat integroitu TIA Portal -sovellukseen, jolloin se toimii myös testaus- ja simulointiympäristönä. TIA Portal -sovellusta voidaan hallinnoida myös Siemensin Openness Engineering -konseptiin kuuluvalla TIA Portal Openness -rajapinnalla. Rajapinnan avulla voidaan toteuttaa esimerkiksi räätälöityjä sovelluksia automatisoimaan osia logiikkaohjelmoinnista. (Siemens 2019b, 4, 20–21; Siemens 2020c, 17.)

Valvomosovellus Siemens WinCC Unified

Ihminen on vuorovaikutuksessa ohjattavaan prosessiin useilla eri tasoilla, mutta kaksi perinteisintä ovat laite- ja prosessinohjaustasot. Vuorovaikutus tapahtuu siihen tarkoitettulla käyttöliittymällä ja sen alla toimivalla sovelluksella. Laitetasolla keskitytään hallinnoimaan yksittäistä konetta ja esittämään sen toimintatilaa kuvaavia tietoja käyttöpäätteissä. Ylemmillä prosessinohjaustasoilla toteutetaan varsinainen kokonais- tai osaprosessin valvonta, ohjaus, tiedonkeruu sekä prosessitiedon arkistointi. (Dey & Sen 2022, 77–78.) Seuraavissa kappaleissa käytetään pääosin yleisnimitystä valvomosovellus, vaikka kyse saattaisikin olla teoreettisesti laitetason ohjauksesta, johon termi valvomo ei varsinaisesti sovellu.

Siemens WinCC-ohjelmistoperhe sisältää lukuisan määrän sovelluksia eri prosessinohjaustasoille. Ylemmän ohjaustason valvomosovelluksia ovat WinCC Advanced, WinCC Professional ja WinCC Unified, jotka skaalautuvat myös laitetasolle. Pienempiin sovellutuksiin on saatavilla WinCC Basic, joka on käytettävissä ainoastaan laitetasolla. Ylemmällä tasolla valvomosovellukset ajetaan pääosin PC-pohjaisissa järjestelmissä ja laitetasolla ne ajetaan Siemensin valmistamissa näppäin- tai käyttöpaneelissa, joiden tuotenimet vastaavat sen kanssa käytettävää sovellusta. Edellä mainitut WinCC-sovellukset ovat kaikki integroituna TIA Portal -suunnitteluohjelmistoon. (Siemens 2021h, 10.) Kuvassa 28 on esitelty vuorovaikutustasoja, sovelluksia ja järjestelmiä. Ylimpänä on valvomotaso, jossa valvomosovelluksia ajetaan PC-pohjaisissa järjestelmissä. Keskellä on ohjaus laitetasolta, jossa sovellusta ajetaan Siemensin näppäin- ja käyttöpaneelissa. Tällä tasolla havainnollistuu myös jako WinCC Basic, Advancedin ja Unifiedin käyttöpäätteiden välillä. Tämän tason alla ovat logiikkajärjestelmät ja moottorihjaukset ja aivan alimpana ovat näihin liitetyt kenttälaitteet ja sensorit.

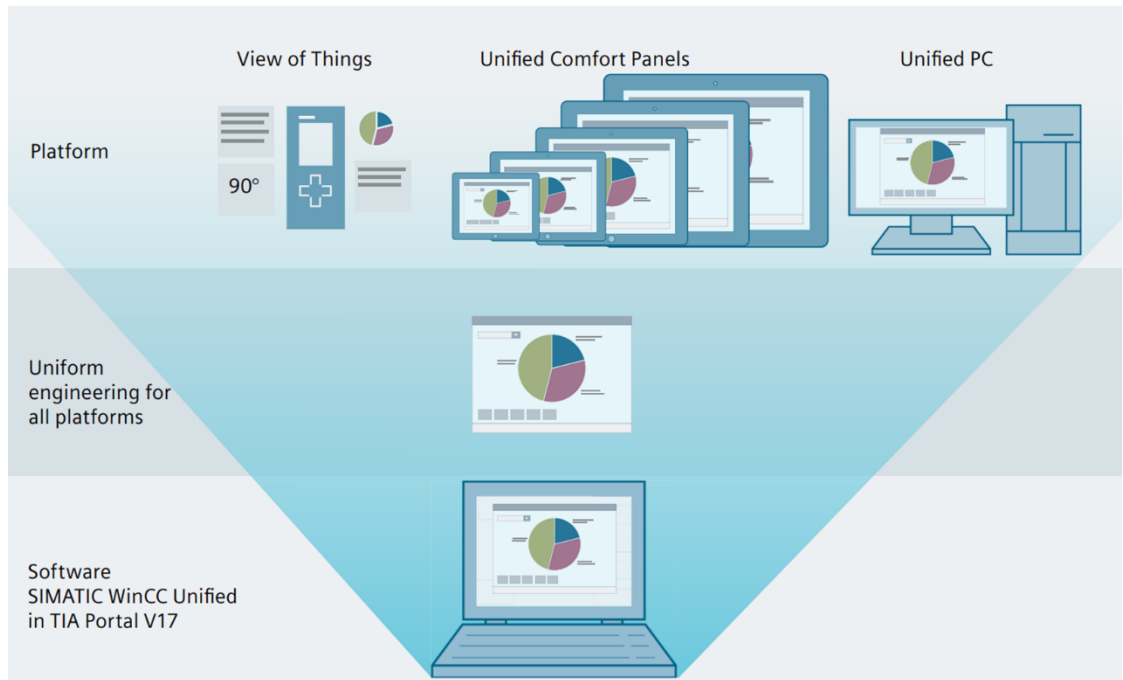


Kuva 28. Prosessin vuorovaikutustasot ja Siemensin käyttöpäätteratkaisut (Siemens 2021h, 11.)

Siemens tarjoaa valvomosovelluksia myös ilman TIA Portal integraatiota. Näistä WinCC V7 on Advanced- ja Professional-versioihin verrattava sovellusratkaisu, joka on tarkoitettu Windows-pohjaisiin järjestelmiin. Sen pääte viittaa versionumeroon ja sen edeltäjänä oli WinCC V6. Toinen vaihtoehtoista on WinCC Open Architecture (OA), joka nimensä mukaisesti on avoimen arkkitehtuurin valvomoratkaisu. Se skaalautuu yhden järjestelmän ratkaisusta, jopa 2048 serverin monimutkaisiin sovellutuksiin. WinCC OA:n tärkeimpinä ominaisuuksina ovat hyvä ja tietoturvallinen integroitavuus eri laitevalmistajien tuotteiden välillä ja alustavapaus. WinCC flexible on laitetason ohjauksiin tarkoitettu valvomosovellus, jota voidaan laajentaa lukuisilla lisäosilla, kuten reseptienhallinta, arkistointi, kulunvalvonta, diagnostiikka jne. Sitä ajetaan Siemensin käyttöpaneelissa tai PC-pohjaisissa järjestelmissä. (Siemens 2021h, 178–179, 203, 208, 220.)

WinCC Unified on Siemensin valvomosovellus, joka on integroituna TIA Portal -suunnitteluohjelmistoon versiosta 16 lähtien. WinCC Unifiedin isoin uudistus sitä edeltäneisiin WinCC Advanced ja WinCC Professional versioihin on käyttöliittymän alustavapaus. Valvomosovellus ajetaan edelleen Windows PC -pohjaisessa järjestelmässä, mutta valvomokäyttöliittymä voidaan esittää minkä tahansa alustan modernissa verkkoselaimessa. Tärkeimpiä teknologioita WinCC Unifiedin taustalla ovat HTML5 web-teknologiat ja siihen olennaisesti liittyvä JavaScript-komentosarjakieli. JavaScriptillä toteutetuilla komentosarjoilla eli skripteillä voidaan automatisoida ohjaustoimintoja, luoda täysin uusia käyttöliittymä toiminnallisuuksia sekä hakea tietoja arkistosta ODBC-rajapinnan avulla. (Siemens 2021h, 130–131; Siemens 2022d, 29, 34–35, 726.)

Kuvassa 29 esitetään, miten yhdellä TIA Portal -suunnitteluasemalla voidaan ohjelmoida kaikki WinCC Unified -yhteensopivat alustat samalla kertaa. Oikealla ylärivissä on PC-pohjainen järjestelmä (Unified PC), jota kuvattiin edellisessäkin kappaleessa. Keskellä ylärivissä ovat Siemensin Unified Comfort -näyttöpaneelit laitetason ohjauksiin. Vasemmalla ylärivissä on WinCC Unifiedin kevyt versio nimeltään View of Things, joka on toiminnallisuuksiltaan hie-man karsittu, mutta sitä voidaan ajaa suoraan Siemensin S7-1500 logiikkajärjestelmässä. Sen käyttöliittymä on myös selainpohjainen. (Siemens 2022d, 29, 38)

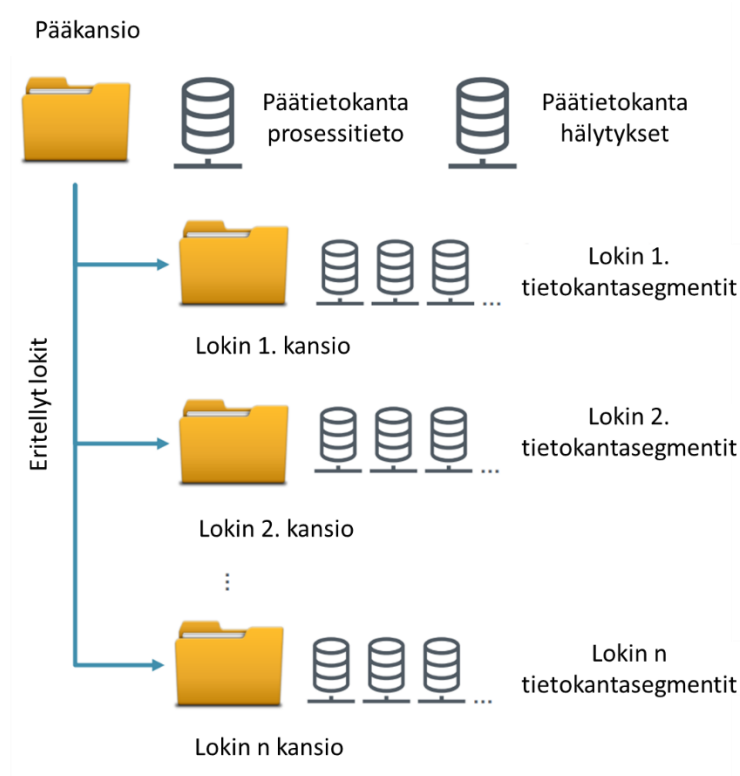


Kuva 29. WinCC Unified -alustojen ohjelmointi (Siemens 2022d, 29)

Unified PC -ympäristössä valvomosovelluksen ytimeistä käytetään nimitystä Runtime, joka toimii mm. serverinä käyttöliittymälle ja vastaa arkistoinnista. Selainpohjaisuudesta johtuen valvomokäyttöliittymän operointi ei vaadi kuin HTML5-yhteensopivan selainsovelluksen, kuten Google Chrome, Microsoft Edge, Apple Safari tai Mozilla Firefox. Yhdistettävien valvomoasemien eli asiakkaiden lukumäärä on kuitenkin rajoitettu lisensoinnilla. WinCC Unified Runtimein yhteydessä tulee lisenssi kahdelle asiakkaalle, mutta niiden määrää voidaan tarvittaessa kasvattaa aina sataan asti. Vastaavalla tavalla WinCC Unified Runtimein arkistointi on rajoitettu lisensoinnilla. Lisenssistä riippuen, arkistointia voidaan toteuttaa 100, 500, 1000, 5000, 10 000 tai 30 000 tagista. (Siemens 2021h, 130–134.)

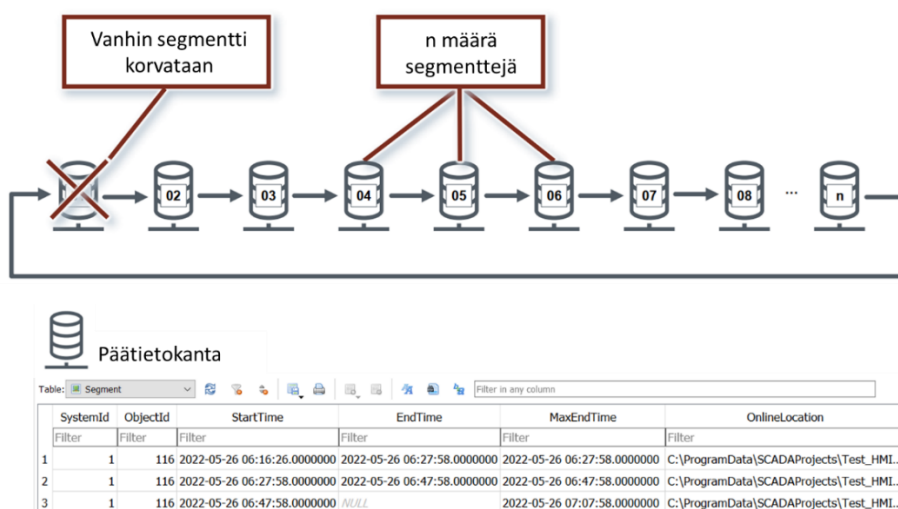
WinCC Unified Runtimein arkistointijärjestelminä toimivat tiedostopohjainen SQLite- tai serveripohjainen Microsoft SQL Server -tietokannanhallintajärjestelmät. Tiedostopohjaisella SQLitellä voidaan arkistoida tietoja 5000 tagista, kun Microsoft SQL Serverin kanssa rajoitteena on ainoastaan arkistointia toteuttavan järjestelmän suorituskyky. Arkistointi jakautuu kahteen pääarkistoon. Hälytysarkistoon tallennetaan nimensä mukaisesti tapahtuvien hälytyksien luokka, tilatieto, viesti ja aikaleima. Prosessitiedon arkistoon tallennetaan tagien arvot aikaleimoineen. Molemmat arkistot jakautuvat useampaan tietokan-

taan. Arkistolle on aina oma päätietokantansa, jolla seurataan varsinaisen prosessitiedon tallentamista ja siihen liittyviä asetuksia. Arkistointia voidaan vielä jakaa eri osiin, joista käytetään nimitystä loki. Jokaiselle lokille on oma kansionsa, jonka sisällä on tiedon tallentamiseen käytetty tietokannat. (Siemens 2021e, 6–7, 10; Siemens 2021h, 131.) Kuvassa 30 on havainnollistettu tätä arkistoinnin tietokanta- ja kansiorakennetta.



Kuva 30. Arkistoinnin rakenne WinCC Unified -ympäristössä (mukaillen Siemens 2021e, 10)

Prosessitiedon ja hälytyksien arkistointiin käytetään kiertävää tietokantasegmentointia. Segmentti rajoittuu joko ajanjakson pituuteen tai suurimpaan sallittuun tiedostokokoon, jotka määritetään valvomosovelluksen asetuksissa. Kumman tahansa rajoitteen täytyessä aloitetaan uusi segmentti eli tietokanta. Kuvassa 31 havainnollistetaan kiertävää tietokantasegmentointia sekä päätietokannan seuranta segmentin tiedoista. Kuvan ylempi osa on mukaillen lähdemateriaalia. Alempi osa on todennus opinnäytetyötä varten luodun valvomokäyttöliittymän päätietokannan Segment-taulusta, josta nähdään segmenttien aloitus- ja päättymisajat, ajanjakson pituuden rajoite sekä tallennussijainti.



Kuva 31. Kiertävä tietokantasegmentointi ja segmenttien seuranta (mukaillen Siemens 2021e, 8)

Arkistoinnin yhteydessä joudutaan useasti käyttämään sovittuja arvoja tiettyjen asioiden esittämiseen. Selkokiehisen merkkijonon käyttäminen ei ole aina tehokas menetelmä ohjelmallisen luettavuuden kannalta. Esimerkkinä hälytyksen tilaa kuvaava State-attribuutti saa arvoja 0:n ja 5:n väliltä. 1 tarkoittaa, että hälytys on aktiivinen. 5 tarkoittaa, että hälytys on aktiivinen, mutta kuitattu operaattorin toimesta. 0 tarkoittaa, että hälytyksen aiheuttanut tila on poistunut. Vastaava tilanne on myös aikaleimojen osalta. Tietotekniikassa yleisesti aikaleiman nolla-arvo on sidottu sovittuun ajanhetkeen. WinCC Unifiedin arkistoinnin kannalta nolla-ajanhetki on päivämäärä 1.1.1601. Arkistointi tapahtuu kokonaislukuna attribuuttiin `pk_TimeStamp`, jossa arvo 1 vastaa 100 nanosekuntia. Sovelluksissa, joissa aikaa käsitellään desimaalilukuina päivissä, tehdään muunnos yhtälöllä 1. Yhtälöä käyttämällä saadaan desimaalilukuarvo, joka vastaa montako päivää on kulunut ajanhetkestä 1.1.601.

$$pk_TimeStamp_{d,1601} = \frac{pk_TimeStamp * 100ns}{10000000 \frac{100ns}{s} * 60 \frac{s}{min} * 60 \frac{min}{h} * 24 \frac{h}{d}} \quad (1)$$

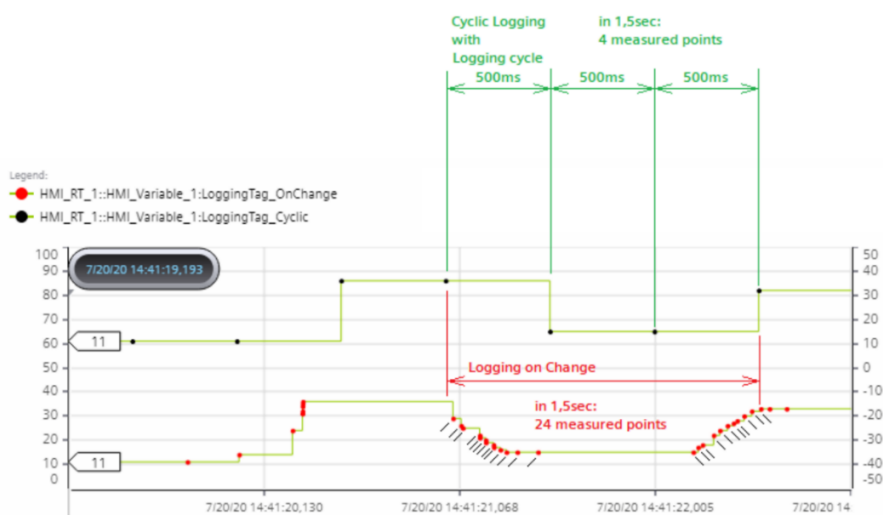
jossa $pk_TimeStamp_{(d,1601)}$ päiviä aikaleiman alusta [d]
 $pk_TimeStamp$ aikaleima [100ns]

Yhtälöllä 1 toteutetun muunnoksen lisäksi on vielä huomioitava sovelluskohtaiset ajankäsittelytavat. Esimerkiksi Microsoft Excel -taulukkolaskentaohjelma käsittelee ajan nollahetkenä, asetuksista riippuen, joko 1.1.1900 tai 1.1.1904.

Täten 1.1.1900 -asetusta käyttävässä Excel-työkirjassa tulee tehdä lisälaskutoimitus, jossa aiemman kaavan muunnoksesta vähennetään arvo 109205, joka on päivien lukumäärä 1.1.1601 ja 31.12.1899 välillä. (Microsoft s.a.; Siemens 2021e, 26.)

Tagin arkistointia määrittäessä on valittava sen arkistoinnissa käytettävä loki-nimi, johon tiedot tallennetaan sekä arkistointimenetelmä. Valittavia menetelmiä ovat syklinen (engl. cyclic) arkistointi, arkistointi pyydettyäessä (engl. on demand) tai arkistointi muutoksesta (engl. on change). Syklinen arkistointi voidaan asettaa toimimaan lyhyimmillään sadan millisekunnin välein. Arkistointi pyydettyäessä ja arkistointi muutoksesta toimivat periaatteellisesti samalla tavalla. Ne molemmat reagoivat muutokseen määritellyssä tagissa, mutta pyydettyäessä-arkistoinnin kanssa on tarkoitus hyödyntää nousevan vai laskevan reunan menetelmää muutoksen tarkkailussa. Arkistointi pyydettyäessä ja arkistointi muutoksesta on mahdollista sitoa myös eri tagiin kuin mikä on varsinaisen arkistoinnin kohteena, eli tagin X muutoksesta tagin Y arvo tallennetaan. Arkistointeihin voidaan määrittää myös raja-arvot, joiden ylä- tai alapuolella tallennus tapahtuu sekä tasoitus (engl. smoothing), jolla loivennetaan muutoksien terävyyttä. (Siemens 2021e, 18–21.)

Raportointi perustuu valvomosovelluksen arkistointiin, joten sen suunnittelu ja oikeat määrytykset ovat olennainen osa myös raportointityökalujen käytettävyyttä. Kuvassa 32 havainnollistetaan syklisen ja muutosperusteisen arkistointimenetelmien eroavaisuutta.



Kuva 32. Syklisen ja muutosperusteisen arkistoinnin ero (Siemens 2021e, 21)

3.2 Kehittämistarve ohjausjärjestelmäympäristössä

Ontecin käyttökohteita ohjausjärjestelmäympäristölle ovat erä- ja panosprosessit, joissa tapahtuu usein raaka-aineiden ja muiden valmistukseen liittyvien suureiden muutoksia. Ympäristössä halutaan jatkossa hyödyntää enemmän WinCC Unified -valvomosovellusta ja tarvittaessa muita Siemens Unified -yhteensopivia käyttöliittymäratkaisuja. Varsinainen raportoinnin luonteen muutos tai uudentyypiset ohjattavat prosessit eivät ole niinkään olleet uudistuksen ajavia tekijöitä.

Yksi merkittävimmistä syistä Unified-ympäristöön siirtymiselle on kaikkien toimintojen integroituminen TIA Portal -suunnitteluohjelmiston alle. Yhden systeemin helppous koetaan tietyissä tilanteissa eduksi, vaikka rajapinnoilla eri osapuolien tuotteiden yhdistelemisestä on hyvät aiemmat kokemukset. Esimerkkinä Käyhty (2022) toteaa TIA Portal integraatiosta, että logiikan muuttajat saadaan suoraan valvomosovelluksen käyttöön. Tämä tarkoittaa TIA Portal -suunnitteluohjelmiston etua, jossa samat muuttajat esiintyvät teoriassa sekä logiikkajärjestelmän sovellusohjelmassa että valvomosovelluksen käyttöliittymässä ja muutokset peilautuvat projektin sisällä automaattisesti. Esimerkiksi Ontecin aiemmin käyttämässä AVEVA-ohjelmistotalon InTouch-valvomosovelluksessa hyödynnetään tiedonsiirtoon OPC DA- tai OPC UA -rajapintoja ja joi-takin valmistaja kohtaisia ajureita, mutta muutoksien tekeminen ja virheiden jäljittäminen on tehtävä manuaalisesti, koska logiikkaohjelman ja valvomosovelluksen suunnitteluohjelmat eivät ole integroituna toisiinsa. (AVEVA 2020, 8; Käyhty 2022.)

Raportoinnin toteuttaminen on yksi Ontecin tuottamista palveluista ohjausjärjestelmäympäristöjen yhteydessä. Raportointi on kasvavassa määrin osa tuotantolaitoksien toimintaa, josta Käyhtykin (2022) toteaa, että nykypäivänä on lisääntynyt tarve seurata prosesseja sekä ympäristön että talouden kannalta ja tallennetusta datasta pystytään analysoimaan prosessien tehokkuutta ja suunnittelemaan muutoksia. Tällä tarkoitetaan esimerkiksi tarvetta toteuttaa toistuvaa päästöraportointia viranomaisille tai valmistusketjun poikkeamien jäljittämistä. Kuten Datasta liiketoimintaosaamisen -mallin kohdalla todettiin, raportoinnin lisääntynyt tarve mahdollistaa myös sen toteutuksen tarjoamisen tilaajayrityksille palveluina. Ontec ei varsinaisesti sovelle mallin palvelutasoja

omassa toiminnassaan, mutta Käyhty (2022) esittää samat havainnot, mitkä ovat verrattavissa Ahonen ym. (2017, 16) määritelmään Datasta liiketoiminta-osaamiseen -mallin pohjaksi, että yrityksillä ei ole välttämättä tarvittavaa osaamista tai resursseja raportoinnin toteuttamiseksi, vaan se halutaan palveluna.

Raportoinnin toteuttamiseen Ontec on aiemmin käyttänyt omia räätälöityjä sovelluksia tai Ocean Data Systems -ohjelmistotalon kehittämää Dream Report -sovellusta. Dream Report on erityisesti automaatiojärjestelmille suunniteltu raportointi- ja tiedon analysointisovellus, joka tukee ISA 95 -standardin mukaista tiedon mallinnusta eri tuotanto- ja toiminnanohjaustasojen välillä. Tietolähteinä sovellukselle voivat toimia logiikkajärjestelmät, tietokannat, valvomosovellukset sekä IoT-laitteet. Tiedonsiirto tapahtuu joko yleisillä avoimilla rajapinnoilla tai valmistajien omilla vastaavilla. Siemensin tuotteista ovat mainittuina yleisesti WinCC-valvomosovellukset ja S7-logiikkajärjestelmät. Dream Report -sovellus voi toimia reaaliaikaisena koje- tai mittaritauluna, tuottaa raportteja suoraan eri formaateissa ja toimia raportointiserverinä, jolloin sitä voi hyödyntää mm. OPC DA-, OPC UA- tai REST-rajapintojen välityksellä. (Käyhty 2022; Ocean Data Systems 2020a, 2–3; Ocean Data Systems 2020b; Ontec 2022b.)

Raportointialustojen tarkastelulle ei asetettu varsinaisia rajoitteita niiden ohjausjärjestelmäympäristöön aiheuttamien muutoksien suhteen. Laitekannan kasvusta, arkistointijärjestelmien muutoksista, ohjelmien lisenssimaksuista tai pienistä verkkoinfrastruktuurin muutoksista ei tarvinnut kantaa huolta. Raportointi voitaisiin toteuttaa joko automaatioverkon tai toimistoverkon puolelle sijoitettavalla raportointikoneella tai hyödyntää jo olemassa olevaa laitekantaa. Raportointialustojen käyttö ei toisaalta saisi missään tilanteessa häiritä tai estää valvomo-operointia.

4 RAPORTOINTIALUSTOJEN OMINAISUUSKARTOITUS

Vertailuun valitut työkalut ja rajapinnat olivat Ontecin asiantuntijoiden ennakkoon määrittämiä vaihtoehtoja. Alustojen ominaisuuksia kartoitettiin seuraavan, yhdessä Ontecin asiantuntijoiden kanssa muodostetun listan pohjalta:

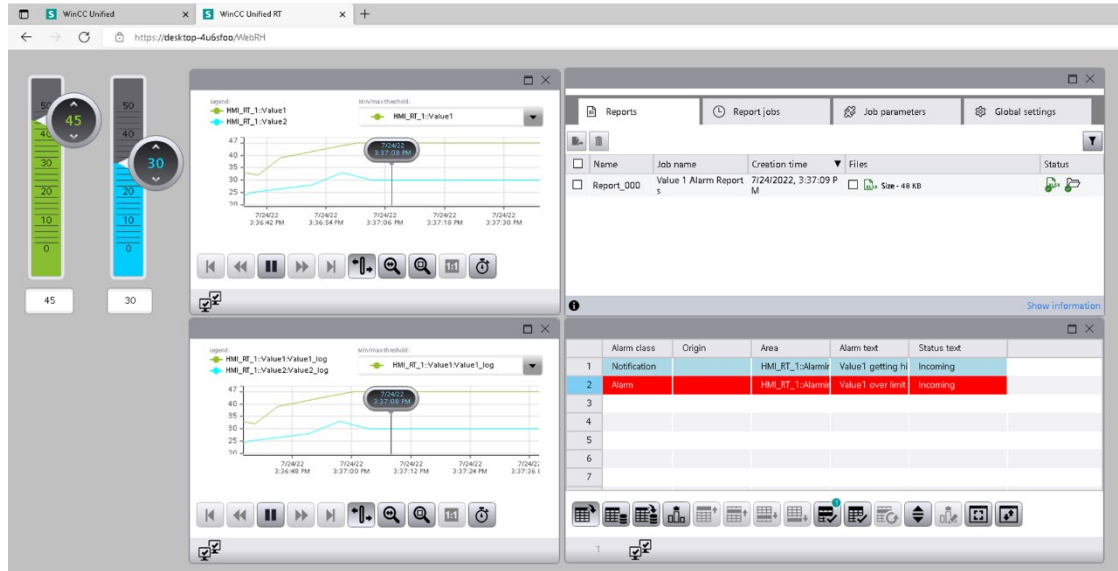
- sovellusohjelmien ajo
- raporttien konfigurointi
- raporttien tiedostoformaatti
- automatisointi aika- ja eräkohtaiseksi
- hälytyksien raportoiminen
- skaalattavuus eri käyttötilanteisiin
- tiedonsiirtomenetelmät
- työkalun vaatimukset logiikkasovellukselle
- muut oleelliset tai lisäarvoa tuovat ominaisuudet.

Kartoitustyö perustuu pääosin Siemensin tukisivujen kautta saataviin materiaaleihin ja tulkintaan näiden pohjalta. Useimmissa tapauksissa havainnot on yhdistelty useamman lähteen pohjalta. Lähteistä kaksi ei ole saatavilla internetistä vaan ovat saatu Siemensin asiakaspalvelulta ja Ontecilta tehtävänannon yhteydessä.

Kartoitustyön tueksi luotiin yksinkertainen WinCC Unified -valvomokäyttöliittymä TIA Portalin versiolla 17. Sen avulla päästiin tutustumaan ensisijaisesti valvomosovelluksen toimintaan ja arkistointiin, mutta testaamaan myös sen lisenssiin sisältyvien työkalujen ja rajapintojen ominaisuuksia. Esimerkkivalvomosovelluksella testatut ominaisuudet mainitaan tekstin yhteydessä.

Kuvassa 33 on tämä opinnäytetyötä varten luotu yksinkertainen valvomokäyttöliittymä. Valvomosovellukseen on määritetty tagit Value1 ja Value2, jotka arkistoidaan muutosperusteisesti lokinimillä Value1_log ja Value2_log. Käyttöliittymässä tagien arvoja voidaan muuttaa 0:n ja 50:n välillä liukusäätimillä, jotka sijaitsevat käyttöliittymän vasemmassa reunassa. Arvojen muutoksia voidaan seurata trendikäyrinä, joista ylempi näyttöobjekti on sidottu suoraan tageihin ja alempi samoihin tageihin, mutta arkistoinnin kautta. Tagien arvoille on asetettu

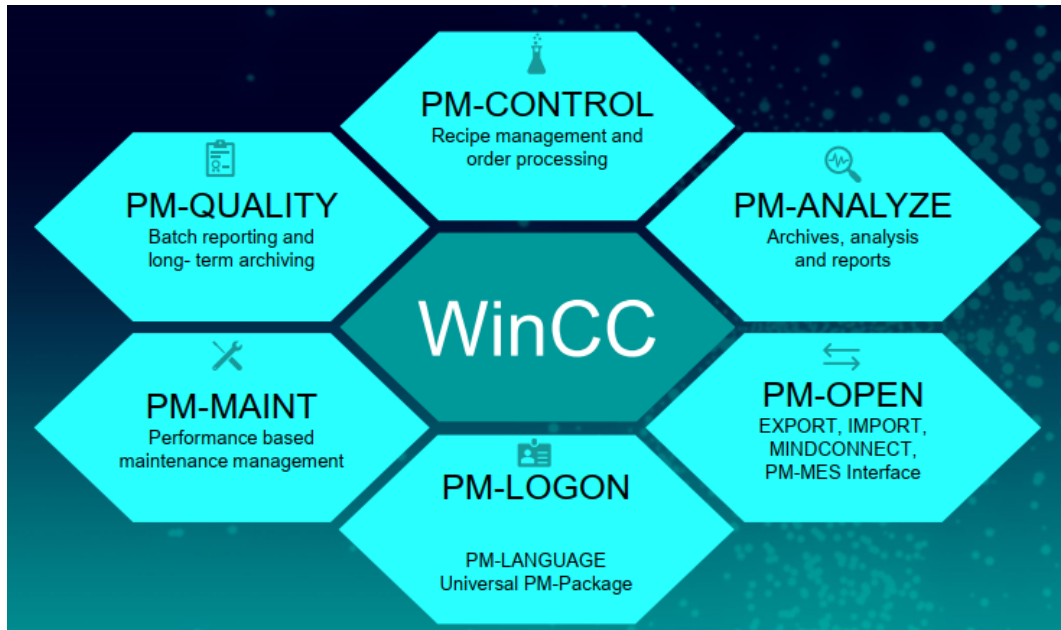
raja-arvot 40 ja 45, joista ensimmäinen laukaisee ilmoituksen (engl. notification) ja seuraava hälytyksen (engl. alarm). Niiden aktivoituminen esitetään hälytyslistaobjektissa, joka on näytössä alaoikealla. Yläoikealla oleva raportointiobjekti liittyy yhteen tässä työssä tarkasteltavista raportointityökaluista ja sen hallintaan.



Kuva 33. Opinnäytetyötä varten luotu valvomokäyttöliittymä

4.1 Process Management -lisäosat WinCC-valvomosovellukselle

Siemens tarjoaa WinCC-valvomosovelluksille modulaarista lisäosakokonaisuutta, joka sisältää työkaluja tuotantolinjojen hallintaan ja -analytiikkaan, kunnossapitoon, erityyppisiin integrointitoteutuksiin sekä tunnistautumiseen. Tuotantolinjan hallintaan ja -analytiikkaan ja kunnossapitoon keskittyvät PM-CONTROL, PM-QUALITY, PM-ANALYZE ja PM-MAINT toimivat valvomosovelluksissa WinCC V7, WinCC Professional, WinCC Advanced ja WinCC Unified. Integrointiin tai tunnistautumiseen suunnitellut laajennukset PM-OPEN ja PM-LOGON eivät tue WinCC Unified -valvomosovellusta. PM viittaa lisäosien nimissä Process Management sanoihin. Tekstissä käytetään jatkossa yleisnimitystä PM-lisäosa, kuvaamaan yleisesti mitä tahansa lisäosista, kuten PM-QUALITY, PM-ANALYZE, PM-CONTROL jne. Lisäosia käsittelevien kappaleiden sisältö perustuu useasta eri lähdemateriaalista kerättyyn ja yhdisteltyyn tietoon ja niiden tulkintaan. (PM Add-ons for WinCC s.a.; Siemens 2022a, 5–7.) Kuvassa 34 ovat Process Management -lisäosat sekä niiden lyhyet käyttötarkoituskuvaukset.



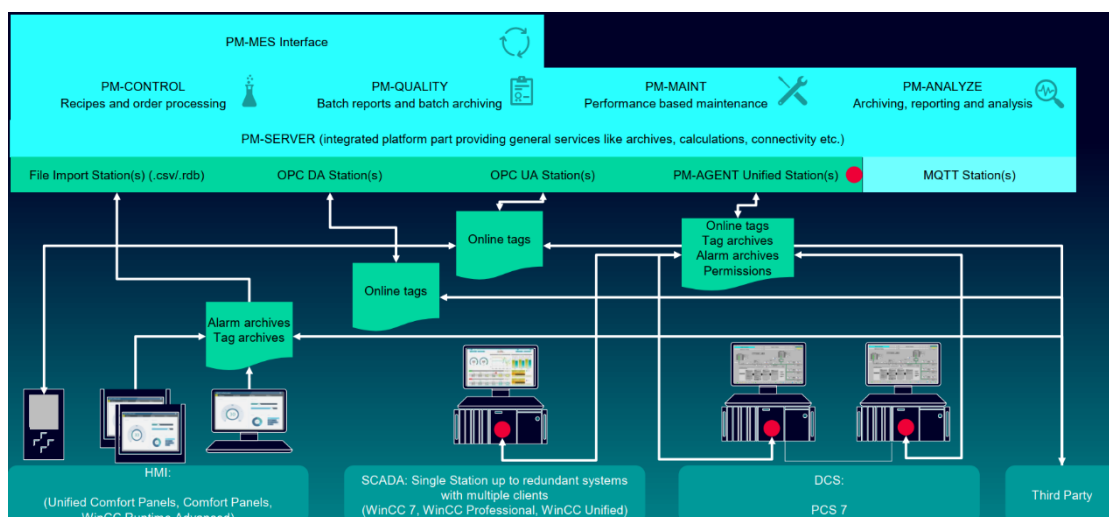
Kuva 34. Process Management -lisäosakokonaisuus (Siemens 2022a, 3)

PM-CONTROL-, PM-QUALITY-, PM-ANALYZE- ja PM-MAINT-lisäosat toimivat PM-SERVER-sovelluksen päällä. Tiedonsiirto Siemensin omien valvomosovelluksien ja PM-SERVER-sovelluksen välillä tapahtuu PM-AGENT-tiedonsiirtorajapinnalla. Tuettuja Siemensin valvomosovelluksia ovat WinCC V7, WinCC Professional ja WinCC Unified. Valvomosovelluksien WinCC Comfort, WinCC Advanced ja WinCC flexible kanssa käytetään tiedostojen tuontia CSV- tai RBD-formaateissa. Kolmannen osapuolen tietolähteiden käyttö on myös mahdollista, koska PM-SERVER tukee OPC DA- ja OPC UA -rajapintoja. PM-SERVER ja PM-AGENT eivät ole itsenäisesti hankittava ohjelmistoja vaan tulevat PM-lisäosan asennusmedian mukana. (Siemens 2020a, 6; Siemens 2022a, 13.)

PM-AGENT on kaksisuuntainen suljettu tiedonsiirtorajapinta, joka toimii TCP/IP-protokollapinon päällä. Se mahdollistaa kattavimman tiedonsiirron Siemensin omien valvomosovelluksien ja PM-SERVER-sovelluksen. PM-AGENT-rajapinnan kanssa voidaan siirtää WinCC-valvomosovelluksesta tagitietoa, tagien arkistot sekä prosessiin liittyvät hälytykset. Tästä huomiona, mikäli kolmansien osapuolten tietolähteiden kanssa halutaan siirtää vastaava tietomäärä, joudutaan käyttämään OPC-rajapinnan ja tiedostojen tuomisen yhdistelmää, koska OPC-rajapintojen kanssa yksistään on mahdollista siirtää vain

tagitietoa, muttei arkistoja. PM-AGENT tukee myös käyttäjäoikeuksien tuomisen valvomosovelluksista WinCC V7 ja WinCC Professional. (Siemens 2016, 7; Siemens 2022a, 13.)

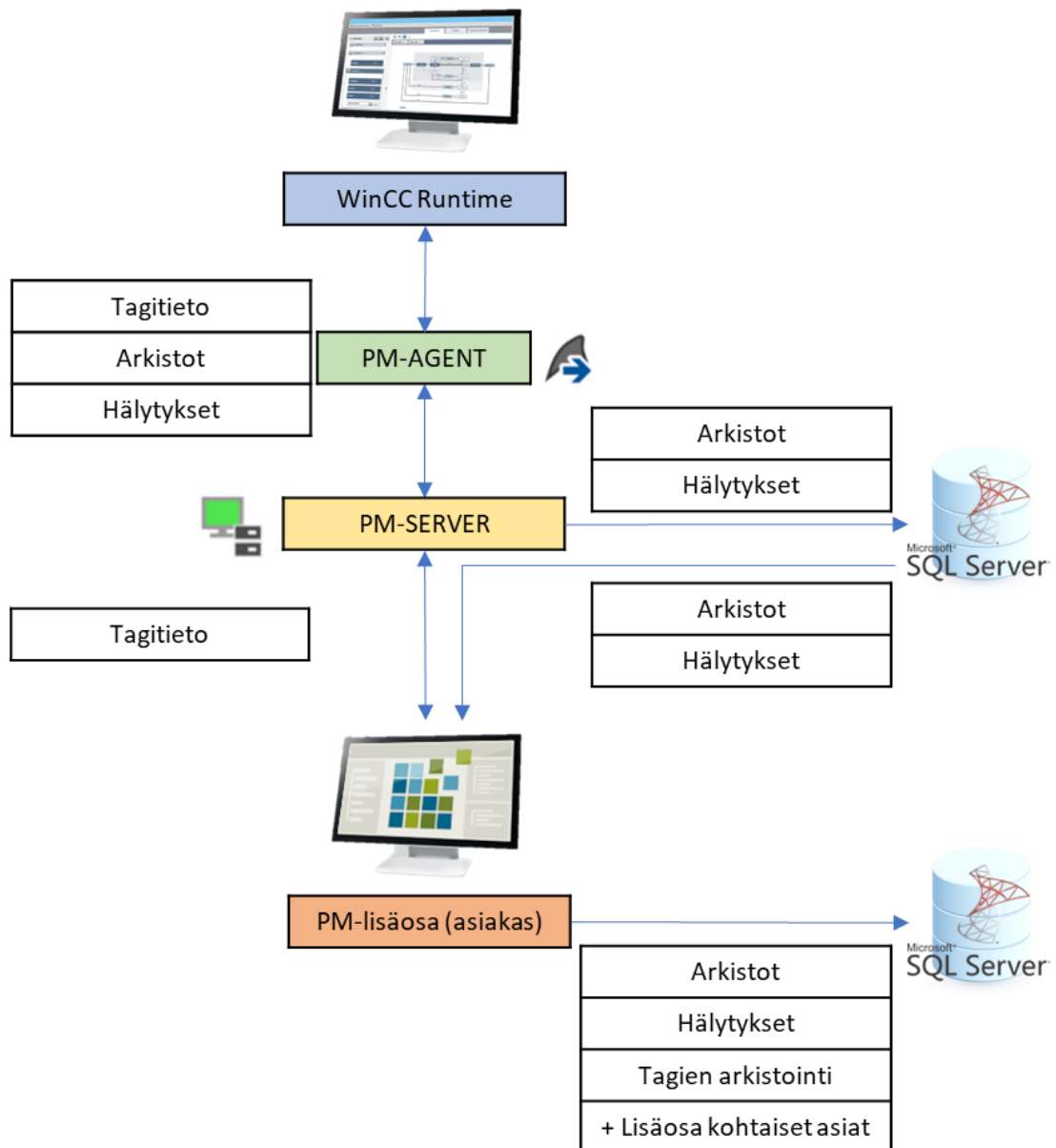
Kuvassa 35 on esitelty PM-lisäosien infrastruktuuria. Alimpana ovat valvomosovellukset, jotka edustavat eri tietolähteitä. Ne linkittyvät tiedonsiirtomenetelmiin laatikoiden kautta, joissa esitellään ko. tiedonsiirtomenetelmällä käytävissä olevat tietotyypit. Tiedonsiirtomenetelmien yläpuolella on PM-SERVER-sovellus koko kuvan levyisenä, mikä kuvaa sen roolia tiedon kerääjänä. PM-lisäosat ovat PM-SERVER-sovelluksen päällä, mikä kuvaa näiden välistä hierarkiaa. PM-CONTROL- ja PM-QUALITY-sovelluksien päällä on mahdollista hyödyntää erillistä PM-MES-rajapintaa, jos halutaan liittyä isompaan tuotannonohjausjärjestelmään. (Siemens 2022a, 119).



Kuva 35. Process Management -sovelluksien tiedonsiirto infrastruktuuri (Siemens 2022a, 13)

PM-SERVER-sovellus vastaa tiedon välittämisestä, arkistotiedon ylläpidosta sekä käyttäjäoikeuksien hallinnasta sen päällä toimiville PM-lisäosille. Yhteen PM-SERVER-sovellukseen voidaan liittää useita PM-lisäosia. Käyttäjäoikeudet voi tuoda tietyistä WinCC-valvomosovelluksista tai määrittää PM-SERVER-sovelluksessa. Tietolähteistä muodostetaan PM-SERVER-sovelluksen projektiin ns. asemat ja samalla konfiguroidaan kuhunkin tietolähteeseen käytettävä tiedonsiirtomenetelmä. Skaalautuvuuden näkökulmasta muodostettavien asemien tai siirrettävien tagien määrää ei ole rajoitettu. (Siemens 2016, 13, 22–23; Siemens 2020a, 6, 10, 24.)

Kuvassa 36 on havainnollistettu tiedonsiirtoa sovelluspinossa PM-AGENT-ra-
japinnan kanssa. PM-SERVER siirtää valvomosovelluksen arkistosta valitut
tiedot ja hälytykset omaan arkistoonsa ja ylläpitää näitä syklisesti. Tagitieto
välittyy PM-SERVER-sovelluksen kautta suoraan PM-lisäosalle. PM-lisäosista
esimerkiksi PM-QUALITY toteuttaa vielä oman arkistointinsa kaikista tiedoista.
Tagien arvoja voidaan välittää myös takaisin PM-lisäosalta WinCC-valvo-
mosovellukselle asti. (Siemens 2020b, 10–11.)

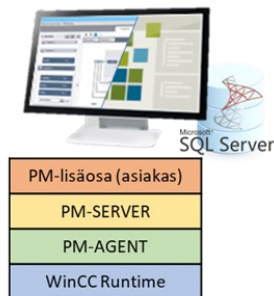


Kuva 36. Tiedonsiirto Process Management -sovelluspinossa (mukailen Siemens 2020b, 10)

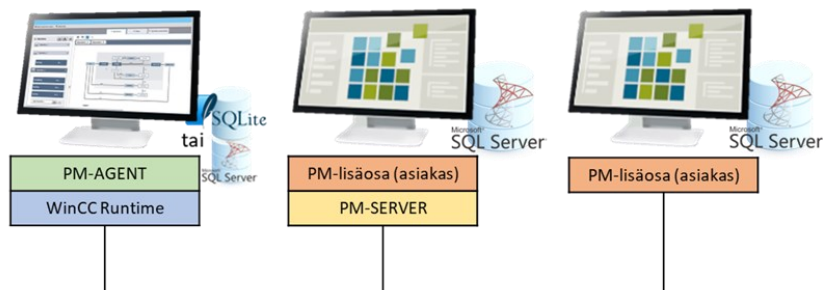
PM-SERVER ja PM-lisäosat käyttävät arkistointiinsa Microsoft SQL Server - tietokantaa. PM-SERVER ja PM-lisäosat voivat jakaa tietokannan WinCC-valvomosovelluksen kanssa, kunhan se tyypiltään Microsoft SQL Server. WinCC-valvomosovelluksen yleisesti hyödyntämän SQLite-tietokannanhallintajärjestelmän soveltumattomuutta arkistointijärjestelmäksi ei ilmaista suoraan lähdemateriaalissa, mutta tämä voidaan tulkita yhdistelemällä tietoa useammasta lähteestä. (Siemens 2021a, 13, 15, 32, 36; Siemens 2022a, 110–111; Siemens 2022b, 10–11, 263–264; Siemens 2022b, 3; Siemens 2022c, 2.)

PM-lisäosat voidaan asentaa samaan järjestelmään PM-SERVER-sovelluksen kanssa tai pelkinä asiakasohjelmina erilliseen järjestelmään. Kuvassa 37 on esitelty eri asennuskonfiguraatioita sekä arkistointijärjestelmiä, kun käytetään PM-SERVER-sovellusta ja PM-AGENT-rajapintaa.

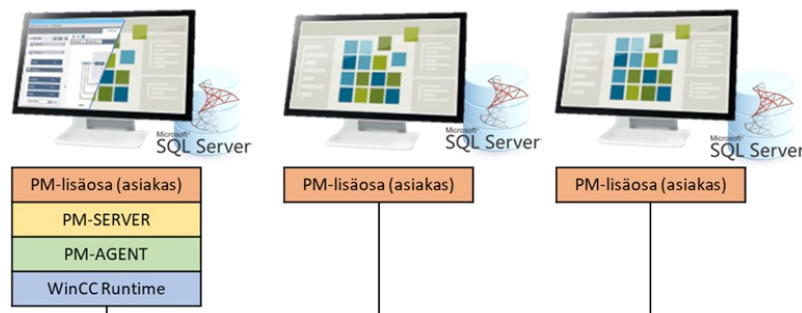
Itsenäinen:



Useamman käyttäjän:



Useamman käyttäjän:



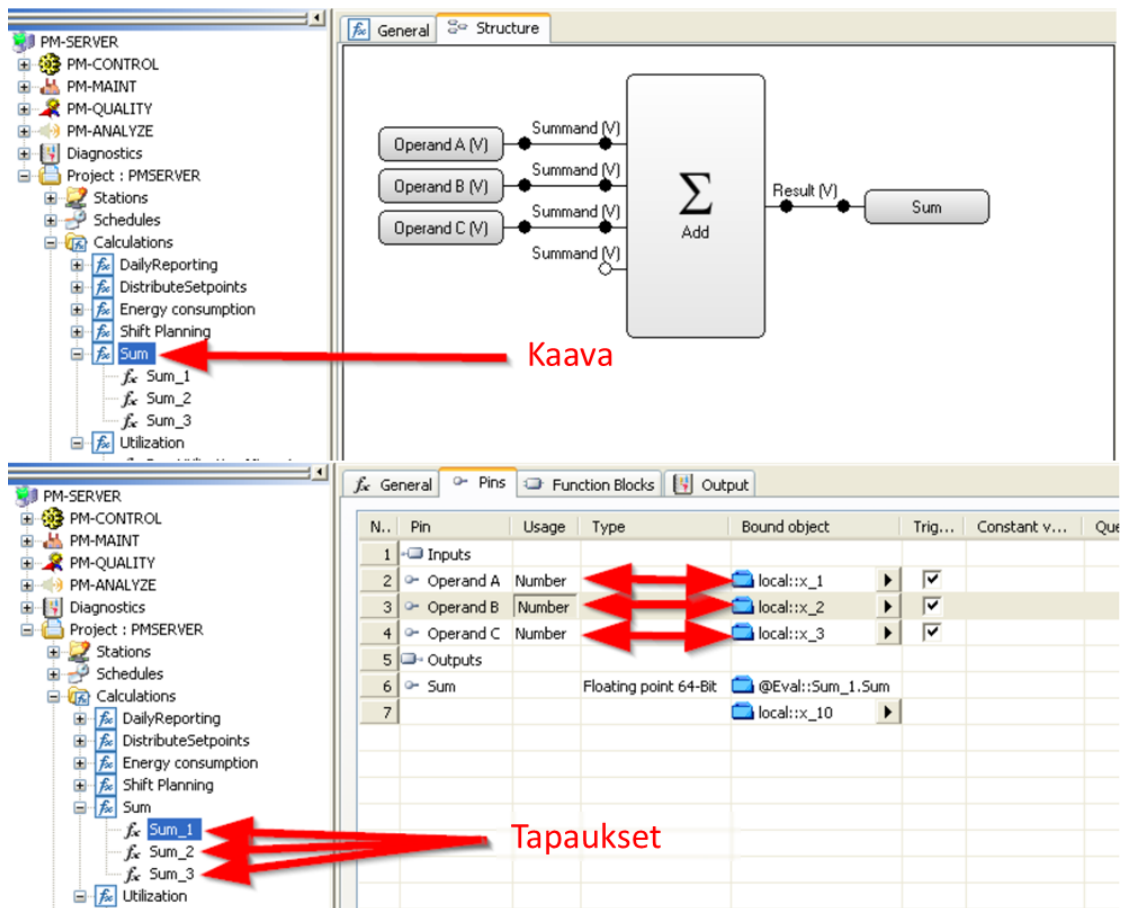
Kuva 37. Asennuskonfiguraatiovaihtoehtoja PM-lisäosille (mukailten Siemens 2019a, 16; Siemens 2022a, 111, 264)

PM-SERVER-sovelluksen raportointia tukevat ominaisuudet

PM-SERVER sisältää lukuisia toimintoja tiedon jatkokäsittelyyn ja uudelleenjäsentelyyn. Nämä toiminnot antavat lisämahdollisuuksia toteutettavaan analysointiin sekä raportointiin sekä PM-lisäosilla että suoraan PM-SERVER-sovelluksella.

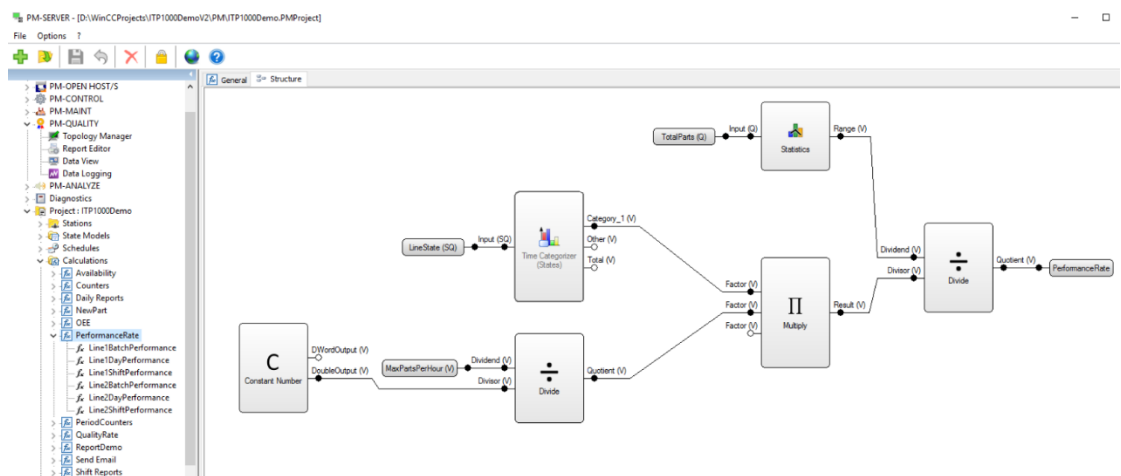
PM-SERVER-sovelluksen Calculations-toiminto mahdollistaa tallennetun tiedon käsittelyn ja erilaisten algoritmien toteuttamisen graafisella lohkokaavio-ohjelmoinnilla. Calculations-toiminnon kokonaisuus muodostuu kaavasta (engl. schema) ja tapauksesta (engl. instance). Kaavaosa määrittää lohkokaa-
vio-ohjelmoinnilla toteutetun ajettavan algoritmin. Yhdelle kaavalle voidaan luoda useita tapauksia, joissa kussakin yksilölliset syötteet ja tuloksien tallennussijainnit. Tulokset tallentuvat PM-SERVER-sovelluksen arkistoon ja niitä voidaan käyttää raporteissa tai välittää esitettäväksi WinCC-käyttöliittymässä. (Siemens 2020a, 25; Siemens 2022a, 104.)

Calculations-toiminnon ohjelmointilohkoja löytyy yleisemmille laskutoimituksille, bittioperaatioille, vertailuoperaattoreille sekä toiminnallisille operaattoreille, kuten viestien muodostus ja lähettäminen, laskurit, tilastolaskenta ja tiedostojen luominen. Lisäksi kunkin asennetun PM-lisäosan mukana tulee vielä lisäosakohtaisia ohjelmointilohkoja. Toiminnolla luotu tapaus voidaan asettaa ajettavaksi mm. aikatauluperusteisesti tai tapaukseen sidotun syöttöarvon muutoksesta, joka voi olla yhdistettynä esimerkiksi prosessinohjauksessa käytettävään tagiin. (Siemens 2016, 15; Siemens 2022a, 107.) Kuvassa 38 on muodostettu Sum-niminen kaava, jolla on tapaukset Sum_1, Sum_2 ja Sum_3. Sum_1:n syötteiksi ovat asetettuna tagit x_1, x_2 ja x_3 ja tulos tallennetaan tagiin Sum_1.Sum.



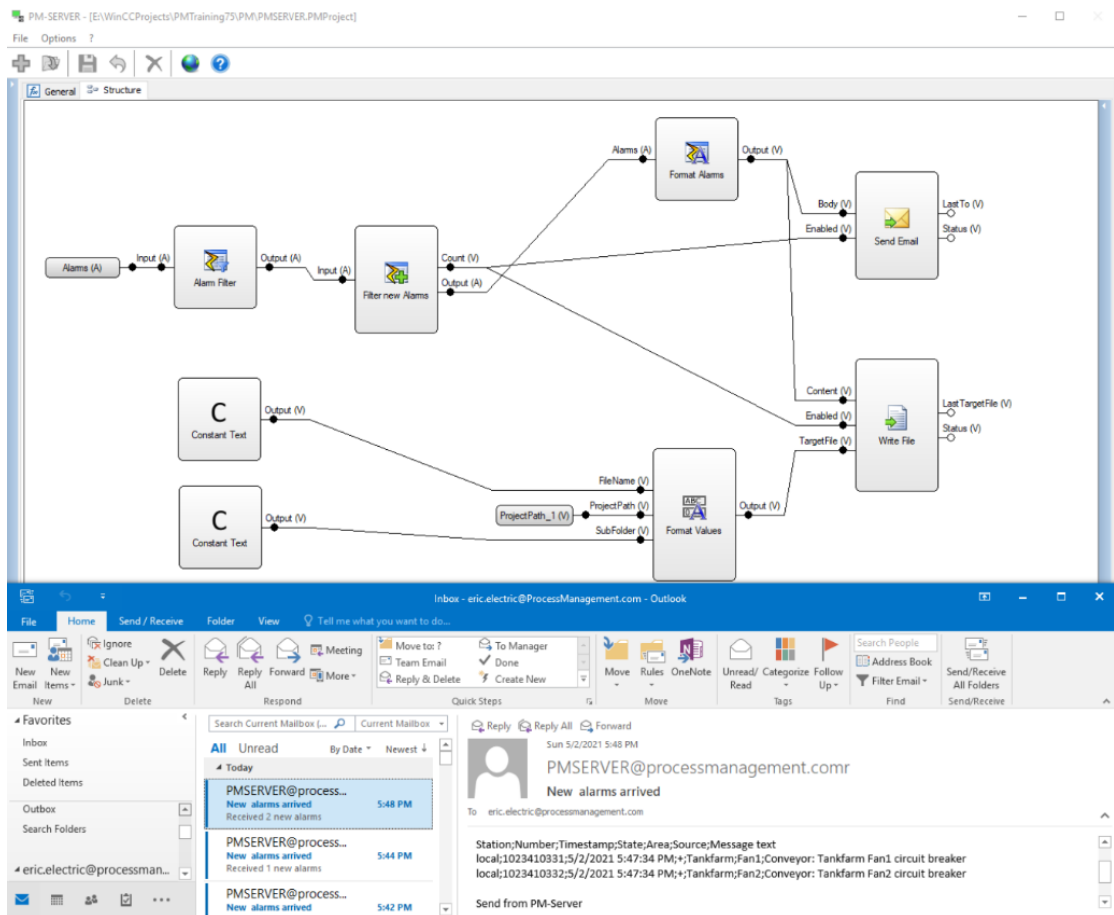
Kuva 38. Calculations-toiminnon kokonaisuus (mukaillen Siemens 2020a, 25)

Kuvassa 39 esitellään suorituskykyindikaattori laskentaa Calculations-toiminnon ohjelmointilohkojen avulla. Performance-kaava sisältää kuvan mukaisen algoritmin, jolla lasketaan erä-, päivä- ja työvuorokohtaiset suorituskykyindikaattorit tuotantolinjoille 1 ja 2 kuudella eri tapauksella.



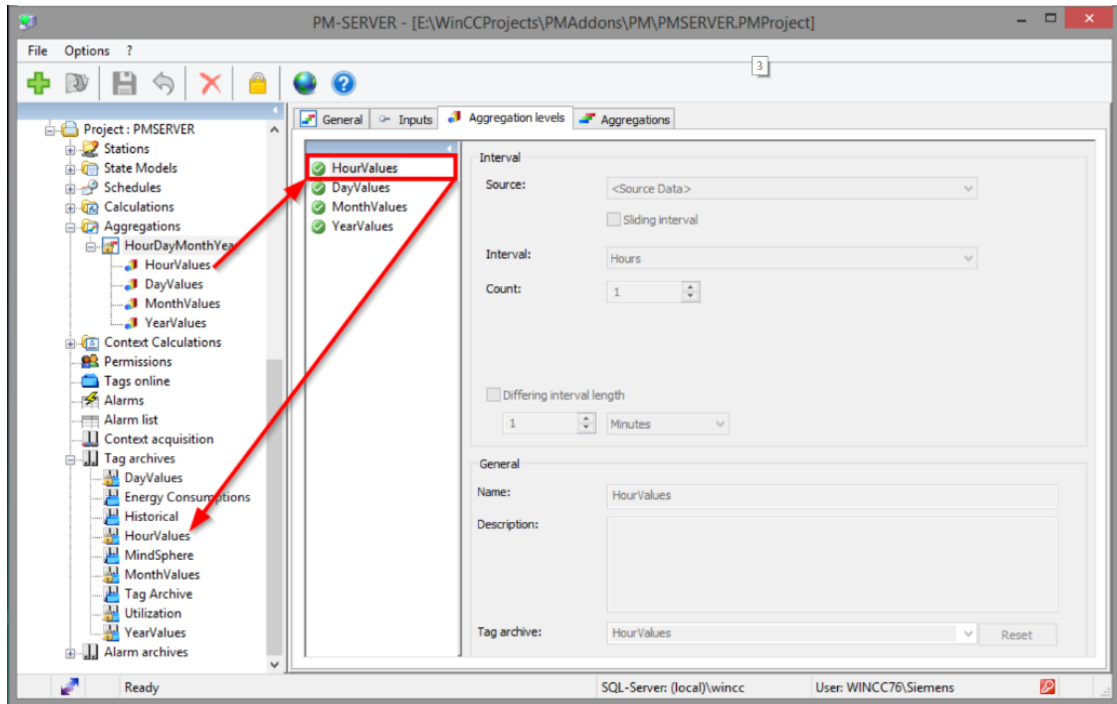
Kuva 39. Suorituskykyindikaattori laskenta Calculations-toiminnolla (Siemens 2022a, 221)

Kuvassa 40 esitellään kaava, jossa hälytyksien perusteella toteutetaan sähköpostin lähettäminen sekä tiedostoon kirjoitus.



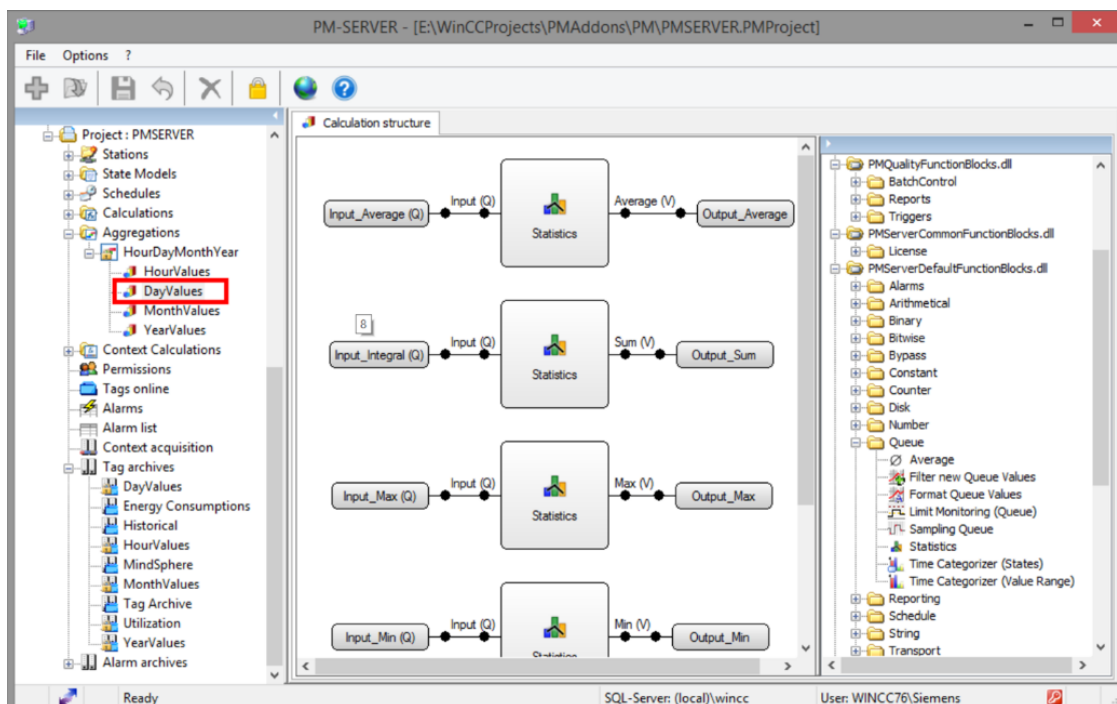
Kuva 40. Sähköpostin ja tiedoston muodostaminen Calculations-toiminnolla (Siemens 2022a, 108)

PM-SERVER-sovelluksen aggregointitoiminnolla on mahdollista yhdistää ja summata prosessiin liittyviä tunnuslukuja. Toimintaperiaate on vastaava kuin Calculations-toiminnossa. Toimintoon muodostetaan kaavoja ja tasoja. Tuloksia voidaan hyödyntää PM-lisäosissa tai muissa kaavoissa. (Siemens 2020a, 26; Siemens 2022a, 226–236.) Kuvassa 41 on luotu HourDayMonthYear-pää-taso, jonka alla ovat aikaväleille omat kaavansa. Tuloksista muodostuu tietokantaan omat kansionsa, joiden sisältöä voidaan välittää PM-lisäosalle, kuten muitakin arkistoitua tietoa.



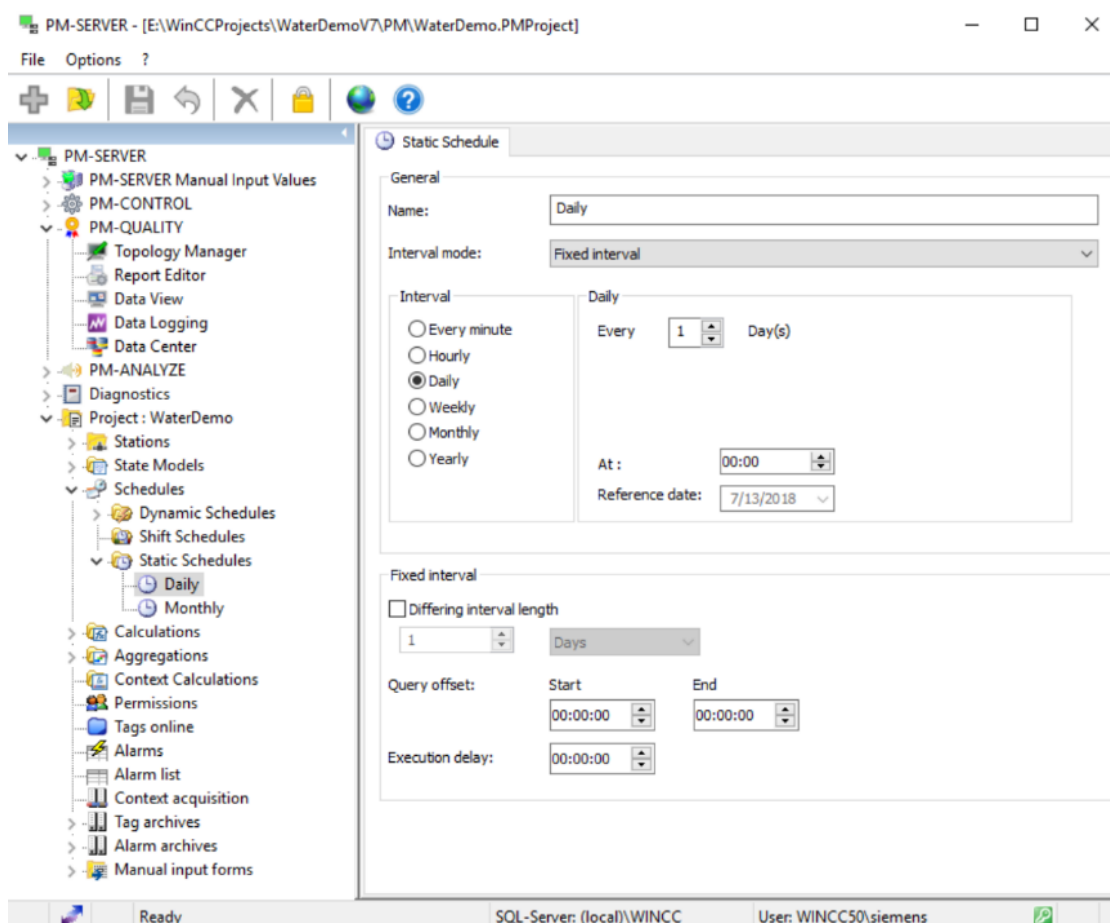
Kuva 41. Tason asetukset ja tuloksien tallentuminen tietokantaan (Siemens 2022a, 233)

Tasojen välisiä arvoja voidaan hyödyntää hiarkisesti eli tuntiperusteisesti summatuista keskiarvoista voidaan laskea päiväkeskiarvot ja päivien keskiarvoista kuukausikeskiarvoja jne. Kuvassa 42 on esitelty kaavojen määrittelyä DayValues-tasolla.



Kuva 42. Kaavoja DayValues-tasolla (Siemens 2022a, 235)

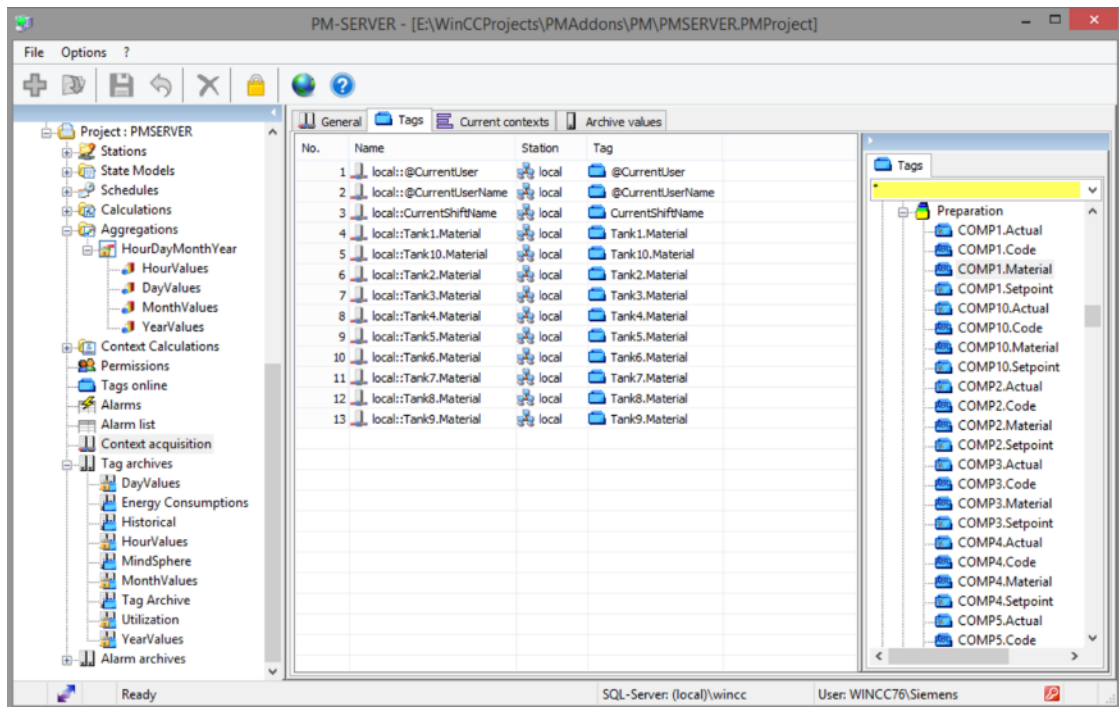
PM-SERVER-sovelluksessa luodaan Schedules-toiminnolla aikatauluja, joita voidaan hyödyntää eri toimintojen aktivointiin, kuten Calculations-toiminnon tapauksien. Aikatauluja on dynaamisia, työvuoroperusteisia ja staattisia. Dynaamisissa aikatauluissa jokin prosessitapahtuma voi toimia aktiivisena tai pysäyttävänä viestinä. Työvuoroperusteisissa aikatauluissa jaetaan työpäivä esimerkiksi aamu-, päivä- ja yövuoroihin. Staattinen aikataulu aktivoituu tietyllä aikavälillä, kuten päivittäin keskiyöllä tai maanantaisin kello 10. Jos aikatauluja käytetään raportointiin, voidaan aikataulun määrittelyyn sisällyttää myös liukuman, esimerkiksi päivittäin aktivoituva aikataulu, mutta se kattaa edelliset 36 tuntia. (Siemens 2018, 15; Siemens 2019a, 77; Siemens 2020a, 25.) Kuvassa 43 on aseteltuna päivittäinen aikataulu ilman liukumaa.



Kuva 43. Aikataulutoiminto PM-SERVER-sovelluksessa (Siemens 2018, 15)

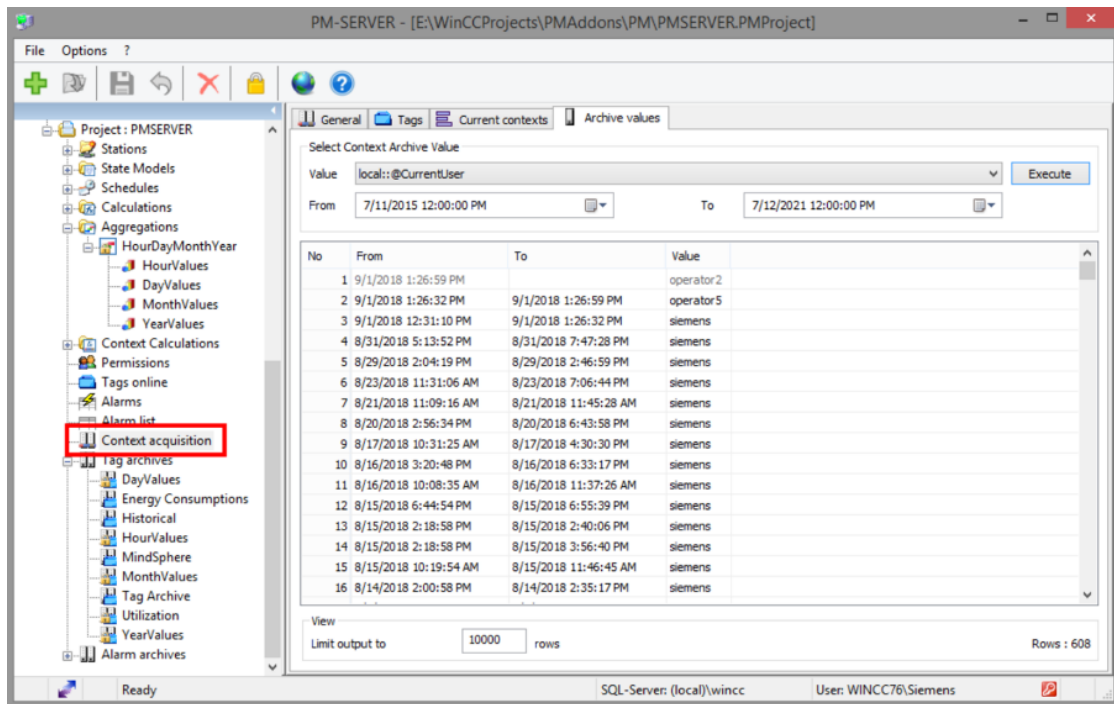
Vaihtoehtona dynaamisten aikataulujen määrittelylle on asiayhteyteen sidottu arkistointi (context acquisition). Arkistointitoiminto valvoo merkkijonon sisältävää tagia ja sen muutoksesta arkistoon tallentuvat tapahtuman alku ja loppu aikaleimoina sekä tekstikentän sisältö. Arkistoja voidaan hyödyntää PM-lisäosissa ajanjaksojen suodattimina. Arkistoinnin toimintaperiaate on, että tagin

arvonmuutos käynnistää tapahtuman ja tagin tyhjennys pysäyttää sen. Mikäli tagin arvo ei ole tyhjä ja sen arvo muuttuu, niin edellinen arkistointi pysähtyy ja uusi alkaa välittömästi. Tagi ei tarvitse olla varsinaisesti arkistointia varten luotu, vaan se voi olla mikä tahansa sovelluksessa esiintyvistä tageista. (Siemens 2022a, 237–241.) Kuvassa 44 on seurattavia tageja aseteltuna toimintoon.



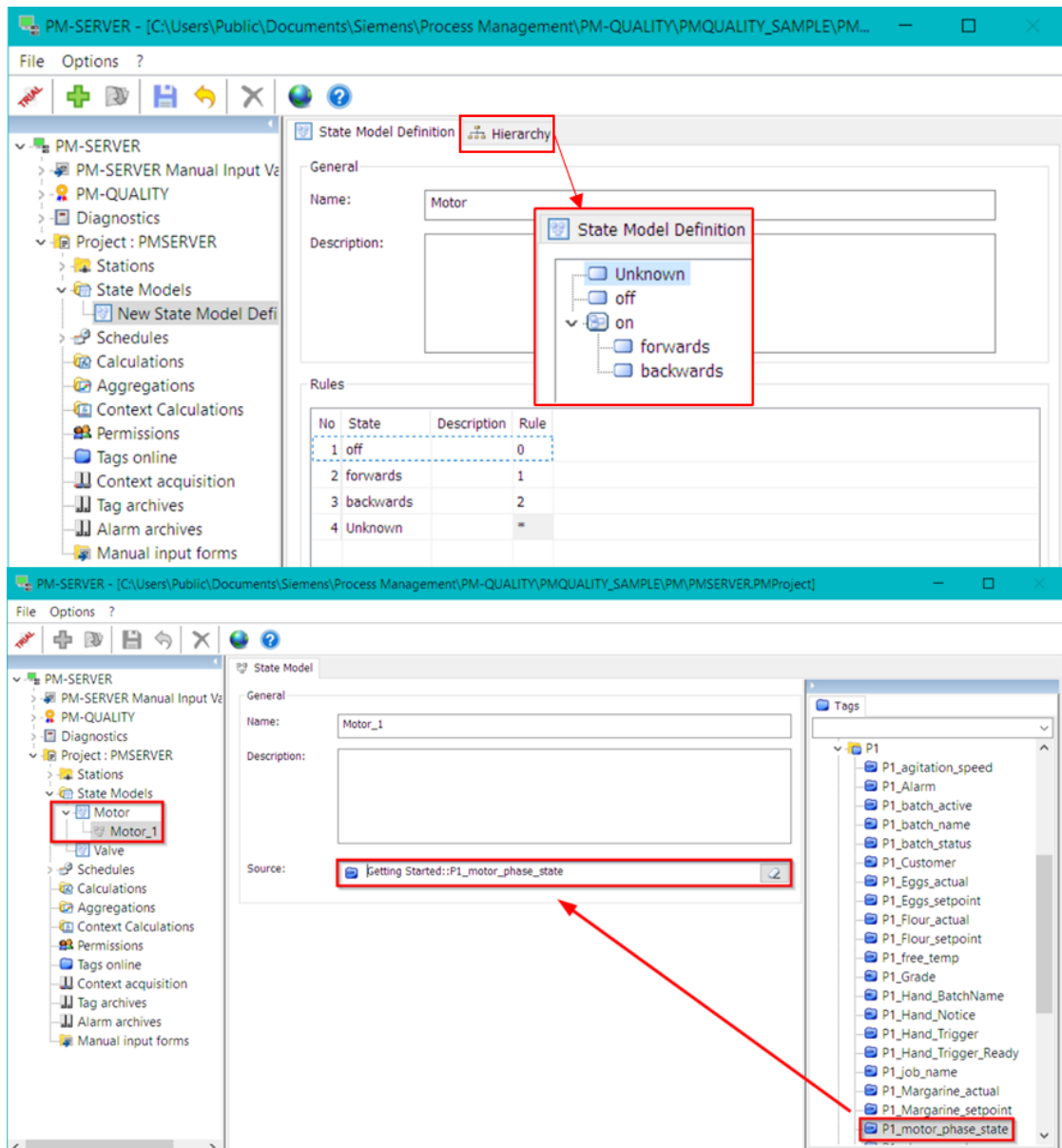
Kuva 44. Seurattavat tekstimuuttujat Context acquisition -toiminnossa (Siemens 2022a, 240)

Kuvassa 45 näkyy, miten CurrentUser-tagin arvonmuutokset ovat tallentuneet. Arvoina ovat tallentuneet käyttäjät siemens, operator5 ja operator2, joka on myös parhaillaan arkistoinnissa.



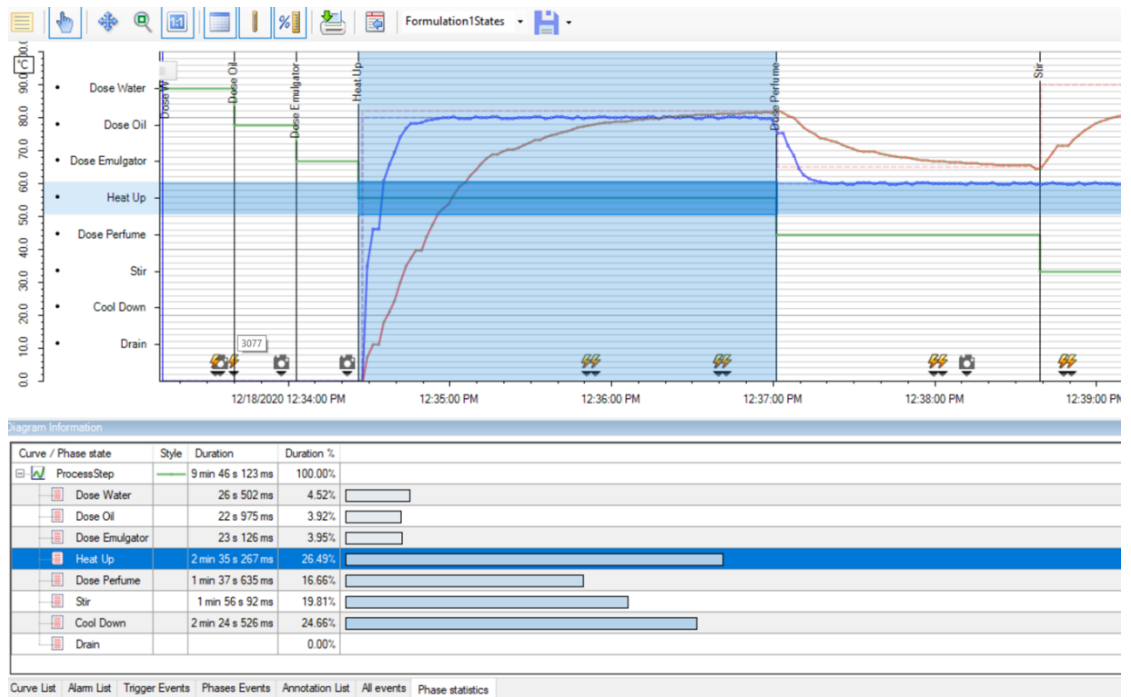
Kuva 45. Arvonmuutoksien arkistointia Context acquisition -toiminnossa (Siemens 2022a, 239)

PM-SERVER-sovelluksessa voidaan määrittellä valvottavan laitoksen tai laitteiden tilat (State Model). Laitoksen ja laitteiden tilaa voidaan hyödyntää PM-lisäosissa esimerkiksi jakamaan trendikäyränäyttöjä eri työvaiheisiin. Työvaiheet kootaan hierarkiaksi State Modelin asetuksiin ja ne aktivoidaan tagin arvonmuutoksilla. Toiminto koostuu State Modelista ja tapauksesta. (Siemens 2020b, 32–34; Siemens 2021b, 14.) Kuvassa 46 on esitelty laitteiden tilan määrittämistä. Siinä on luotu Motor-niminen State Model. Sen tilaa ja toimisuuntaa seurataan, kun määritetyn tagin arvo muuttuu 0:n, 1:n ja 2:n välillä. State Modelin alle on luotu Motor_1-niminen tapaus, joka on sidottu laitteen tilaa ja toimisuuntaa edustavaan tagiin.



Kuva 46. State Model -toiminto ja esimerkkitapaus Motor_1 (Siemens 2020b, 33–34)

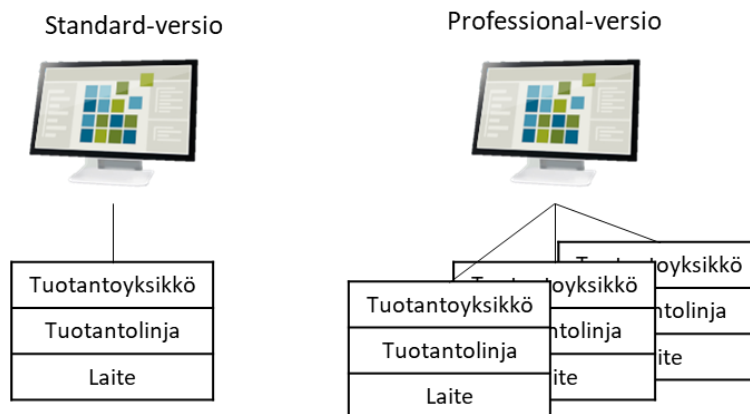
Kuvassa 47 on esimerkki PM-QUALITY-sovelluksen trendinäytöstä. Kuvassa ei esiinny aiempaa Motor-esimerkkitapausta, mutta siitä nähdään trendikäyrän jakautuminen vaiheisiin. Lisäksi trendikäyrän alapuolella nähdään, miten eri vaiheiden kestoa voidaan analysoida.



Kuva 47. State Model PM-QUALITY-sovelluksen trendinäytössä (Siemens 2022a, 73)

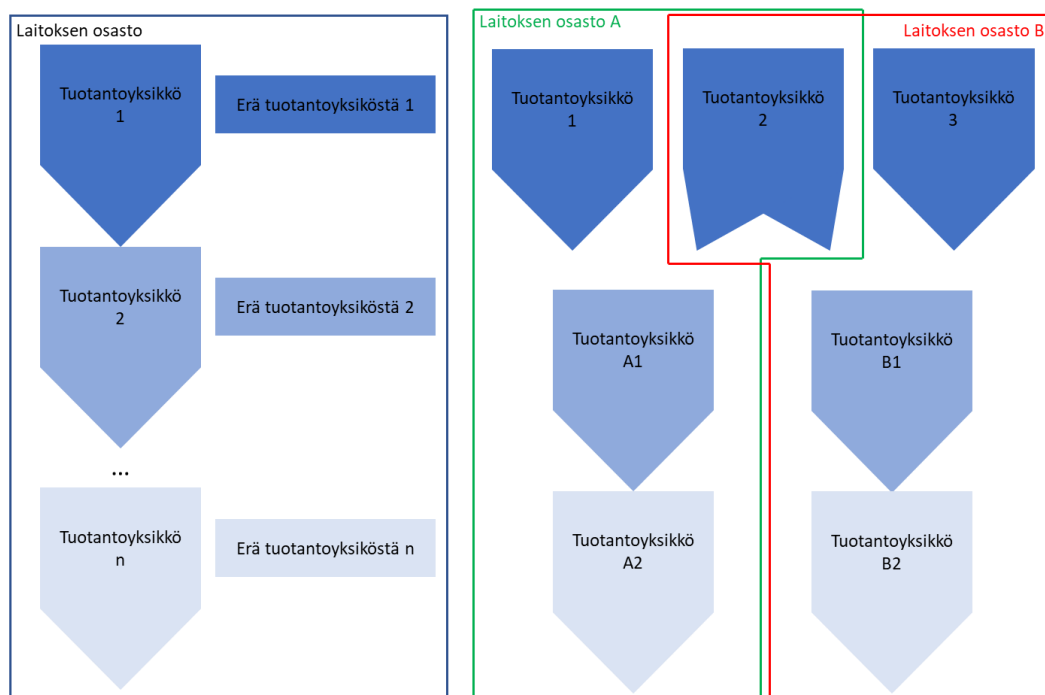
4.1.1 Tuotantoerätyökalu PM-QUALITY

PM-lisäosista PM-QUALITY on suunniteltu erä- ja kappaletavaraprosessien analysointiin ja raportointiin. Tähän PM-QUALITY-sovellus hyödyntää PM-SERVER-sovelluksen määrittelyjä ja toimintojen tuloksia, mutta täydentää sitä omilla tietosisällöillä, kuten tuotantoyksiköillä ja tuotantoerillä. Sovelluksesta on saatavilla Standard- ja Professional-versiot, joiden erona on määritettävien tuotantoyksiköiden lukumäärä. Standard-versioon voidaan määrittää yksi tuotantoyksikkö ja Professional-versiossa niitä voidaan määrittää sataan tuotantoyksikköön asti. (Siemens 2016, 4.) Kuvassa 48 on esitetty versioiden eroa.



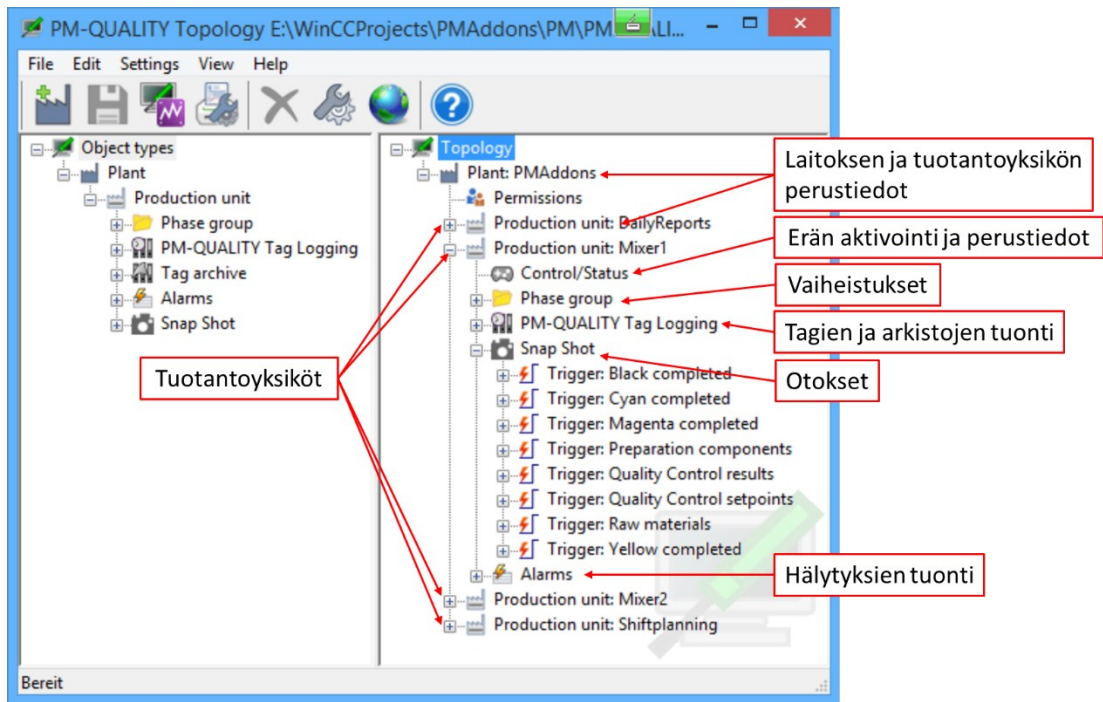
Kuva 48. PM-QUALITY Standard- ja Professional-versioiden ero (mukaillen Siemens 2016, 4)

Professional-versiossa on mahdollisuus määrittää useamman tuotantoyksikön yhdistelmiä, joista kustakin voidaan tuottaa eräkohtaista analyysijä sekä raporteja. Kuvassa 49 on havainnollistettu, miten Professional-versiossa voidaan ketjuttaa tuotantoyksiköitä tai tehdä niiden monimutkaisia yhdistelmiä.



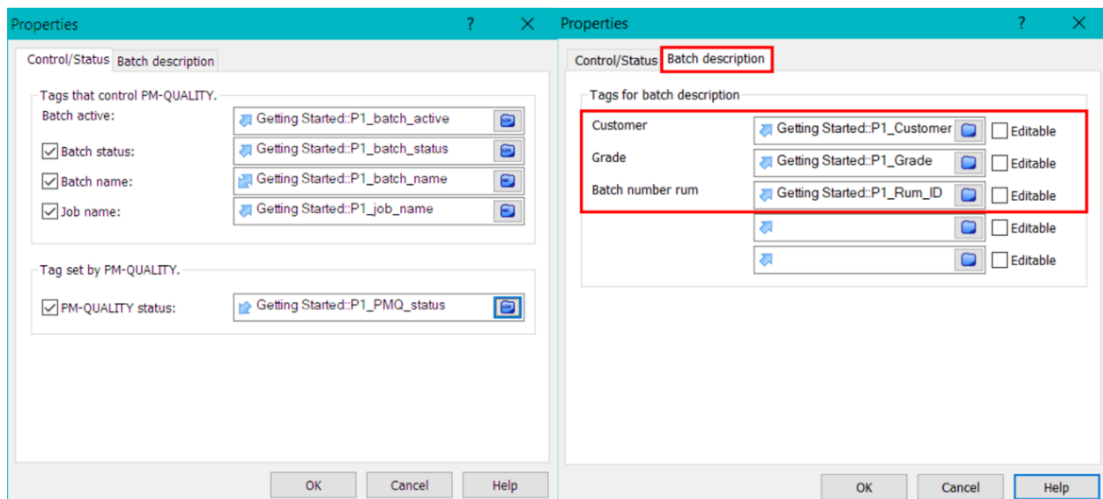
Kuva 49. Yhdistetyt tuotantoyksiköt PM-QUALITY Professional-versiossa (mukaillen Siemens 2016, 5)

PM-QUALITY-sovelluksen asetukset määritetään Topology Manager -toiminnossa. Asetushierarkian korkeimmalla tasolla ovat laitoksen ja tuotantoyksiköiden perustiedot. Tuotantoyksikön toimintaan ja sen eräkohtaisiin asetuksiin voidaan liittää erä- ja työnimiä sekä vapaasti määritettyjä rivitietoja. Koko sovelluksen toiminnan kannalta on minimivaatimuksena, että prosessissa on määriteltynä yksi tagi, joka indikoi tuotantoyksikön aktiivisuuden. (Siemens 2020b, 15, 47, 52–55.) Kuvassa 50 on Topology Manager -toiminnon pääikkuna ja korostettuna PM-QUALITY-sovelluksen eri toiminnot.



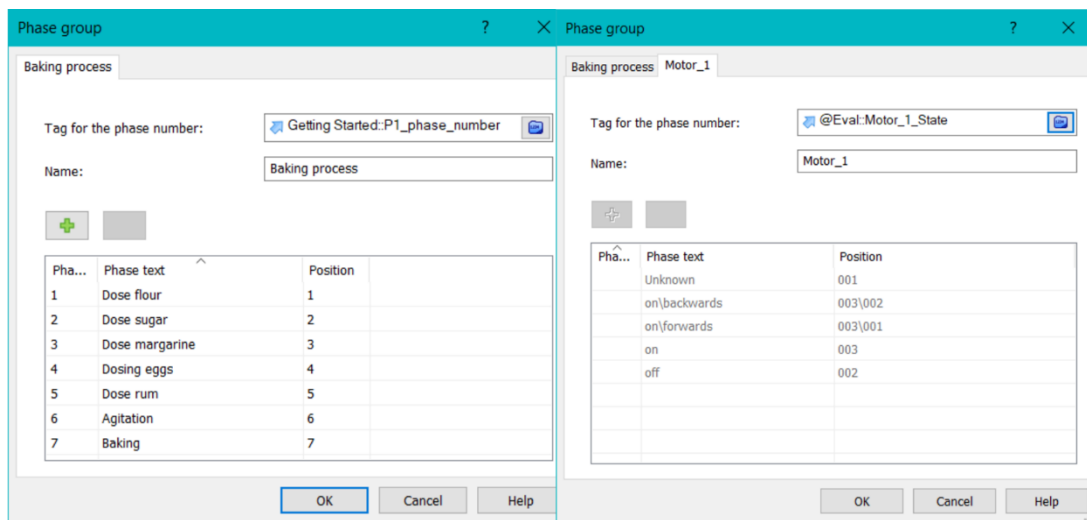
Kuva 50. PM-QUALITY-sovelluksen Topology Manager -toiminnon pääikkuna ja selitteet (Siemens 2016, 13)

Kuvassa 51 on tuotantoyksikön asetukset Control/Status ja Batch description. Control/Status-asetuksien ylimmällä Batch active -muuttujalla seurataan tuotantoyksikön aktiivisuustilaa ja seuraavalla kolmella muuttujalla voidaan erään liittää muita perustietoja. Alinta PM-QUALITY status -muuttujaa voidaan käyttää viestittämään WinCC-valvomosovellukselle, että yhteys PM-QUALITY-sovellukseen on kunnossa. Batch description -asetuksien rivien otsikot ovat itse määritettävissä ja niihin voidaan sitoa tekstimuuttujia. (Siemens 2020b, 54–55.)



Kuva 51. Aktiivisuustila ja perustiedot tuotantoyksikössä (Siemens 2020b, 54–55)

Tuotantoyksikön työvaiheet määritetään Phase-asetuksella. Vaiheiksi voidaan hyödyntää PM-SERVER-sovelluksessa määritettyjä State Model -vaiheistuksia tai ne voidaan luoda erikseen PM-QUALITY-sovelluksessa. Vaiheistuksia voidaan hyödyntää sekä analyseissä että raportoinnissa. Kuvassa 52 on vasemmalla luotu täysin uusi vaiheistus perustuen P1_phase_number-tagin arvoon ja oikealla hyödynnetään PM-SERVER-sovelluksessa luotua tapausta Motor_1. (Siemens 2020b, 55–58.)



Kuva 52. Vaiheistuksen luominen tuotantoyksikköön (Siemens 2020b, 57–58)

Tuotantoyksikön toiminnasta voidaan tallentaa otoksia Snap Shot -toiminnolla. Se tallentaa otoshetkellä tagien arvot, jotka ovat määritettyinä otoksen asetuksiin. Otoksen laukaisijaksi voidaan määrittää muutos tagin arvossa tai raja-arvon ylitys. Kun otos on suorittanut tehtävänsä, se muuttaa Finished message -asetuksen tagin arvon ykköseksi. Otokseen voidaan lisäksi sitoa käynnissä olevan erän nimi lukemalla se Batch name -asetuksen tagista. (Siemens 2016, 14; Siemens 2020b, 67–74.) Kuvassa 53 on määritetty Ingredients-niminen otos. Otoksen laukaisijana toimii P1_phase_number tagin arvon muuttuminen 6:ksi. Tehtävän suoritettuaan otos ilmoittaa sen tagin P1_trigger_ready arvolla.

Snap Shot

Batch start trigger | Batch end trigger | Batch start/end trigger | **Ingredients**

Trigger name: Notes:

Trigger event: Upon change If the value changes to

Values: Read cyclic Read once for each event Monitoring cycle: seconds

Finished message Batch name

Bit:

Kuva 53. Vaiheistuksen arvojen tallennus otoksella (Siemens 2020b, 68)

Otoksissa tallennettavat tagit asetetaan Edit values -painikkeen kautta. Tallennettavat tagit voidaan ryhmitellä useampaan kokonaisuuteen ja niille voidaan määrittää yksilölliset raja-arvot sekä yksiköt. Kuvassa 54 on luotu ryhmät Actual value ja Setpoint value. Niihin ovat sidottuina nimensä mukaisesti tapahtumaan soveltuvat prosessin mittaus- ja asetusravot.

Snap Shot

Trigger: Ingredients - [Eggs actual]

Actual value | Setpoint value

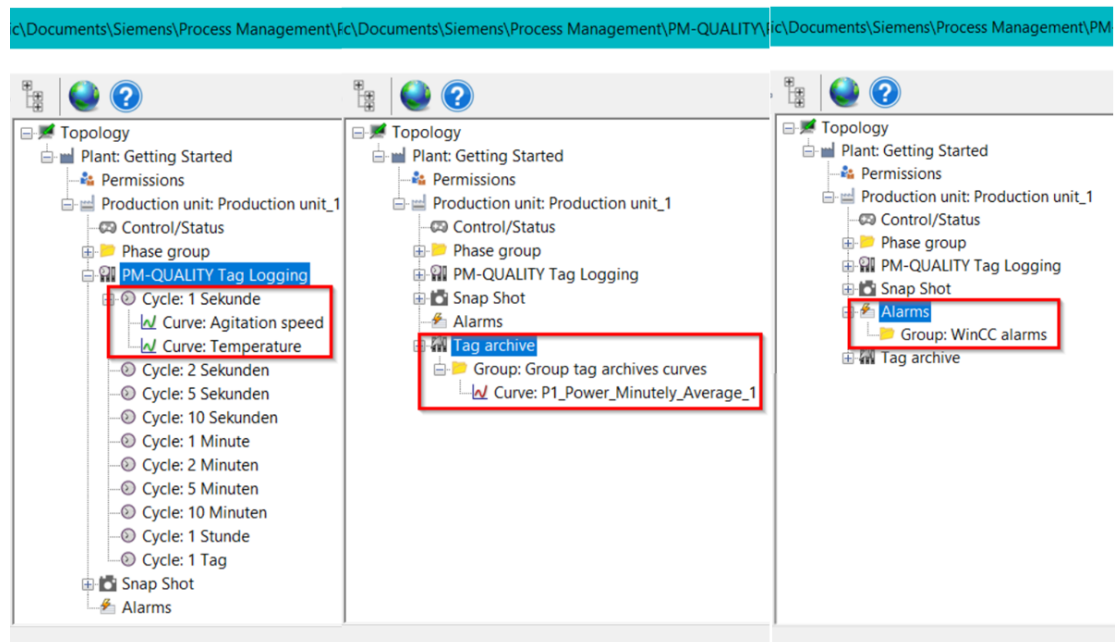
Group name:

| Nr. | Name of the value: | Tag | Lower limit | Upper limit | Decimal | Unit | Notes |
|-----|--------------------|---|-------------|-------------|---------|------|-------|
| 1 | Eggs actual | <input type="button" value="Getting Started::P1_Eggs_actual"/> | 0 | 3 | 0 | pcs. | |
| 2 | Flour actual | <input type="button" value="Getting Started::P1_Flour_actual"/> | 0 | 500 | 2 | g | |
| 3 | Margarine actual | <input type="button" value="Getting Started::P1_Margarine_actual"/> | 0 | 500 | 2 | g | |
| 4 | Sugar actual | <input type="button" value="Getting Started::P1_Sugar_actual"/> | 0 | 250 | 2 | g | |
| 5 | Rum actual | <input type="button" value="Getting Started::P1_Rum_actual"/> | 0 | 50 | 1 | ml | |

Kuva 54. Tallennettavien tagien asetukset otoksessa (Siemens 2020b, 69, 71)

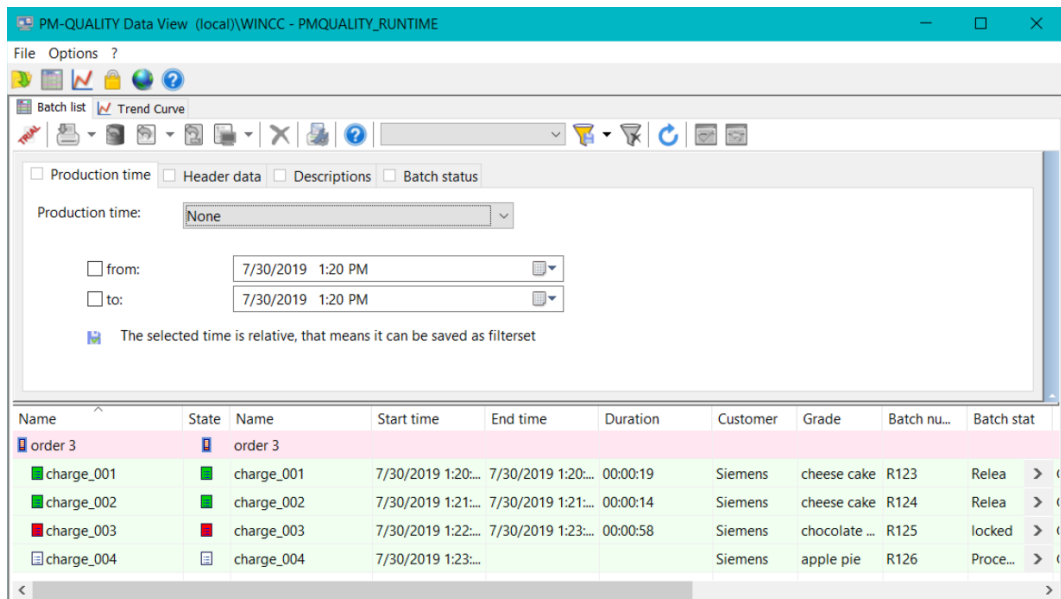
PM-QUALITY-sovelluksen käyttämät tiedot tuodaan PM-SERVER-sovellukselta. Tätä tiedonsiirtoa havainnollistettiin kuvassa 36 sivulla 60. Tagitietojen päivitys tapahtuu erikseen määritetyllä aikavälillä ja erikseen valituista tageista. Arkistojen kanssa valitaan vastaavasti tiedot tagi kohtaisesti. Hälytyksien kanssa käytetään suodatusta, jolla valitaan halutut hälytykset PM-SERVER-sovelluksen hälytysarkistosta. PM-QUALITY-sovellus toteuttaa myös omaa arkistointiaan käsittelemistään tiedoista, jos sen Data Logging -toiminto on aktivoituna. PM-QUALITY-sovelluksen arkistoima tieto voidaan määrittää vielä erikseen pitkäaikaisarkistoitavaksi HTML-muotoon, XML-formaattiin tai MDF-tiedostoksi, joka on Microsoft SQL Serverin käyttämä tiedostomuoto. Pitkäaikaisarkistoinnilla voidaan rajoittaa sovelluskohtaisen arkiston kokoa tietokantajärjestelmässä. Tämä toiminto määritetään kunkin tuotantoyksikön Export/Print-asetuksista. (Siemens 2020b, 59–66, 74–76, 78.)

Kuvassa 55 vasemmalla on luotu sekunnin sykleissä päivitettävät arvot tageista sekoittajan nopeus ja lämpötila. Keskellä kuvassa haetaan PM-SERVER-sovelluksessa laskettu tehon minuuttikeskiarvo. Oikealla kuvassa haetaan PM-SERVER-sovellukseen arkistoidut valvomosovelluksen hälytystiedot. Tietojen käyttö PM-QUALITY-sovelluksen puolella liittyy oleellisesti trendikäyriin, joten kohteista käytetään Curve-nimitystä.



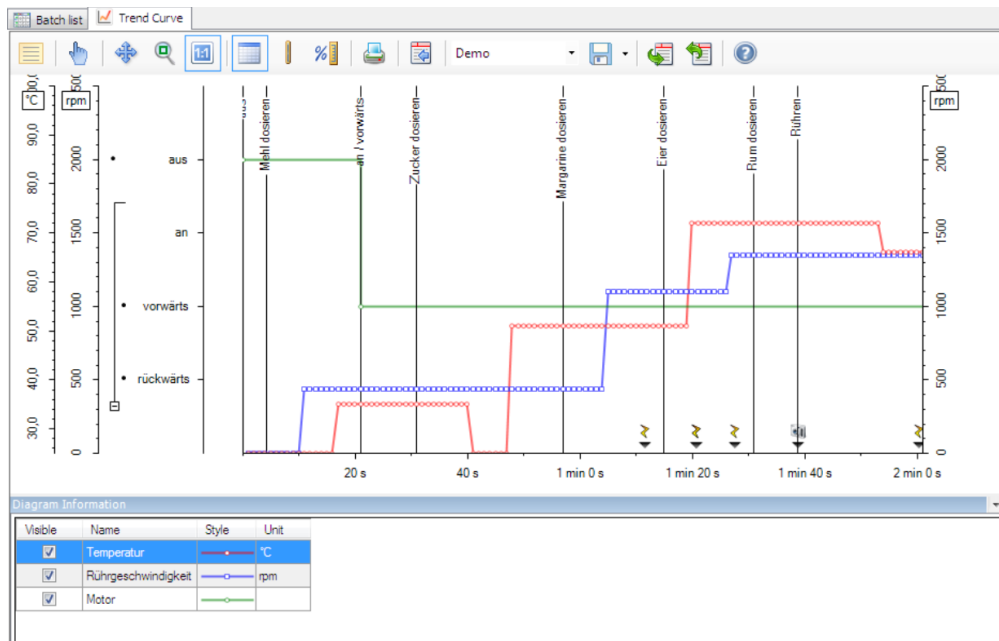
Kuva 55. Arkistointi PM-QUALITY-sovelluksessa (Siemens 2020b, 62, 65, 67)

PM-QUALITY-sovelluksen Data View -toiminnolla voidaan tarkastella tuotantoerien tietoja lista- ja trendinäkymissä. Näitä näkymätoimintoja kutsutaan Siemensin lähdemateriaalissa moduuleiksi. BatchTable-moduulin listanäkymässä voidaan tehdä suodatuksia erään sidottujen tietojen tai ajanjakson perusteella. (Siemens 2016, 11.) Kuvassa 56 on BatchTable-moduulin listanäkymä ja aikaväliperusteisen suodatuksen välilehti.



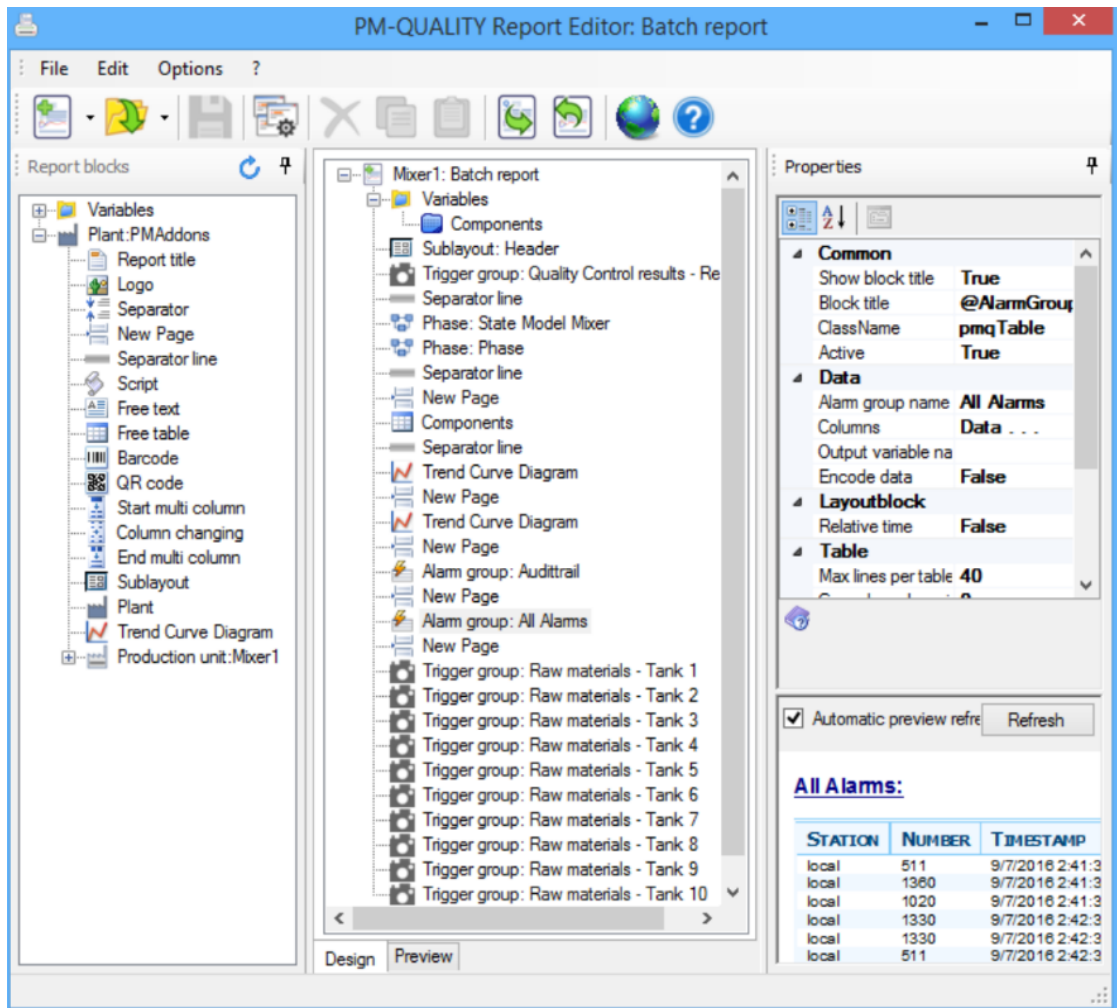
Kuva 56. BatchTable-moduuli ja aikavälisuodatus PM-QUALITY-sovelluksessa (Siemens 2020b, 81)

TrendCurve Control -moduulin voidaan analysoida yhden tai useamman erän prosessiarvoja trendinäkymässä. Näyttöjen rakenne ja esitettävät tiedot määritetään Data View -toiminnossa. Näyttöihin voidaan erikseen määrittää X- ja Y-akselit, esitettävät käyrät, ajanhetkiin sidotut otokset, tuotantoyksikön vaiheistukset ja hälytykset. Käyrät voivat muodostaa tagi- tai arkistotiedoista. Otokset voidaan näyttää myös pistemäisinä arvoina koordinaatistossa. (Siemens 2016, 12; Siemens 2020b, 81–90.) Kuvassa 57 on vaiheistuksen alkuun sidottu X-akseli. Y-akselilla näytetään lämpötilaa ja moottorin pyörimisnopeutta. X-akselilla näkyvät hälytykset salamamerkinnällä ja otokset kameramerkinnällä. Koordinaatistoon sijoitettuja otoksia ei ole kuvassa 57.



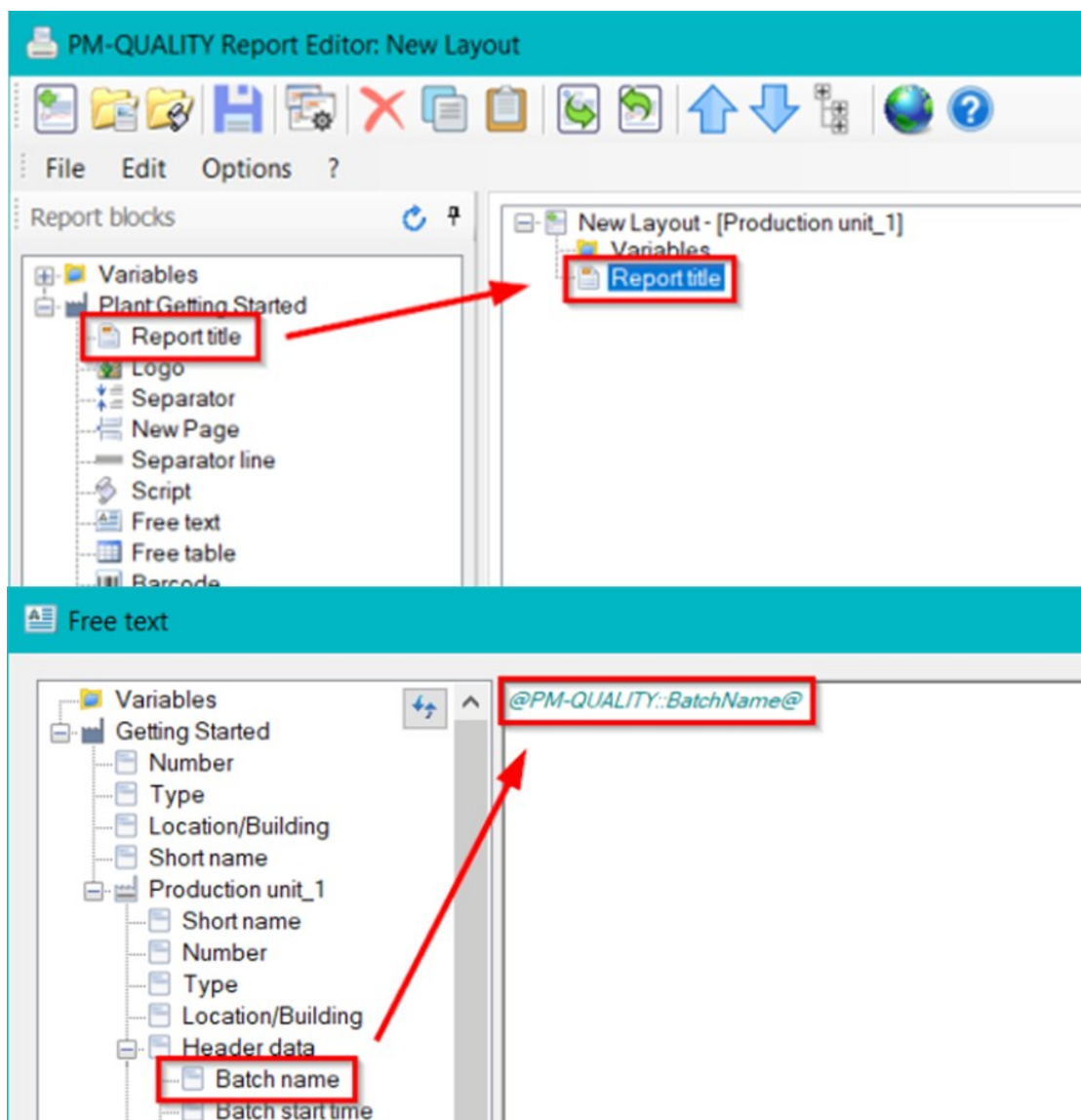
Kuva 57. Vaiheistuksia, hälytyksiä ja otoksia TrendCurve Control -moduulissa (Siemens 2020b, 90)

Raportoinnin kannalta oleellisin toiminto PM-QUALITY-sovelluksessa on Report Editor. Toiminnolla voidaan muodostaa yhden tai useamman tuotantoyksikköön kohdistettuja raportointipohjia. Raporttipohjat kootaan valmiista rakennosista, joita on tuotantoyksikön yleistietoa ja prosessitietoa esittävien osien lisäksi lukuisa määrä erityyppisiä rakenne-, laskenta-, formatointi-, muotoilu- osia. Prosessitietoja voidaan liittää raporttipohjalle joko taulukoina tai trendikäyrinä. Formatoidut osat voivat olla esimerkiksi QR- tai viivakoodeja. Formatointiosiin lukeutuvalla Scripts-rakennosilla voidaan Visual Basic -ohjelmointikielessä toteuttaa laskutoimituksia prosessitiedon pohjalta. (Siemens 2016, 19.) Kuvassa 58 on Report Editor -toiminnon perusnäky. Vasemmassa reunassa näkyvät käytettävät rakennosat ja keskelle koottuna raporttipohjan rakenne. Oikeassa alakulmassa näkyy pienennettynä raportin reaaliaikainen esikatselu.



Kuva 58. PM-QUALITY-sovelluksen Report Editor -toiminnon perusnäky (Siemens 2016, 20)

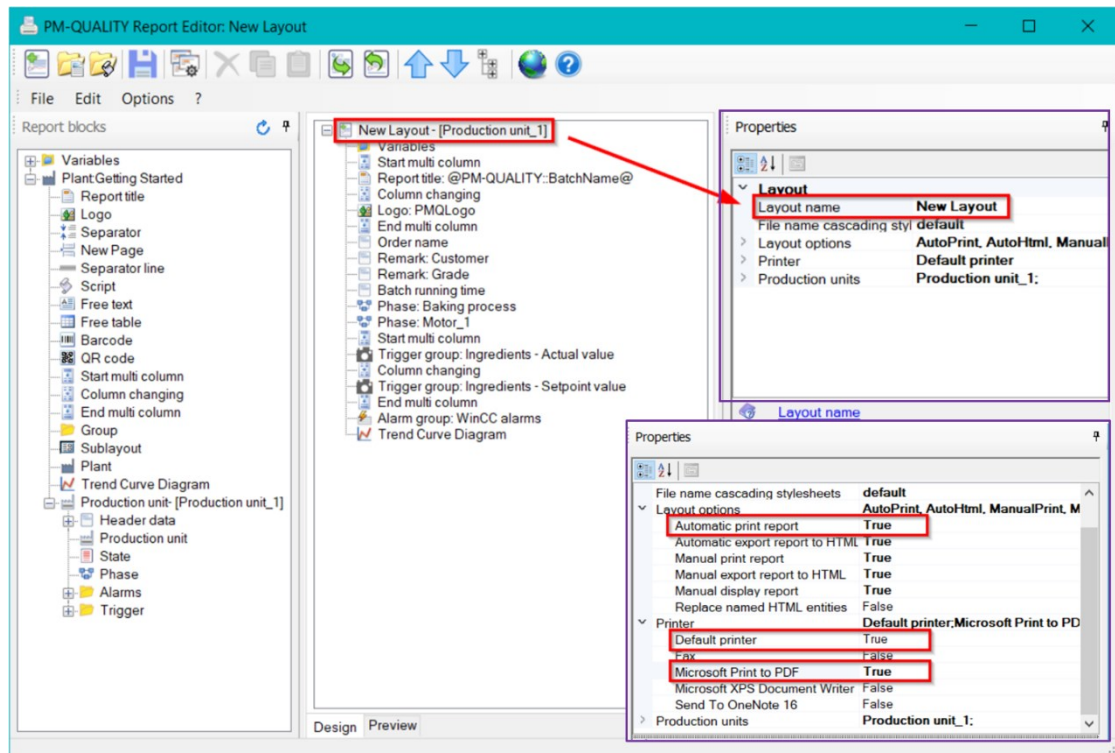
Jokaiselle rakenneosalle ovat omat osakohtaiset asetuksensa. Prosessitietojen ja hälytyksien taulukoista voidaan valita esitettävät sarakkeet, tekstikenttiin voidaan sitoa viittauksin eri prosessitietoja jne. (Siemens 2016, 19.) Kuvassa 59 esitellään, miten otsikkokenttään voidaan sitoa viittauksella erän nimi. Ylemmässä kuvassa on valittu Report title -rakenneosa ja alemmassa kuvassa on sisällöksi valittu Batch Name -tieto tuotantoyksiköstä Production unit_1.



Kuva 59. Viittauksen liittäminen tekstikenttään (Siemens 2020b, 92, 94)

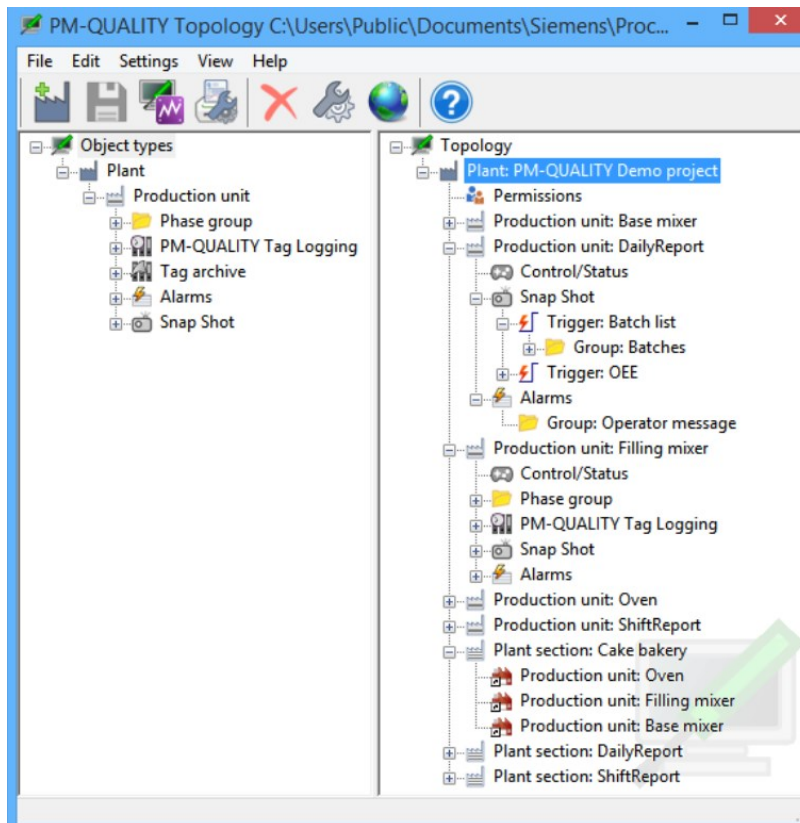
Raportointiajoon liittyvät asetukset määritellään kootun raporttipohjan ylimmästä eli päärakennosasta. Raportit voidaan määrittää tallennettavaksi HTML-muotoon tai tulostettavaksi. Tulostamisen kohdalla on huomionarvoista, että tulostimeksi voidaan valita myös Print to PDF -ohjelma. Lähdemateriaalissa ei ole mainintaa, miten raportointiajo vaiheittain tapahtuu, mutta sen voidaan olettaa toimivan tuotantoyksikön aktiivisuustilan ja pitkäaikaisarkistoinnin määryksien mukaisesti. Kun tuotantoyksikkö on ollut aktiivisena Batch Active -asetuksen mukaisesti, raportit siirtyvät Report Editor -toiminnosta tulostus- tai tallennusjonoon päärakennososan asetuksien mukaisesti. Varsinainen raportointiajo tapahtuu vasta pitkäaikaisarkistoinnin Export to HTML- tai Print-välilehtien määryksien mukaisesti. (Siemens 2020b, 105–106.) Kuvassa 60 on

esitelty raporttipohjan päärakenteosan asetuksia sekä tulostuksen asetusvaihtoehtoja.



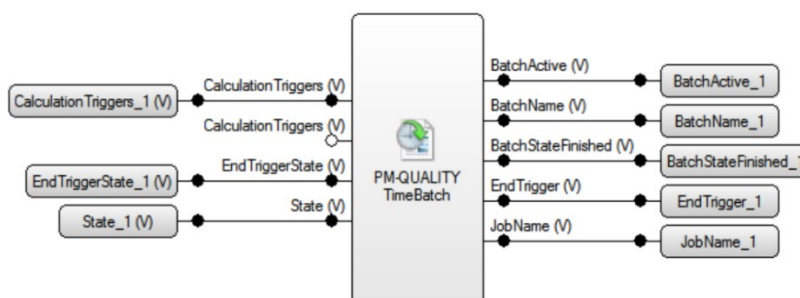
Kuva 60. Raporttipohjan automaattisen tulostuksen asetukset (Siemens 2020b, 105–106)

Raportointi on mahdollista sitoa myös ajanjaksoihin. Tämä vaatii, että tuotantoyksiköiden aktivoijana toimiikin aikaan sidottu tagi. Tätä menetelmää ei ole dokumentoitu Siemensin materiaaleissa, mutta kuvasta 61 nähdään, miten tämän kaltaisia tuotantoyksiköitä on muodostettu. Tämän menetelmän käyttö vaatii PM-QUALITY-sovelluksesta Professional-version, jolla voidaan määrittää useampia tuotantoyksiköitä.



Kuva 61. Aikaperusteisen raportoinnin tuotantoyksiköt PM-QUALITY-sovelluksessa (Siemens 2016, 10)

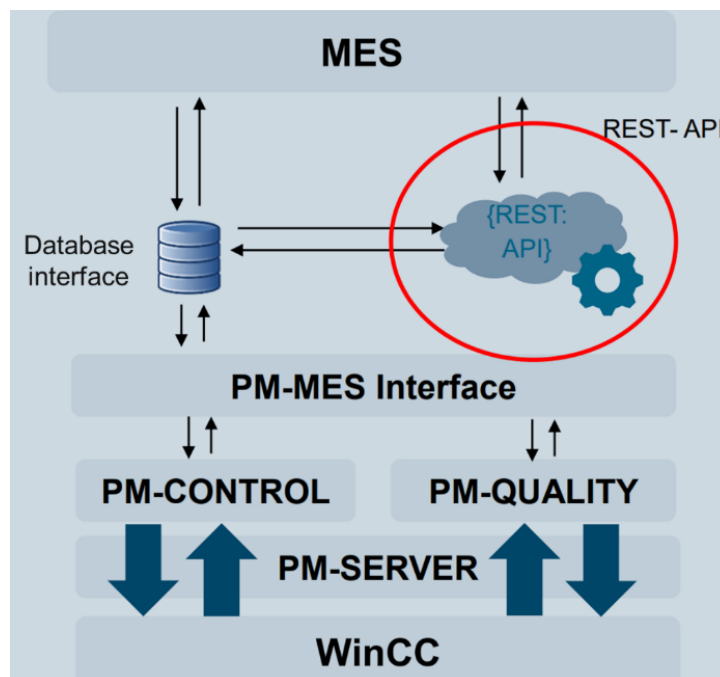
PM-QUALITY laajentaa PM-SERVER-sovelluksen Calculations-toiminnossa valittavien ohjelmointilohkojen määrää. TimeBatch-lohkolla voidaan toteuttaa aikaan sidottuja raportteja. Cyclic Schedule Trigger -lohkolla tehdään otoksia määrätyn aikavälein. TimeBatch Calculation -lohkolla voidaan liittää laskennan tuloksia aika- ja vuoroperusteisiin raportteihin. DeferBatchEnd-lohko aktivoi, että erän lopusta tehdään otos. Dynamic Schedule Trigger -lohko liittää laskennan tuloksista otoksen erän loppuun. Näillä lohkoilla toteutettavien raporttien formaatista ei ole tietoa lähdemateriaalissa. (Siemens 2016, 16.) Kuvassa 62 on TimeBatch-lohko.



Kuva 62. TimeBatch-ohjelmointilohko PM-SERVER-sovelluksen Calculations-toiminnossa (Siemens 2016, 16)

Report Editor -toiminnon ja ohjelmointilohkojen lisäksi PM-QUALITY-sovelluksen mukana tulevalla Microsoft Excel -liitännäisellä, voidaan toteuttaa raportointia.

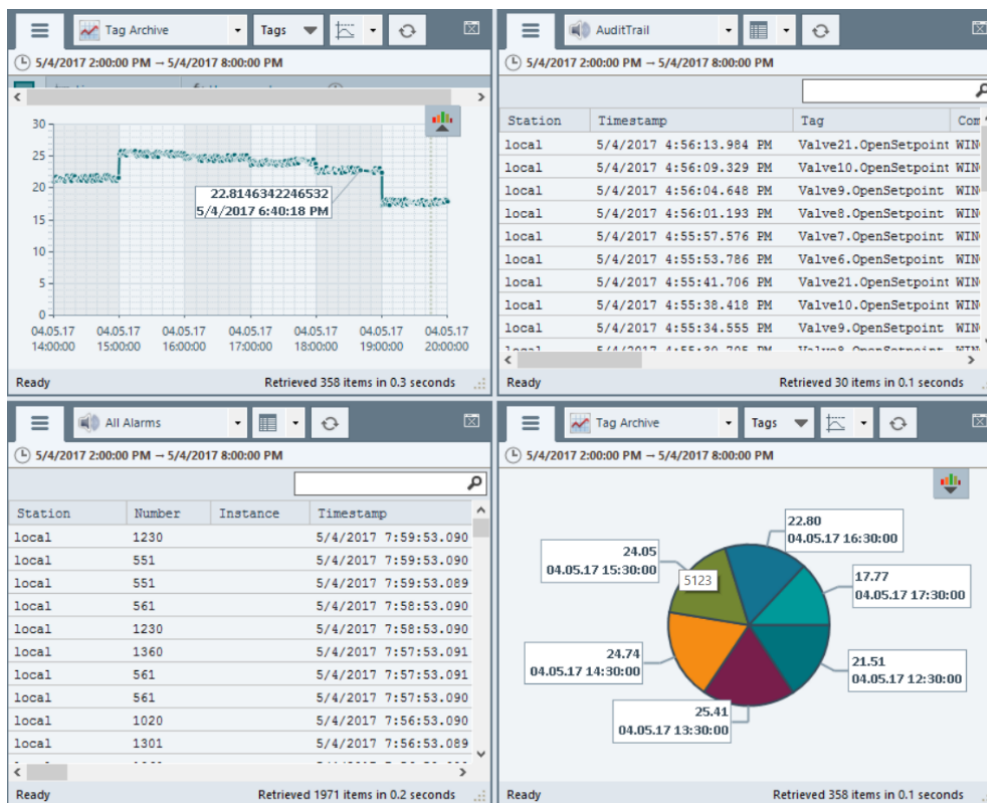
PM-QUALITY-sovellus tukee Siemensin PM-MES-rajapintaa, joka on ensisijaisesti suunniteltu automaatiojärjestelmän ja tuotannonohjausjärjestelmän väliseen integrointiin yhdessä PM-CONTROL-sovelluksen kanssa. Rajapinta on erikseen lisensoitava tuote ja kuuluu Siemensin Process Management -lisäosakokonaisuuteen. Kuvassa 63 esitellään rajapinnan toimintaperiaatetta. Rajapinta on kaksisuuntainen ja tiedonsiirto voidaan toteuttaa Microsoft SQL Server tietokannan välityksellä tai REST-rajapinta-arkkitehtuurilla. Tiedonsiirtoon voidaan käyttää myös OPC UA -rajapintaa, joka ei esiinny kuvassa. Rajapinnan välityksellä voidaan siirtää PM-QUALITY-sovelluksesta hälytyksiä ja poikkeamatietoja, raportteja, trendikäyriä sekä Snap Shot -toiminnon tallentamia otoksia. (Siemens 2022a, 119–123, 132–133.)



Kuva 63. PM-MES-rajapinnan infrastruktuuri (Siemens 2020e, 5)

4.1.2 Prosessitietotyökalu PM-ANALYZE

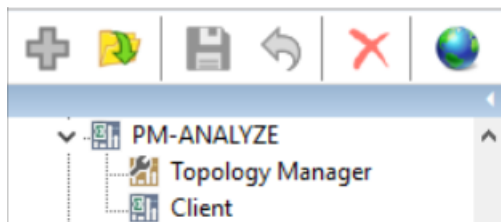
PM-ANALYZE-sovellus on suunniteltu prosessitiedon analysointiin ohjelmassa määritettävien työtilojen avulla. Analyysit perustuvat PM-SERVER-sovelluksen arkistoihin ja sen eri toiminnoilla käsiteltyihin tietoihin ja hälytyksiin. PM-ANALYZE skaalautuu täten PM-SERVER-sovelluksen mukaisesti. Työtiloissa voidaan esittää tietoja listoina, taulukoina, kaavioina ja trendikäyriä. Tietoja voidaan suodattaa luotujen aikataulujen ja vaiheistuksien avulla tai Context acquisition -toiminolla arkistoituihin asiansyhteyksiin pohjautuen. Mikäli PM-ANALYZE-sovelluksen yhteyteen on asennettu myös PM-QUALITY, voidaan suodatuksia ja analysointeja tehdä tuotantoerien mukaisesti. Työtilat voidaan viedä myös esitettäväksi WinCC-käyttöliittymään. (Siemens 2020a, 4, 24.) Kuvassa 64 on esitelty erilaisia työtiloja.



Kuva 64. Työtiloja PM-ANALYZE-sovelluksessa (Siemens 2022a, 206)

Sovelluksesta on saatavilla asennuspaketit Server ja Client. Server-asennuspaketti sisältää PM-ANALYZE-sovelluksen täydellisenä sekä PM-SERVER-sovelluksen ja PM-AGENT-rajapinnan palvelut. Client-asennuspaketti sisältää vain analysointiin ja työtilojen muodostamiseen soveltuvat osat PM-ANALYZE-sovelluksesta. (Siemens 2020a, 6–7.)

Server-asennuspaketilla hallittavat määrytykset ovat varsin suppeat. Topology Manager -toiminnossa määritetään suodatuskriteerit, joita voidaan hyödyntää Client-sovelluksessa hälytyksien analysointiin. Client-sovelluksien käyttöoikeudet eri arkistoihin ja sallitut analyysimenetelmät määritellään Server-asennuspaketissa tulevan hallintaikkunan Client-toiminnossa. (Siemens 2020a, 14, 27.) Kuvassa 65 on PM-ANALYZE-sovelluksen Server-puolen pääikkuna ja eri toiminnot.



Kuva 65. Server-puolen pääikkuna PM-ANALYZE-sovelluksessa (Siemens 2020a, 24)

PM-ANALYZE-sovelluksessa ei ole vastaavaa raportointitoimintoa kuin PM-QUALITY-sovelluksen Report Editor. Työtiloista on mahdollisuus tallentaa analyysien tuloksia CSV- ja XML-formaateissa tai viedä ne suoraan Excel-taulukkolaskentaohjelmaan, mutta ainoa raportointiin liittyvä toiminnallisuus on sovelluksen mukana toimitettava Microsoft Excel -liitännäinen. (Siemens 2020a, 23, 28.)

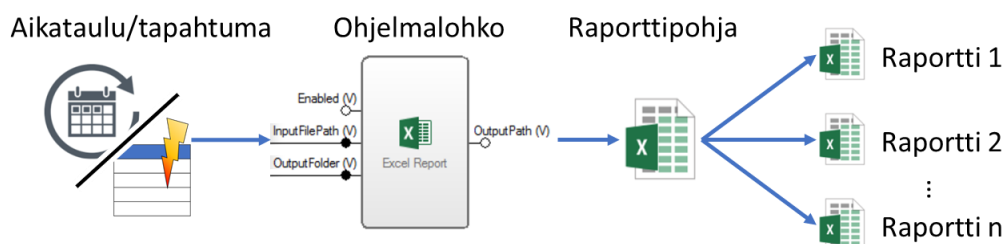
PM-ANALYZE-sovelluksen mukana toimitetaan REST-rajapinta-arkkitehtuuria hyödyntävä PM-API-rajapinta. REST-rajapinta-arkkitehtuuri on vastaava kuin OPC UA yhteydessä mainittu SOAP-rajapinta-arkkitehtuuri. Se toimii TCP/IP-protokollan päällä ja tukee tiedon välittämistä HTML-, XML- ja JSON-formaateissa. PM-API-rajapinnan kautta voidaan hyödyntää PM-SERVER-sovelluksen arkistoitimia prosessitietoja ja hälytyksiä. Tiedot noudetaan `"/api/pmserver/..."` URL-osoitteen takaa. Taulukossa 1 on esitelty osoitteiden logiikkaa. Niitä on lyhennetty jättämällä em. toistuva osuus pois. (Siemens 2020a, 29; Siemens 2022a, 197.)

Taulukko 1. PM-API URL-osoitteet ja selitteet (mukaillen Siemens 2020a, 29)

| URL-osoite | Määritelmä |
|---|----------------------------------|
| .../tagarchives | Listaus prosessitietoarkistoista |
| .../tagarchives/{archiveId} | Prosessitietoarkiston tiedot |
| .../tagarchives/{archiveId}/tags | Listaus arkiston tageista |
| .../tagarchives/{archiveId}/tags/{tagId} | Tagin tiedot |
| .../tagarchives/{archiveId}/tags/{tagId}/values | Tagin arvot |
| .../tagarchives/{archiveId}/tags/{tagId}/aggregatedValues | Aggregoidut tagin arvot |
| .../alarmarchives/ | Listaus hälytysarkistoista |
| .../alarmarchives/{archiveId} | Hälytysarkiston tiedot |
| .../alarmarchives/{archiveId}/blocks | Hälytyksien viestit |
| .../alarmarchives/{archiveId}/alarms | Hälytykset |

4.1.3 PM-lisäosien Microsoft Excel -liitännäinen

PM-ANALYZE- ja PM-QUALITY-sovelluksien mukana toimitetaan Microsoft Excel-liitännäinen, joka mahdollistaa raporttipohjien toteuttamisen Excel-taulukkolaskentapohjalle. Varsinainen raportointiajo tapahtuu Excel Report -ohjelmointilohkolla PM-SERVER-sovelluksen Calculations-toiminnossa. Ohjelmointilohkoon liitetään käytettävä raportointipohja ja tiedot tallentuvat pohjalle raportointiajon yhteydessä. Raportointiajon aktivoijana voidaan hyödyntää PM-SERVER-sovelluksessa luotuja aikatauluja tai muutosta prosessi-arvoissa, kuten muissakin Calculations-toiminnolla luoduissa ohjelmissa. Huomioitavaa on, että raportit voidaan luoda eri ajanhetkellä verrattuna raportin tietosisällön ajanhetkeen. Lähdemateriaalin pohjalta ei ole yksiselitteisesti tulkittavissa, miten ohjelmointilohko ja raportointipohja tarkalleen yhdessä käsittelevät tietosisällön aikaleimoja ja näissä PM-SERVER-sovelluksen aikatauluja ja asetettua liukumaa. (Siemens 2019a, 72; Siemens 2022a, 251–253, 257.) Kuvassa 66 on havainnollistettu tätä toimintaa PM-SERVER-sovelluksessa.



Kuva 66. Raportoinnin toimintaperiaate Excel-liitännäisen kanssa (mukaillen Siemens 2019a, 72)

Excel Report -ohjelmointilohkon asetuksista määritetään, tallennetaanko raportti XLSX-, CSV-, HTML- vai PDF-formaatissa. Lisäämällä viestin lähettämiseen soveltuvia ohjelmointilohkoja, voidaan raportti välittää suoraan esimerkiksi sähköpostilla raportointiajon yhteydessä. (Siemens 2022a, 251, 262.)

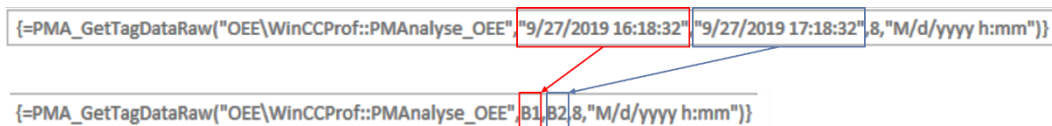
Kuvassa 67 näkyy Excel Report -ohjelmointilohkon asetuksia. Raporttipohja voidaan antaa syöttötietona tai tiedostopolulla. Tiedoston nimeen voidaan muotoilla aikatauluun sidottuja tietoja, kuten raportoinnin suoritus aika, raportin tietosisällön aloitus- ja lopetus aika tai työvuoro. Ohjelmointilohkolle voidaan antaa syöttötietoina aikaleimoja, mutta lähdemateriaali ei tarkenna mihin niitä käytetään. Mikäli ne kirjoitetaan Custom Cells -asetuksien kautta määritettyihin soluihin raporttipohjalle, olisi niitä mahdollista hyödyntää raportoinnin aloitus- ja päättöaikoina. Toinen vaihtoehto on, että ohjelmointilohkoon syötettyjä tai soluista luettuja aikaleimoja hyödynnetään vain raporttitiedoston nimeämisessä. (Siemens 2019a, 78; Siemens 2022a, 258–261.)

The image shows a 'Configuration' dialog box with the following settings:

- Input:** Path: From Pin, Constant. Path: C:\Users\Admin\Desktop\Excel\template\template.xlsx
- Custom Cells:** Worksheet: table1. Start Time Cell: B1 (checkbox: Get start time from pin). End Time Cell: B2 (checkbox: Get end time from pin). Custom Text Cell: (empty), Content: (empty)
- Output:** Filename: [endlyyyy-MM-dd HH:mm:ss] (tooltip: Interval Start, Interval End, Execution Time, Shift Name). Example: 2019-09-27 15_33_24. Folder: From Pin, Constant. Folder: C:\Users\Admin\Desktop\Excel\Report. File Format: Excel

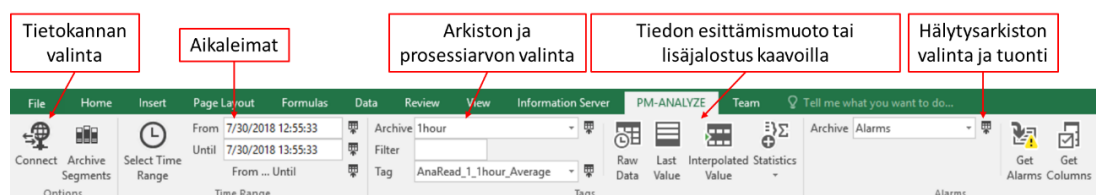
Kuva 67. Excel-ohjelmointilohkon asetukset (Siemens 2019a, 78)

Raporttipohjien toiminta perustuu kaavoihin, joissa tietosisältönä voidaan hyödyntää PM-SERVER- ja PM-QUALITY-sovelluksien arkistoitimia prosessitietoja, aggregoinnin ja laskennan tuloksia sekä hälytyksiä. Liitännäisen työkalurivi auttaa kaavojen muodostamisessa, mutta tarvittaessa niiden sisältöä voidaan muokata kuten minkä tahansa Excel-kaavojen. Lähdemateriaali ei tarkenna, mitkä ovat työkalun käyttämät yhteys- ja tiedonsiirtomenetelmät sen hakiessa tietoa arkistoista. (Siemens 2019a, 76, 78; Siemens 2022a, 254.) Kuvassa 68 näkyy, miten valittu arkisto sisältyy työkalun muodostamaan kaavaan. Arkistoksi on valittu Siemensin (2019a) esimerkissä laskettu tuotantolinjan tehokkuuden tulosarkisto. Lisäksi kuvassa 68 on havainnollistettu, miten kiinteät aikaleimat ovat vaihdettu soluviittauksiksi. B1- ja B2-soluihin kirjataan dynaamisesti raportoinnin alkamis- ja päättymisajat ja voidaan hyödyntää arkistotietoja hakevassa kaavassa.



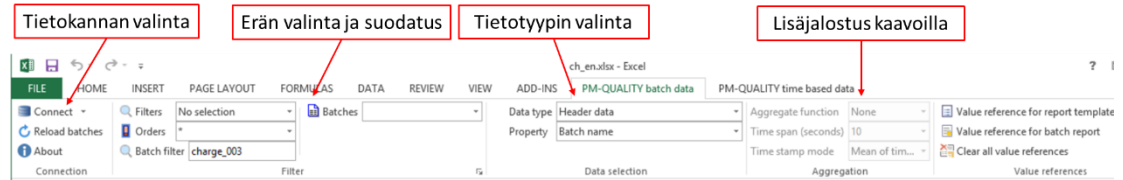
Kuva 68. Liitännäisen muodostamien kaavojen muokkaaminen (Siemens 2019a, 76)

Työkalurivin rakenne ja ominaisuudet riippuvat PM-lisäosasta, jonka mukana se toimitetaan. PM-ANALYZE-sovelluksen liitännäisen työkalurivillä voidaan toteuttaa ajanjaksoon perustuvia raportteja. PM-QUALITY-sovelluksen liitännäinen sisältää työkalurivit sekä eräkohtaiseen että ajanjaksoon perustuvien raporttien luomiselle. Eräkohtaisiin raportteihin voidaan valita tietotyyppinä eräkohtaisia tietoja, vaiheistuksia, otoksia ja trendikäyriä. (Siemens 2016, 31; Siemens 2020a, 28) Kuvassa 69 on PM-ANALYZE-sovelluksen Excel-liitännäisen työkalurivi selitteineen. Aikaleiman osalta lähdemateriaalissa ei ole tarkennettu, miten siinä voitaisiin hyödyntää muuta kuin kuvan mukaista absoluuttisia aikaa.



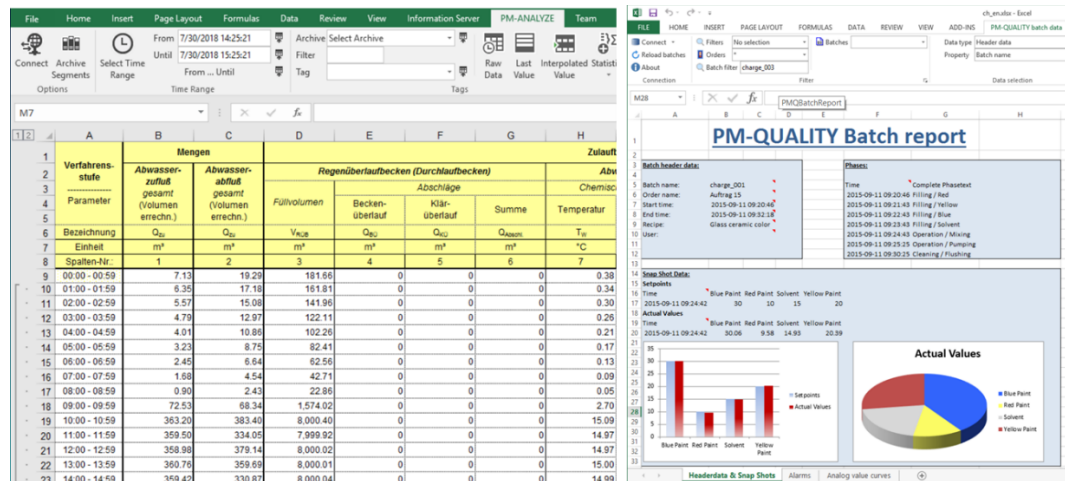
Kuva 69. PM-ANALYZE:n Excel-liitännäisen työkalurivi (mukaillen Siemens 2020a, 28.)

Kuvassa 70 on PM-QUALITY-sovelluksen Excel-liitännäisen eräkohtaisen raportoinnin työkalupalkki selitteineen. Ajanjaksoon perustuvan työkalurivin välilehtipainike näkyy valitun välilehden vieressä.



Kuva 70. PM-QUALITY:n Excel-liitännäisen työkalurivi eräkohtaiseen raportointiin (mukaillen Siemens 2016, 31)

Liitännäisten avulla on mahdollista toteuttaa raporttipohjia, jotka sisältävät yksinkertaista taulukoitua prosessitietoa, mutta myös informatiivisempia raportteja, joissa hyödynnetään kuvaajia, diagrammeja sekä muita Excelin ominaisuuksia (Siemens 2022a, 256). Kuvassa 71 vasemmalla on taulukkopohjainen raakatietoa sisältävä raportti ja oikealla on kuvaajilla täydennetty eräraportti, joka jakaantuu useammalle työkirjan välilehdelle.



Kuva 71. Esimerkkiraportteja PM-lisäosan Excel-liitännäisellä (Siemens 2016, 31; Siemens 2022a, 256)

4.2 Raportoinnin toteuttaminen rajapintojen välityksellä

Raportointi on mahdollista toteuttaa myös rajapintojen avulla WinCC Unified -valvomosovelluksesta. Sen molemmat arkistointimenetelmät ovat hyödynnettävissä ODBC-rajapinnan avulla ja lisäksi valvomosovellus tukee kahta Siemensin omaa rajapintaa: Siemensin Runtime Openness -konseptiin kuuluvaa

WinCC Unified Runtime Openness -rajapintaa sekä Siemensin Open Pipe -rajapintaa. Siemensin omat rajapinnat ovat periaatteellisesti avoimia eli niiden ominaisuudet ovat vapaasti hyödynnettävissä, mutta vaativat lisensoidun WinCC Unified -valvomosovelluksen. ODBC-rajapinta on täysin avoin. Kaikki kolme rajapintaa ovat kaksisuuntaisia, mutta Open Pipe ominaisuuksiltaan hyvin rajoittuneesti. (Siemens 2020c, 11.)

4.2.1 Siemens WinCC Unified Runtime Openness -rajapinta

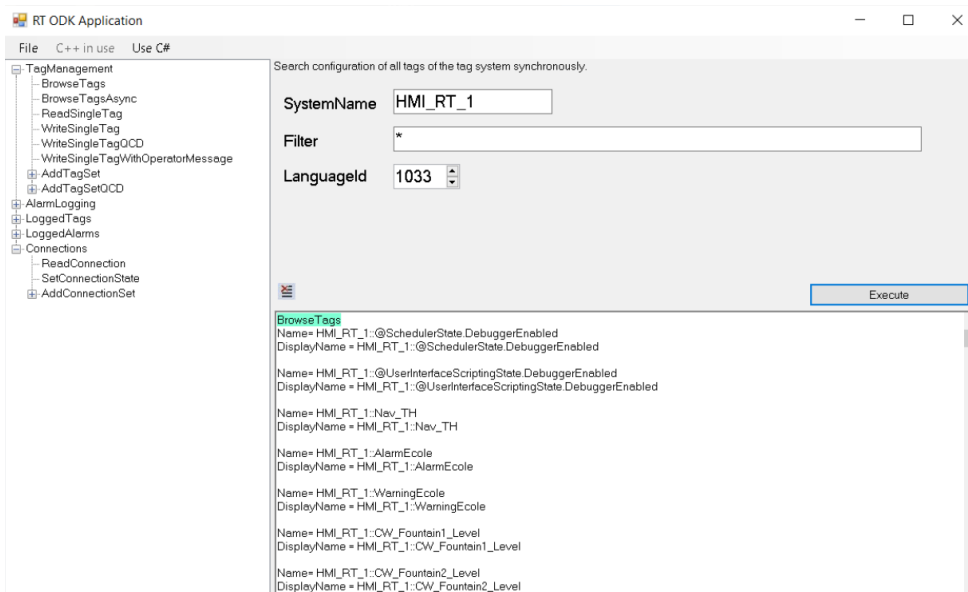
Runtime Openness -rajapinta soveltuu suurten tietomäärien ja järjestelmäasetuksien muokkaamiseen asiakasohjelman ja WinCC Unified Runtime -järjestelmän välillä. Rajapinnalla on mahdollista lukea ja muokata WinCC Unified Runtime -järjestelmän objekteja, kuten tageja, arkistoja ja hälytyksiä sekä koko Runtime-järjestelmän asetuksia, kuten yhteyksiä ja laitosmalleja. Esimerkiksi yhdellä suoritteella voidaan lukea arvot 1000 tagista tai katkaista yhteys tiettyyn käyttöpaneeliin. Ohjelmointikielinä rajapinnan kanssa toimivat C++ ja C#. Sovelluksien ajo tapahtuu paikallisesti WinCC Unified Runtime -järjestelmässä. Runtime Openness -rajapintaa nimitetään joissakin yhteyksissä myös Open Development Kit (ODK) -kokonaisuudeksi, jolla viitataan sen asennuspakettiin sovelluskehitysympäristöjä varten. (Siemens 2020c, 21; Siemens 2021c, 9, 11.)

Rajapinnan komentoja on kaikkiaan yli kuusikymmentä, joten niiden läpi käyminen ei ole tarkoituksenmukaista, mutta ne voidaan jakaa kahdeksan pääotsikon alle seuraavasti:

- järjestelmän yleinen hallinta
- järjestelmän sisäisten virheilmoitusten lukeminen
- tagien ja arkistojen lukeminen ja tagien hallinta
- hälytyksien lukeminen ja hallinta
- järjestelmän yhteyksien hallinta
- laitosmallin hallinta
- kalenterin hallinta
- asiayhteysarkistoinnin ja -määrytyksien hallinta.

Komennot antavat vastauksensa ohjelmointikielen mukaisina muuttujina ja rakenteina, kuten kokonais- ja liukulukuina, merkkijonoina, tauluina jne. (Siemens 2021c, 28, 35, 39, 65, 82, 104, 115, 134.)

Kuvassa 72 on Siemensin tarjoama esimerkkiohjelma, jonka lähdekoodi on ladattavissa heidän tukisivustoillaan. Ohjelmalla ei ole varsinaista tosimaailman käyttötarkoitusta, mutta sillä voi harjoitella ohjelman kääntämistä sovelluskehitysympäristössä ja kokeilla osaa rajapinnan komennoista graafisen käyttöliittymän kautta. (Siemens 2020d, 4.)



Kuva 72. Siemensin esimerkkiohjelma Runtime Openness -rajapinnalle (Siemens 2020d, 4)

Mainitsemisen arvoinen lisämahdollisuus on, että rajapinnalla voidaan luoda asiayhteyshmäärittämiä (engl. Contexts) WinCC-valvomosovellukseen. Ne vastaavat PM-SERVER-sovelluksen asiayhteyssarkistointia, jossa arkistoon tallentuvat tagin arvo ja tilanteen alku- ja päätösaikaleimat. Lähdemateriaalin perusteella näitä asiayhteyshmäärittämiä ei ole mahdollista toteuttaa valvomosovellukseen muuten kuin hankkimalla Performance Insight- tai Calendar-lisäosa tai määrittämällä ne tämän rajapinnan komentojen avulla. Tällä on välillinen vaikutus raportointiin, koska asiayhteyksiä voidaan käyttää raportoinnin suodatuskriteereinä eri työkaluissa. (Siemens 2022d, 2769–2770.)

4.2.2 WinCC Unified Open Pipe -rajapinta

Open Pipe -rajapinta on ominaisuuksiltaan rajoittuneempi kuin Runtime Openness -rajapinta, mutta asiakasohjelma voidaan toteuttaa millä tahansa ohjelmointikielellä. Rajapintaa hyödyntävät sovellukset ajetaan yhtä lailla WinCC Unified Runtime -järjestelmässä. Tiedonsiirto tapahtuu hyödyntäen Pipe-tekniologiaa, joka on kahden sovellusprosessin välinen tiedonsiirtomenetelmä. Siinä komentoja suoritetaan FIFO-jonokurin periaatteella eli komennot käsitellään saapumisjärjestyksessä. Rajapinnalla on mahdollista lukea ja kirjoittaa taageihin sekä lukea hälytyksiä. Open Pipe -rajapinnan kirjoitusasu vaihtelee lähteestä riippuen ja esiintyy osassa lähteistä yhteen kirjoitettuna muodossa OpenPipe tai OpenPIPE. (Siemens 2020c, 21; Siemens 2021d, 5–6, 9.)

Rajapintaa on mahdollisuus käyttää Basic- tai Expert-tason komennoilla. Basic-tason komennot soveltuvat yksinkertaisiin komentojonoihin, joita ajetaan järjestelmän komentorivillä. Expert-tason komennot ja vastaukset ovat muotoilultaan JSON-formaattia, joten niitä on käytännöllisempi hyödyntää korkeamman tason sovelluksissa, jotka ovat toteutettu esimerkiksi ohjelmointikielillä, kuten C#, C++, JavaScript tai Python. (Siemens 2021d, 7.)

Basic-tason komennoilla voidaan käsitellä yksittäistä objektia kerrallaan eli yhtä tagia tai hälytystä. Komennot eivät saa sisältää erikoismerkkejä, mikä on huomioitava tagien ja hälytyksien nimeämisessä. Kutsu muodostuu annetusta komennosta, kohteesta ja mahdollisesta arvosta, jos kutsulla halutaan kirjoittaa kohteeseen. Vastaus palautuu Notify- tai Error-etuliitteellä sallittuna tai virheellisenä. Vastauksen loppurakenne koostuu annetusta komennosta, kohteesta ja palautusarvosta tai virheviestistä. Basic-tason komennoilla voidaan lukea ja kirjoittaa tagin arvoja, käynnistää tagin monitorointi ja listata järjestelmässä olevat tagit. Hälytyksistä voidaan vain listata järjestelmässä olevien hälytyksien nimet ja luokat, mutta ei monitoroida hälytyksiä. (Siemens 2021d, 13–14.)

Expert-tason komennoilla voidaan käsitellä yhtä tai useampaa objektia kerrallaan. Komennot ja vastaukset ovat muotoilultaan JSON-formaattia. Vastaukseen viestin sisältävään elementtiin on sisällytetty samat Notify- ja Error-etuliitteet, kuten Basic-tason komennoissa. Expert-tason komennoilla voidaan

tehdä vastaavat toimenpiteet kuten Basic-tason komennoilla, mutta lisäksi monitoroida ja suodattaa hälytyksiä. Huomionarvoisena seikkana on, että Basic-tai Expert-tason komennoilla ei voida lukea Runtime-järjestelmän arkistoja. (Siemens 2021d, 27–29.)

Kuvassa 73 on opinnäytetyötä varten WinCC Unified -ympäristöön toteutettu valvomokäyttöliittymä, jossa Open Pipe -rajapinnan toimintaa on testattu Siemensin esimerkkiohjelmakoodia muokkaamalla. JavaScript-ohjelmointikielellä toteutettu asiakasohjelmalla tulostaa rajapintakutsulla saamansa tiedot komentoriville. Ohjelmalla monitoroidaan vihreällä värillä kuvattua Value1-tagia. Käytetty komento on SubscribeTagValue. Palaute sisältää tagin nimen, tagin arvon sekä erillisen laatuarvon yhteydelle. Tagin arvoa on vaihdeltu 28 ja 13 välillä käyttöliittymässä.



Kuva 73. Open Pipe -rajapinta ja tagin monitorointi Basic-tason komennolla

Vastaavanlaisella sovelluksella on kuvassa 74 testattu hälytyksien monitorointia Expert-tason komennolla SubscribeAlarm. Valvomosovellukseen on määritetty Value1_high-niminen hälytysraja, joka laukaisee ilmoituksen ”Value 1 getting high”, jos prosessiarvo nousee arvoon 40 tai sen ylitse. Nämä tiedot ovat luettavissa vastauksesta, aikaleimojen ja lukuisten muiden tietojen lisäksi.

The screenshot displays the WinCC Unified Expert-level monitoring interface. On the left, a console window shows the following alarm details:

```

Start client
Client: on connection
alarmName: HMI_RT_1::Value1:Value1_alarm1
alarmClassName: Notification
alarmClassSymbol:
alarmText1:
alarmText2:
alarmText3:
alarmText4:
alarmText5:
alarmText6:
alarmText7:
alarmText8:
alarmText9:
area:HMI_RT_1::Alarming
backColor:428958334
changeReason:3
clearTime:1970-01-01 00:00:00.0000000
connection:1.0.0.0:0
deadBand:No deadband configured.
eventText:Value1 getting high
flashing:FALSE
hostName:DESKTOP-4U6SFOO
ID:1
InfoText:
InstanceID:9
LoopInAlarm:
modificationTime:2022-07-24 12:53:16.4723780
alarmName:HMI_RT_1::Value1:Value1_alarm1
notificationReason:1
origin:
priority:4
resetTime:2022-07-24 12:53:16.4723780
resetTime:1970-01-01 00:00:00.0000000
sourceID:
sourceType:1
state1
stateMachine:3
stateText:Incoming
suppressionState:0
systemSeverity:0
tag:1.200.1.0.0.0
textColor:4278190000
userName:
value:40
valueLimit:40
valueQuality:192
AlarmGroupID:0

```

The main interface shows two vertical gauges with values 40 and 30, and a line graph with two data series. Below the graph is a table with the following data:

| Alarm class | Origin | Area | Alarm text | Status text |
|-------------|--------------|--------------------|---------------------|-------------|
| 1 | Notification | HMI_RT_1::Alarming | Value1 getting high | Incoming |
| 2 | | | | |
| 3 | | | | |
| 4 | | | | |
| 5 | | | | |
| 6 | | | | |
| 7 | | | | |

Kuva 74. Open Pipe -rajapinta ja hälytyksien monitorointi Expert-tason komennolla

4.2.3 Open Database Connectivity -rajapinta

WinCC Unified -valvomosovellus käyttää arkistointiin SQLite- tai Microsoft SQL Server tietokannanhallintajärjestelmiä, jotka molemmat ovat tuettuina ODBC-rajapinnan määrittelyissä ja tarvitsevat vain soveltuvat ajurit rajapinnan hyödyntämiseksi. SQLite-hallintajärjestelmälle on saatavilla avoimen lähdekoodin ajuri, jonka viimeisin päivitys on vuodelta 2020 ja tukee SQLite 3.32.3 versiota (Werner 2020). Microsoft SQL Server -hallintajärjestelmän ajuri on saatavilla Microsoftin dokumentaationsivuston kautta. Sen viimeisin päivitys on helmikuulta 2022 ja tukee Microsoft SQL Server -hallintajärjestelmän eri versioita sekä Microsoftin Azure pilvipalvelun Azure SQL Database -hallintajärjestelmää (Microsoft 2022).

ODBC-rajapinnalla toteutetun raportoinnin skaalautuvuus periytyy WinCC Unifiedn lisenssistä ja sen määrittelemästä käytettävien tagien määrästä. Raportointiin käytettävä asiakassovellus voidaan toteuttaa millä tahansa ohjelmointikielellä, jossa on ODBC-rajapinnan käyttöön soveltuvat kirjastot. Asiakassovelluksella pitää olla vain pääsy WinCC Unifiedn arkistoon rajapinnan välityksellä. Se voi tapahtua ODBC-rajapinnan määrittelyjen mukaisesti paikallisesti

tai verkkoyhteyden ylitse. SQLiten ollessa tiedostopohjainen arkistointijärjestelmä, verkkoyhteysprotokollat perustuvat luonnollisesti käytettyyn tiedostonjakomenetelmään. Microsoft SQL Server -hallintajärjestelmän ODBC-ajurin kuljetuskerroksessa käytetään TCP-protokollaa (Microsoft 2022).

Manipuloidessa tietokantoja ODBC-rajapinnalla SQL-kielen komennot ja kyselyt sisällytetään asiakasohjelman ohjelmakoodiin. Asiakasohjelman ohjelmointi rajautuu tämän opinnäytetyön ulkopuolelle, mutta kuvan 75 esimerkillä halutaan testata SQL-kielillä toteutettua kyselyä, joka soveltuisi yksinkertaisen raportin toteuttamiseen WinCC Unified -valvomosovelluksen SQLite-tietokannoista. Käytetty SQL-kysely on sovellettu Siemensin antamasta esimerkistä ja se toimii Siemensin suosittelemassa DB Browser for SQLite -sovelluksessa. Kysely ei sellaisenaan ole siirrettävissä mihinkään ohjelmakoodiin. Esimerkissä yhdistetään tietoja kahdesta taulusta. Vasemmanpuoleinen taulu sijaitsee prosessitiedon päätietokannassa ja sisältää tagin nimen sekä sille yksilöidyn numeron. Oikeanpuoleinen taulu sijaitsee tietokantasegmentissä ja sisältää vastaavan tagille yksilöidyn numeron sekä prosessitietoa aikaleimoinneen. Aikaleima muunnetaan koordinoituun yleisaikaan kyselyn yhteydessä. (Siemens 2021e, 90, 99–100.)

Kysely

```

1 SELECT
2 strftime('%Y-%m-%d %H:%M:%f',
3     datetime(((("pk_TimeStamp"/86400/1.0E+7)-134774)*86400), 'unixepoch'))
4     AS UTC_TimeStamp,
5     LogTagName.pk_Key AS TagID,
6     LogTagName.Name AS TagNameAndLog,
7     LogSegment.Quality AS Quality,
8     LogSegment.Value
9 FROM Test_HMI_SIM_TLG116_20220526_062758.LoggedProcessValue AS LogSegment
10 INNER JOIN LoggingTag AS LogTagName
11 ON LogTagName.pk_Key = LogSegment.pk_fk_Id

```

Taulut

| pk_Key | TagSystemId | TagObjectId | LoggingTagId | Name |
|--------|-------------|-------------|--------------|-----------------------------|
| 1 | 1 | 114 | 687865857 | Test_HMI::Value1:Value1_log |
| 2 | 2 | 120 | 687865857 | Test_HMI::Value2:Value2_log |

| pk_TimeStamp | pk_fk_Id | Quality | Value |
|--------------------|----------|---------|-------|
| 132980201153942637 | 2 | 192 | 6.0 |
| 132980201161054980 | 1 | 192 | 7.0 |

Tulos

| | UTC_TimeStamp | TagID | TagNameAndLog | Quality | Value |
|---|-------------------------|-------|-----------------------------|---------|-------|
| 1 | 2022-05-26 06:28:35.000 | 2 | Test_HMI::Value2:Value2_log | 192 | 6.0 |
| 2 | 2022-05-26 06:28:36.000 | 1 | Test_HMI::Value1:Value1_log | 192 | 7.0 |
| 3 | 2022-05-26 06:29:31.000 | 1 | Test_HMI::Value1:Value1_log | 192 | 20.0 |
| 4 | 2022-05-26 06:29:30.000 | 2 | Test_HMI::Value2:Value2_log | 192 | 14.0 |
| 5 | 2022-05-26 06:29:50.000 | 1 | Test_HMI::Value1:Value1_log | 192 | 27.0 |
| 6 | 2022-05-26 06:29:51.000 | 2 | Test_HMI::Value2:Value2_log | 192 | 21.0 |
| 7 | 2022-05-26 06:29:55.000 | 2 | Test_HMI::Value2:Value2_log | 192 | 32.0 |

Kuva 75. Raportointiesimerkki ODBC-rajapinnalle ja sen SQL-kysely (mukaan Siemens 2021e, 99–100)

4.3 Raportointilisäosa WinCC Unified Report Execution

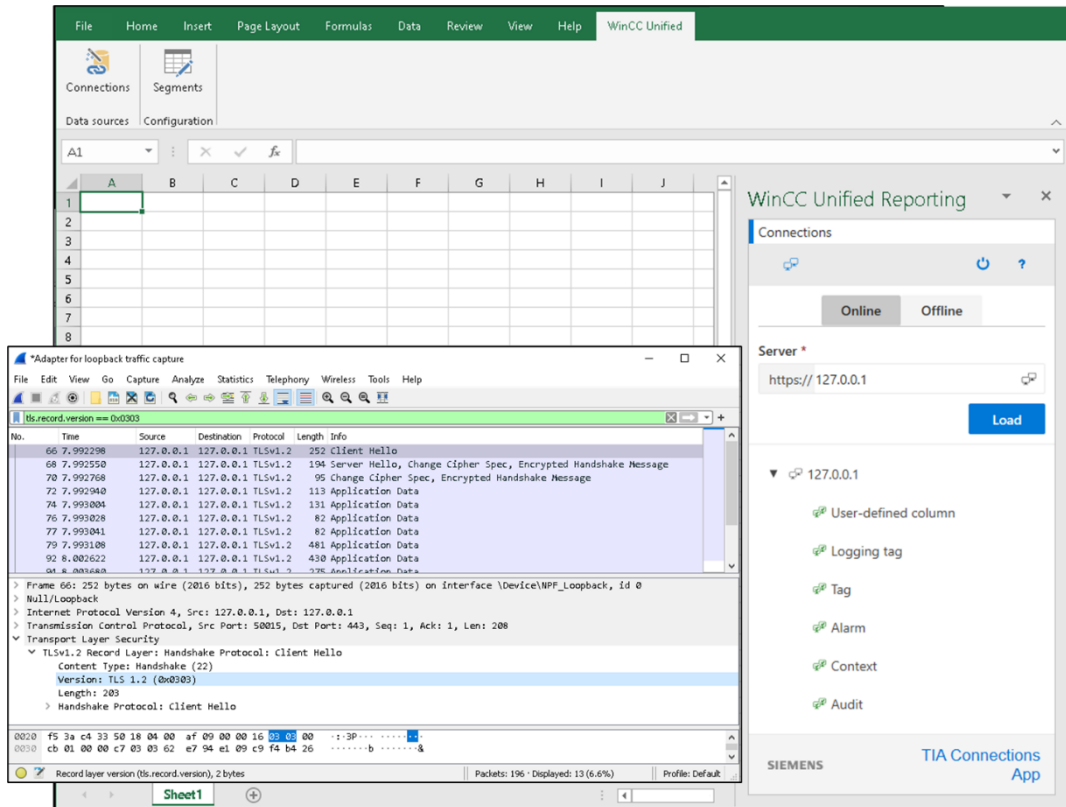
WinCC Unified -valvomosovelluksen mukana toimitetaan raportointilisäosa Report Execution. Se on Microsoft Excel -taulukkolaskentaohjelmaan asennettava liitännäinen ja vastaa toimintaperiaatteeltaan PM-lisäosien Excel-liitännäistä. Raporttien ajaminen ja määrittelyt tapahtuvat suoraan valvomokäyttöliittymän Reports Control -objektissa. Raportoinnin tietolähteinä voidaan käyttää valvomosovelluksen tagitietoa, arkistoja ja hälytyksiä. Mikäli WinCC Unifiedin oheen on asennettu Plant Intelligence-, Line Coordination- tai Audit Basis -lisäosa, voidaan toteuttaa lisäksi näiden tuotanto- ja seurantaraportteja. Raportit tallennetaan PDF- tai XLSX-tiedostoiksi. Raportointiajosta on mahdollista lähettää automatisoitu sähköpostiviesti ja sisällyttää raportti liitetiedostoksi viestiin. Lähdemateriaalissa ei ole mainintaa raportoinnin skaalautuvuuden rajoitteista, joten sen voi olettaa periytyvän valvomosovelluksen lisenssistä ja arkistointiasetuksista. (Siemens 2021h, 131; Siemens 2022d, 457–460, 519.)

Tässä kappaleessa esitellyt toiminnot on testattu WinCC Unified -valvomosovelluksessa ja opinnäytetyötä varten toteutetussa valvomokäyttöliittymässä. Report Execution -raportointilisäosasta käytetään versiota 3, joka toimitetaan TIA Portalin version 17 mukana. Excel-taulukkolaskentaohjelmasta käytetään version 2008 koontiversiota 13127.20616, joka on osa Microsoft Office Pro plus 2019 -ohjelmistopakettia.

Raportointipohjat toteutetaan Excel-taulukkolaskentaohjelmaan asennettavalla työkalupalkilla. Raportointipohjan luominen aloitetaan haluttujen tietolähteiden valinnalla. Tämä tapahtuu joko yhteydellisessä (engl. online) tai yhteydettömässä (engl. offline) tilassa. Yhteydellisessä tilassa tietolähteet voidaan lukea suoraan valvomosovelluksesta paikallis- tai verkkoyhteyden kautta. Yhteydettömässä tilassa käytetään valvomosovelluksesta luotua JSON-määrittelytiedostoa, joka sisältää käytössä olevat tagit ja muut soveltuvat tietolähteet. (Siemens 2022d, 465, 538).

Kuvassa 76 on työkalupalkki Excel-taulukkolaskentaohjelmassa. Työkalupalkki on käytännössä Excel-taulukkolaskentaohjelman sisällä toimiva verkkoselain. Kuvassa 76 näkyy myös tietolähteiden valintaa valvomosovelluksesta

paikallisyhteyden ylitse. Yhteys on suojattu TLSv1.2-protokollalla, joka on todennettu tarkkailemalla verkkoyhteyttä kirjautumishetkellä Wireshark-paketti-analysaattorin avulla.

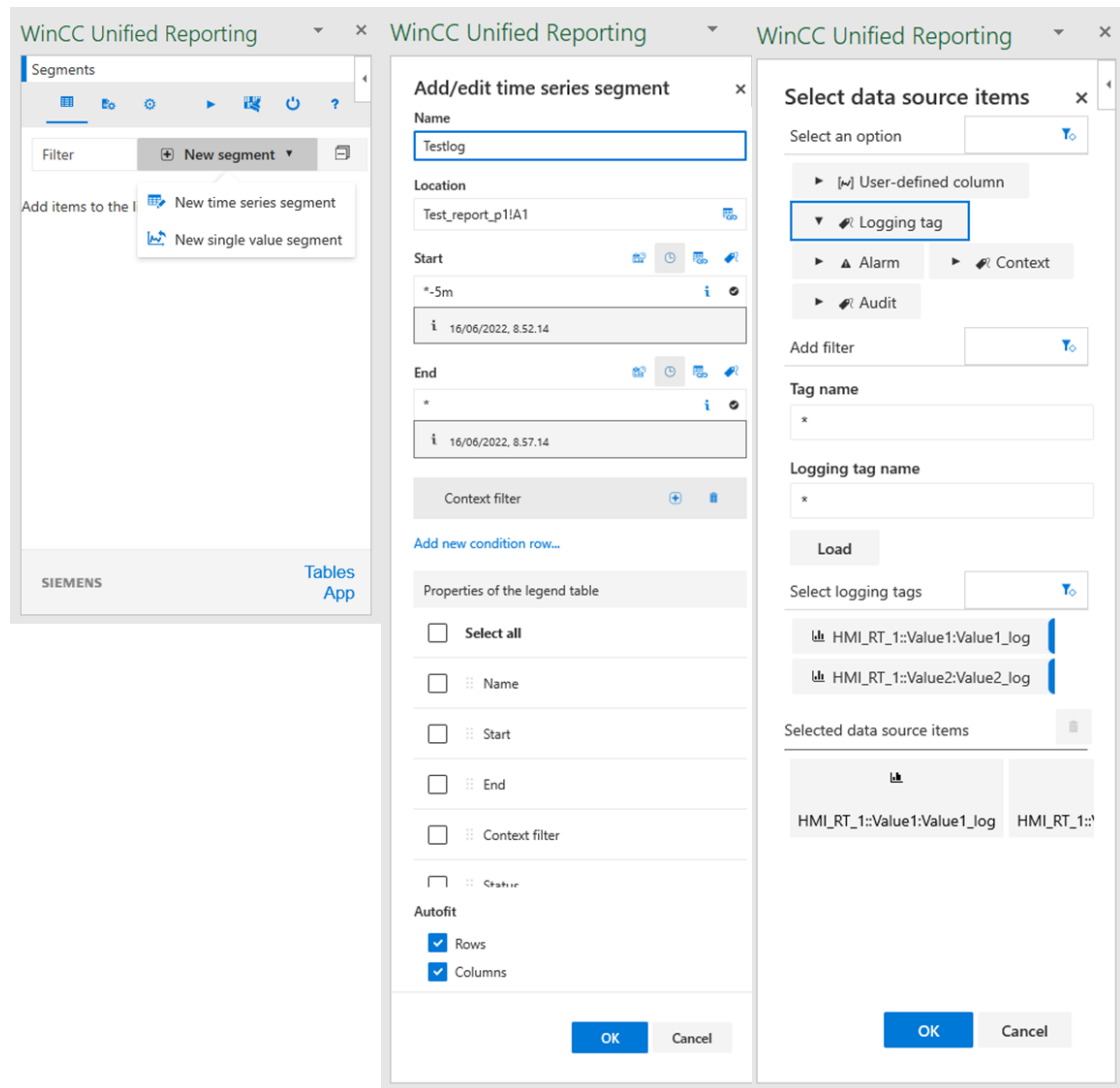


Kuva 76. Lisäosan Excel-työkalupalkki, tietolähteiden valinta ja tiedonsiirron suojauksen todennus

Raportointipohja koostetaan segmenteistä ja niiden sisällä esitettävistä tietolähteistä. Segmentti on yleisnimitys solualueesta, johon raportointitieto tallennetaan. Segmenttejä on kahta tyyppiä: aikasarjasegmentti (engl. time series segment) ja yhden arvon segmentti (engl. single value segment). Aikasarjasegmentti on taulukkomuotoinen solualue, joka esittää määritellyltä aikaväliltä kerättyä arkistotietoa tai hälytyksiä. Aikaväliksi voidaan määrittää absoluuttinen-, suhteellinen-, soluun kirjattu-, tagista luettu- tai prosessiin määritettyjen asiayhteyksien välinen aika. Yhden arvon segmentti esittää nimensä mukaisesti tietyn tiedon yhdessä solussa. Siinä tietolähteenä voidaan näyttää tietoa tagista tai arkistosta. (Siemens 2022d, 479–482.)

Tietolähteitä voi saman segmentin sisällä olla useita. Esimerkiksi aikasarjasegmentti voi koostua arkistoidun prosessitiedon ja hälytyksien yhdistel-

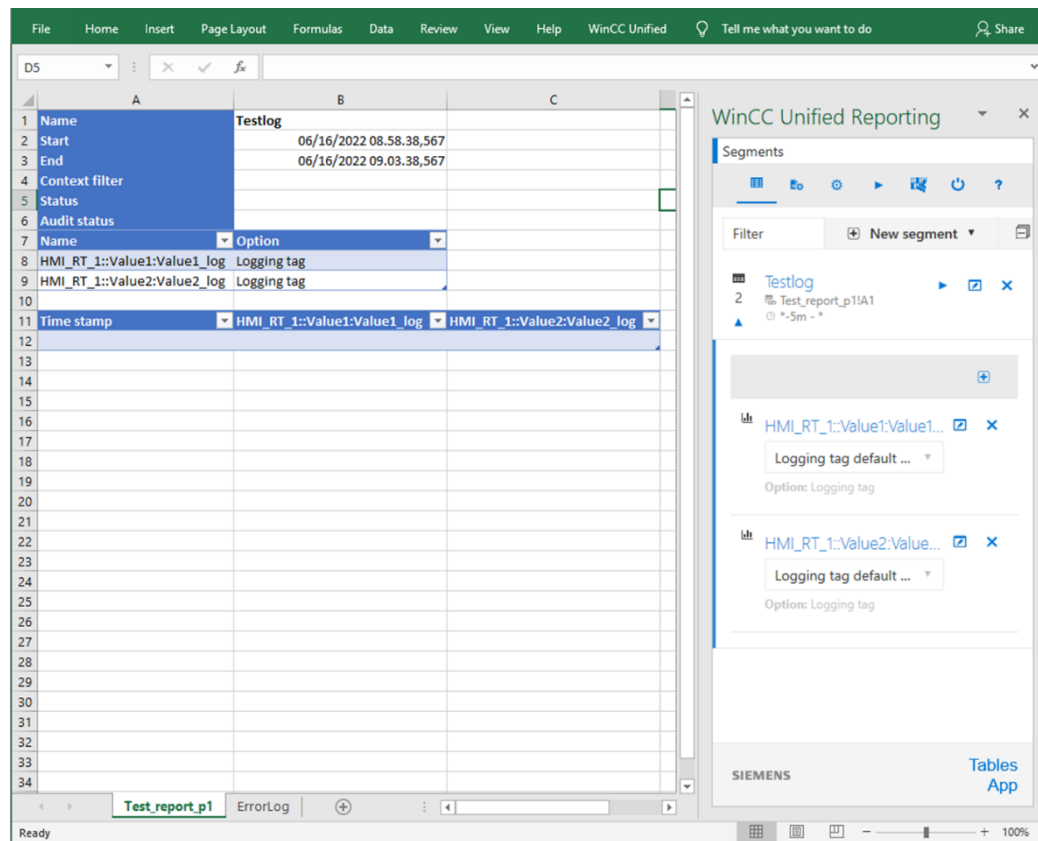
mästä. Aikasarjasegmentin tietolähteenä voi olla myös ns. itsemääritelty sarakke, joka voi sisältää esimerkiksi tekstiä tai Excel-kaavan. (Siemens 2022d, 481–482.) Kuvassa 77 on luotu uusi aikasarjasegmentti, joka tallennetaan A1-solusta alkaen työkirjan välilehdelle Test_report_p1. Segmenttiin haetaan tiedot raportointihetkestä viisi minuuttia taaksepäin. Legend table -kohdassa tehdyt valinnat tulevat erilliseksi otsakkeeksi aikasarjasegmentin taulukon yläpuolelle. Tietolähteinä ovat arkistot Value1_log ja Value2_log.



Kuva 77. Aikasarjasegmentin määrittelyn vaiheet

Työkalu muodostaa raportointipohjan tehtyjen valintojen pohjalta automaattisesti. Jos valintoihin tehdään muutoksia, rakentuu koko pohja uudelleen. Poiketen PM-lisäosien Excel-liitäntäisestä, raporttipohjan soluissa ei esiinny kaavoja, joten pohjat eivät ole käyttäjän muokattavissa ilman työkalupalkkia. Kuvassa 78 on aikasarjasegmentti sijoittuneena raportointipohjalla. Otsake alkaa

solusta A1 ja otsakkeen alapuolella on solusta A11 alkaen varsinainen taulukko, johon tiedot täydentyvät.

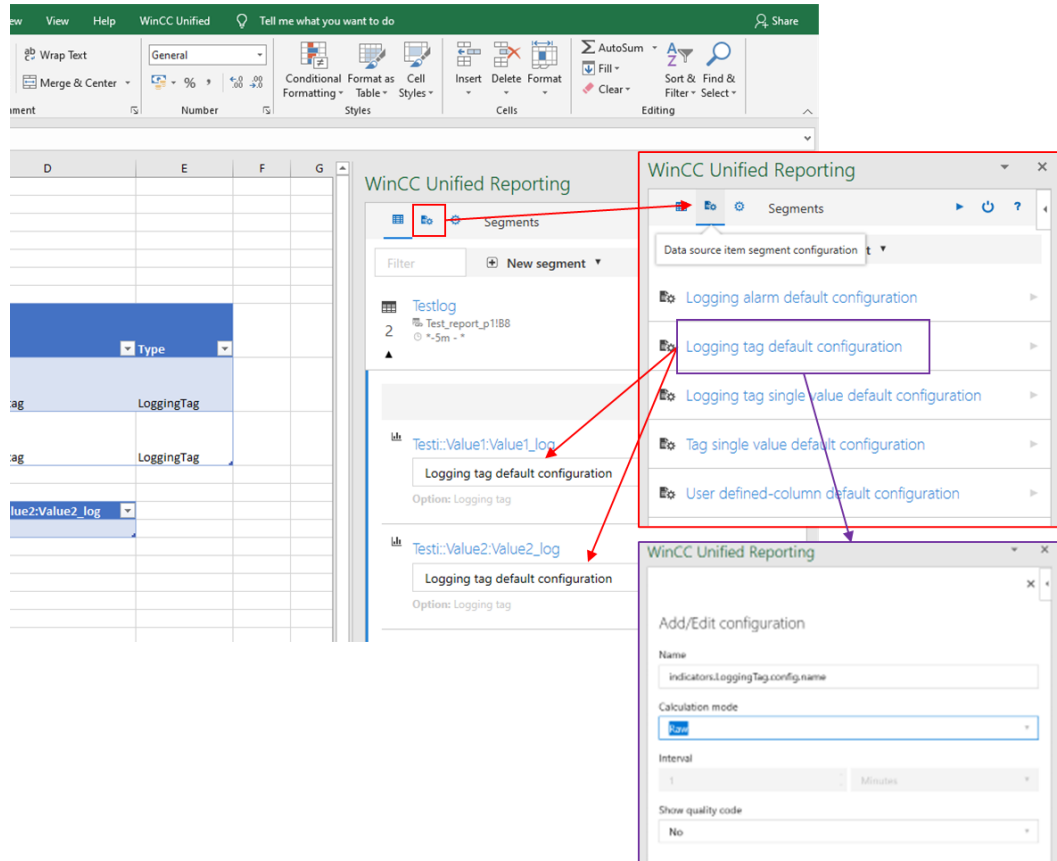


Kuva 78. Aikasarjasegmentti sijoittuneena raportointipohjalle

Segmenttien ja tietolähteiden sisältöä voidaan muokata niiden lisäämisen yhteydessä. Aikasarjasegmenttien yhteydessä voidaan valita esitettävien sarakkeiden määrää sekä otsakkeessa esitettäviä tietoja. Yhden arvon segmenttien kohdalla voidaan määrittää ympäröiviin soluihin haettavia tietoja. Arkistotiedon kanssa on käytössä Calculation mode -asetus, jolla voidaan määrittää, mikä arvo esitetään, jos aikaleiman kohdalla ei ole arvoa saatavilla. Aikasarjasegmentin Raw-asetuksella ei esitetä arvoa ollenkaan, keep last value -asetuksella esitetään edellinen arkistoitu arvo ja interpolate-asetuksella tehdään lineaarinen interpolointi aikaväliltä. Yhden arvon segmentissä on myös käytössä sama interpolate-asetus sekä lisäksi left- ja right-asetukset, joilla valitaan arvo edeltävältä tai seuraavalta aikaleimalta. (Siemens 2022d, 503–511, 516.)

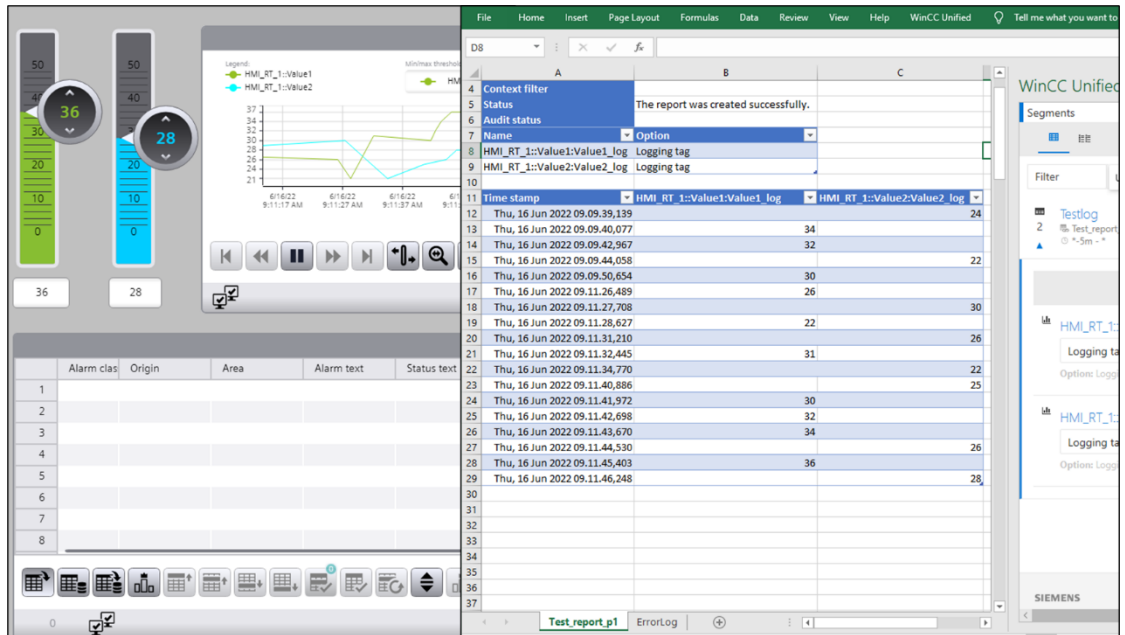
Kaikki muokkaukset ja asetellut voidaan tallentaa konfiguraatioksi, täten samaa tietosisältöasettelua voidaan käyttää muissakin segmenteissä. Itsemäärittelyssä sarakkeessa käytettävä kaava tai esitettävä sisältö määritellään myös

konfiguraatioiden kautta. Kuvassa 79 havainnollistetaan, miten arkistojen Value1_log ja Value2_log kanssa käytetään Logging tag default configuration -nimistä konfiguraatiota. Sen sisällä on määritelty, että tagin yhteyden laatuarvoa ei esitetä ja Calculation mode -asetus on arvossa Raw.



Kuva 79. Segmenttien tietosisällön muokkaus konfiguraatioilla

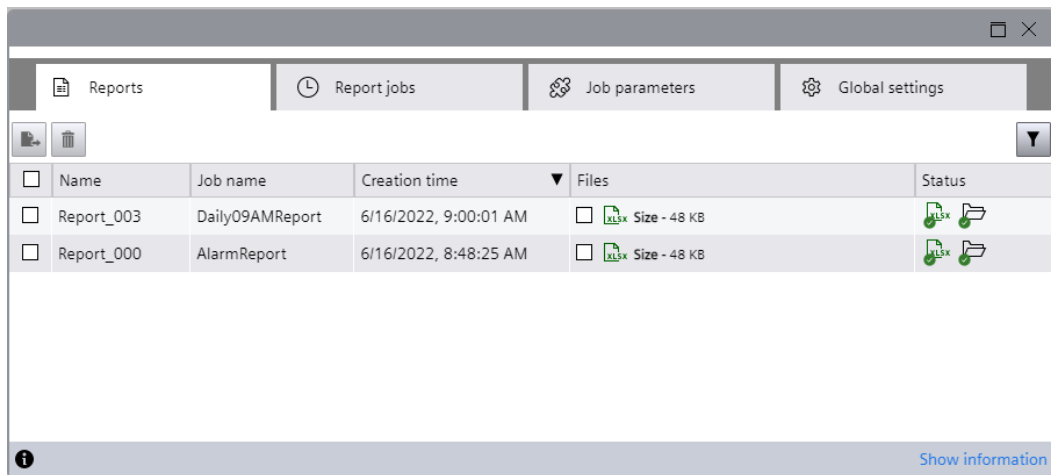
Mikäli raportointipohjaa työstetään yhteydellisessä tilassa, voidaan pohjan toimivuutta koestaa suoraan valmistelun aikana. Työkalupalkissa olevilla toistopainikkeilla voidaan päivittää pohjalle yhden tai kaikkien segmenttien tiedot (Siemens 2022d, 515). Kuvassa 80 on opinnäytetyötä varten luotu valvomokäyttöliittymä ja raportointipohja koestettuna. Value1_log- ja Value2_log-arkistot näkyvät taulukossa aikaleimoinen. Siinä havainnollistuu myös, miltä valvomosovelluksessa asetettu muutosperusteinen arkistointi näyttää raporttipohjalla, kun sen kanssa on käytössä Calculation mode -asetuksista RAW.



Kuva 80. Raportointipohjan koestaminen valmistelun yhteydessä

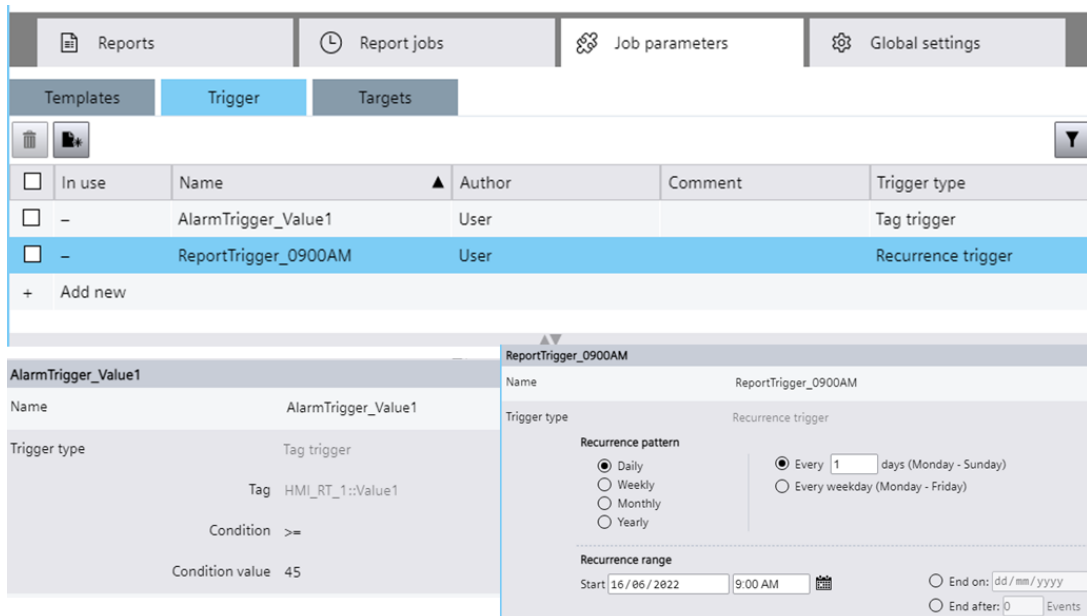
Raporttien ajaminen sekä niihin liittyvät määrittelyt tehdään valvomosovelluksen Reports Control -objektilla. Se koostuu välilehdistä Reports, Report jobs, Job parameters ja Global settings sekä niiden alla olevista asetuksista. Reports-välilehdelle tulevat kaikki ajettut raportit lista näkymässä. Report jobs -välilehdellä valitaan raportointiajon asetukset, joiden tarkemmat määrittelyt tehdään Job parameters -välilehdellä. Job parameters -välilehdelle tuodaan Excel-työkalulla valmistellut raporttipohjat, määrittellen raportointia ajon käynnistämisen perusteet ja raporttien tallennussijainnit. Global settings -välilehdellä määritetään sähköpostin lähettämisen asetukset sekä voidaan luoda JSON-määrittelytiedosto käytettäväksi Excel-työkalun yhteydettömän tilan kanssa. (Siemens 2022d, 519–521, 533.)

Kuvassa 81 on Reports Control -objekti opinnäytetyön valvomokäyttöliittymässä. Aktiivisena on Reports-välilehti, jossa näkyy kaksi ajettua raporttia, jotka ovat ajettu aiemmin Excel-työkalulla valmistellulle raportointipohjalle.



Kuva 81. WinCC Unified -valvomosovelluksen Reports Control -objekti

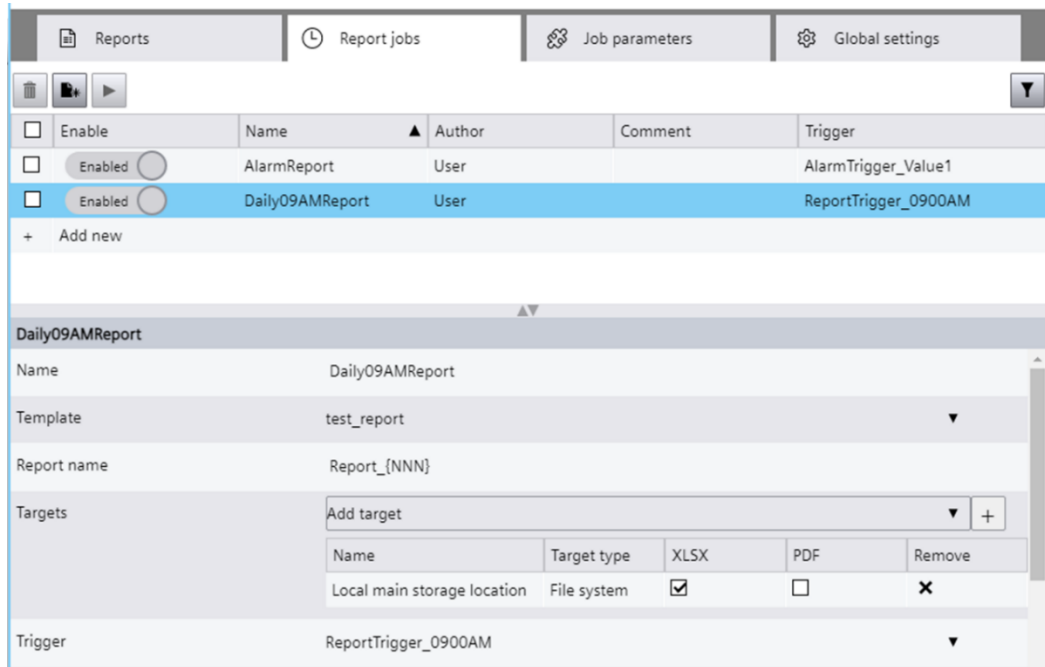
Raportointiajo voidaan käynnistää manuaalisesti sekä aikataulun, tagin arvon tai asiayhteysmäärittelyn perusteella. Raportointiajon käynnistävät tilat määritellään Job parameters -välilehden Trigger-asetuksissa. (Siemens 2022d, 528–530.) Kuvassa 82 on luotu kaksi raportointiajon käynnistysehtoa. Alla olevassa isossa kuvassa on Value1 tagin hälytysrajan käynnistysehto. Oikeaan alakulmaan päälle liimatussa kuvassa näkyy käynnistysehto, joka aktivoituu päivittäin aamuyhdeksältä.



Kuva 82. Raportointiajon käynnistysehtoja Job parameters -välilehdellä

Kuvassa 83 on määritetty kaksi raportointiajoa, jotka perustuvat aiemmin määritettyihin käynnistysehtoihin. Esimerkkinä aamuyhdeksän raportointiajon yle-

set asetukset, jossa käynnistysehtona on aiemmin määritetty ReportTrigger_0900AM, raportointipohjana käytetään test_report-raportointipohjaa ja raportti tallennetaan juoksevilla numerolla.



Kuva 83. Raportointiajon yleiset asetukset Report jobs -välilehdellä

Kuvassa 84 on AlarmReport-raportointiajon tuottama raportti. Siinä on aikasarjasegmentissä arkistojen Value1_log ja Value2_log prosessitietoa sekä niiden hälytystiedot. Raportointiajo käynnistyi, kun tagin Value1 saavutti arvon 45.

| Name | Start | End | Status |
|---------|-------------------------|-------------------------|--------------------|
| Logging | 06/16/2022 08:38:23,945 | 06/16/2022 08:48:29,945 | Report successful. |

| TimeStamp | HMI | Value1 | Value2 | Alarm | Name | Alarm class name | Event text | State | Time when raised |
|-------------------------------|-----|--------|--------|-------|---------------|------------------|---------------------|-------|-------------------------------|
| Thu, 16 Jun 2022 08:41:19,581 | | 16 | | | | | | | |
| Thu, 16 Jun 2022 08:41:19,873 | | 10 | | | | | | | |
| Thu, 16 Jun 2022 08:41:20,510 | | | | 24 | | | | | |
| Thu, 16 Jun 2022 08:41:21,134 | | 19 | | | | | | | |
| Thu, 16 Jun 2022 08:45:50,286 | | 23 | | 20 | | | | | |
| Thu, 16 Jun 2022 08:45:51,113 | | | | | | | | | |
| Thu, 16 Jun 2022 08:45:00,823 | | 18 | | 25 | | | | | |
| Thu, 16 Jun 2022 08:45:01,652 | | | | 26 | | | | | |
| Thu, 16 Jun 2022 08:45:05,945 | | 22 | | | | | | | |
| Thu, 16 Jun 2022 08:46:08,353 | | | | 22 | | | | | |
| Thu, 16 Jun 2022 08:46:08,265 | | 28 | | | | | | | |
| Thu, 16 Jun 2022 08:46:11,425 | | | | | | | | | |
| Thu, 16 Jun 2022 08:46:12,761 | | 20 | | | | | | | |
| Thu, 16 Jun 2022 08:47:26,102 | | 26 | | | | | | | |
| Thu, 16 Jun 2022 08:47:26,996 | | | | 19 | | | | | |
| Thu, 16 Jun 2022 08:47:31,905 | | | | 22 | | | | | |
| Thu, 16 Jun 2022 08:47:34,989 | | 40 | | 3 | Value2_alarm1 | Notification | Value2 getting high | 1 | Thu, 16 Jun 2022 08:47:34,989 |
| Thu, 16 Jun 2022 08:47:34,989 | | 40 | | 3 | Value2_alarm1 | Notification | Value2 getting high | 0 | Thu, 16 Jun 2022 08:47:34,989 |
| Thu, 16 Jun 2022 08:47:39,345 | | 42 | | | | | | | |
| Thu, 16 Jun 2022 08:47:41,894 | | | | 39 | | | | | |
| Thu, 16 Jun 2022 08:47:46,543 | | 29 | | | | | | | |
| Thu, 16 Jun 2022 08:47:50,453 | | | | 22 | | | | | |
| Thu, 16 Jun 2022 08:47:53,732 | | 36 | | | | | | | |
| Thu, 16 Jun 2022 08:47:56,428 | | 31 | | | | | | | |
| Thu, 16 Jun 2022 08:48:00,851 | | | | 35 | | | | | |
| Thu, 16 Jun 2022 08:48:02,543 | | 36 | | | | | | | |
| Thu, 16 Jun 2022 08:48:06,875 | | 40 | | 1 | Value1_alarm1 | Notification | Value1 getting high | 1 | Thu, 16 Jun 2022 08:48:06,875 |
| Thu, 16 Jun 2022 08:48:09,005 | | | | 32 | | | | | |
| Thu, 16 Jun 2022 08:48:10,136 | | 30 | | | | | | | |
| Thu, 16 Jun 2022 08:48:11,489 | | | | 27 | | | | | |
| Thu, 16 Jun 2022 08:48:13,439 | | 42 | | | | | | | |
| Thu, 16 Jun 2022 08:48:14,827 | | | | 43 | | | | | |
| Thu, 16 Jun 2022 08:48:17,259 | | 45 | | 2 | Value1_alarm2 | Alarm | Value1 over limit | 1 | Thu, 16 Jun 2022 08:48:17,259 |

Kuva 84. AlarmReport-raportointiajon tuottama raportti

5 RAPORTOINTIALUSTOJEN SOVELTUVUUSTARKASTELU

Opinnäytetyön tavoitteena oli tehdä suunnitelma tiedonkeruun ja raportoinnin osalta Siemens WinCC Unified -valvomoympäristössä. Opinnäytetyön tehtävänanto ei sisältänyt varsinaisia käyttötapauksia, joiden pohjalta alustoja tulisi tarkastella. Täten suunnitelma tehdään saman ominaisuuslistauksen pohjalta kuin kartoitustyökin ja katetaan tämän kappaleen yleisluontoisella tarkastelulla. Kappaleessa tarkastellaan miten kartoitustyötä varten luodun listan ominaisuudet toteutuvat eri raportointialustoille ja tehdään niistä johtopäätöksiä.

Process Management -lisäosat

Tässä kappaleessa vertailu tehdään PM-QUALITY- ja PM-ANALYZE-sovelluksien välillä. PM-SERVER-sovelluksen toimintoja hyödynnetään molempien lisäosien kanssa, joten vertailua ei luonnollisesti tehdä lisäosien ja PM-SERVER-sovelluksen välillä. Lisäosien osalta voidaan tarkastella kaikkia ominaisuuksia, jotka olivat kartoitustyötä varten määritellyllä listalla.

PM-QUALITY- ja PM-SERVER-sovellukset yhdessä tarjoavat kattavimmat mahdollisuudet tiedon jalostamiselle. PM-SERVER-sovelluksen toiminnot mahdollistavat tiedonjalostamisen hyvinkin pitkälle. PM-QUALITY-sovellus täydentää tätä muodostamalla yksittäisistä prosessimuuttujista tuotantoyksiköjä ja laitososastoja. Jalostaminen vaatii kuitenkin kohdeprosessin syvällistä tuntemista sekä mahdollisesti tilastotieteellisten sekä matemaattisten menetelmien hallintaa. Siemens esittelee lähdemateriaaleissaan joitakin laskentamenetelmiä, mutta nämä eivät tule valmiina pohjina sovelluksen mukana. PM-QUALITY-sovelluksen mukana tulevat ohjelmointilohkot PM-SERVER-sovelluksen Calculations-toiminnolle sisältävät myös toimintoja raportointiin ja tiedon jalostukseen, mutta näistä on ainoastaan hyvin lyhyet selitteet lähdemateriaalissa.

PM-QUALITY-sovelluksen Report Editor -toiminnon soveltuvuutta raportoinnin toteuttamiselle jouduttiin arvioimaan pelkän lähdemateriaalin pohjalta, mutta ominaisuuksiltaan toiminto vaikuttaa kattavalta. Vaaditut hälytykset sekä prosessitieto saadaan havainnollistettua, mutta ulkoasultaan miellyttävän raportin toteuttamisen vaativuudesta ei voida tehdä johtopäätöksiä. Raportit voidaan

tallentaa useassa eri formaatissa haluttuun sijaintiin ja välittää tarvittaessa myös sähköpostilla. Raportointiominaisuuksien hyödyntäminen vaatii logiikkaohjelman sovellukseen lisää tageja kuvaamaan tuotantoyksiköiden ja laitteiden tilaa, muttei varsinaisia muutoksia, jotka vaikuttaisivat logiikkaohjelman prosessinohjauksen sovellusosaan.

PM-SERVER-sovellus mahdollistaa myös itsessään yksinkertaisten raporttien ajamisen Calculations-toiminnon avulla. Varsinaisten raporttien rakenteesta ei ole lähdemateriaalin pohjalta tietoa, mutta prosessitapahtuman kirjaaminen tiedostoon tai lähettäminen sähköpostilla onnistuu periaatteellisesti.

PM-ANALYZE-sovelluksen vahvuudet ovat enemmän analyysityökaluna kuin raportoinnin toteuttamisessa. Työtiloista analyysituloksien siirtäminen on periaatteessa mahdollista CSV- ja XML-formaateissa, mutta Siemens (2019a) itsekin hyödyntää PM-ANALYZE-sovelluksen raportointiesimerkissään pääosin PM-SERVER-sovelluksen toimintoja. PM-ANALYZE-sovelluksen REST-rajapinta-arkkitehtuuriin perustuva PM-API-rajapinta tuo lisämahdollisuuksia raportoinnin toteuttamiseen. Se on myös ilmainen verrattuna PM-QUALITY-sovelluksen PM-MES-rajapintaan.

PM-lisäosien skaalautuvuus periytyy lopulta WinCC Unified -valvomo-sovelluksen lisenssiltä. Ainoastaan PM-QUALITY-sovelluksen Standard- ja Professional-versioilla on viitteellinen vaikutus skaalautuvuuteen, mutta aikaperusteisen raportoinnin toteuttaminen Report Editor -toiminnolla vaatii käytännössä Professional-version. Siemens käyttää tätä menetelmää myös omassa esimerkissään (Siemens 2016, 10).

PM-SERVER-sovellus sekä PM-lisäosat voidaan asentaa erillisiin työasemiin ja täten niiden käyttö ei häiritse valvomotyöskentelyä. Kuormittavuuden kannalta valvomoasemaan täytyy ainoastaan asentaa PM-AGENT-tiedonsiirtorajapinnan palvelut. Excel-liitännäisten tiedonsiirtomenetelmistä ei ole lähdemateriaalissa mainintaa, mutta ne hyödyntävät PM-SERVER-sovelluksen ja PM-lisäosien arkistoja, jolloin eivät myöskään kuormita itse valvomoasemaa.

Tiedonsiirto tapahtuu PM-SERVER-sovelluksen ja WinCC Unified -valvomosovelluksen välillä PM-AGENT-tiedonsiirtorajapinnalla. Sen voidaan katsoa olevan suljettu rajapinta, koska sen toiminnasta on saatavilla ainoastaan tieto, että se käyttää tiedonsiirroissaan TCP/IP-protokollapinoa, mutta ei esimerkiksi sen käyttämää porttiavaruutta tai salausprotokollaa. Näiden tarkastelu on tosin rajattu opinnäytetyön ulkopuolelle, mutta ovat oleellisia kysymyksiä, mikäli tiedonsiirtoa pitää suorittaa palomuurin läpi toimistoverkon puolelle.

PM-QUALITY- ja PM-ANALYZE-sovelluksien Excel-liitännäisten ominaisuuksien kartoittaminen on jouduttu tekemään suppeasta määrästä lähdemateriaalia. PM-QUALITY-sovelluksen liitännäinen on kattavampi, koska se mahdollistaa sekä aika- että eräperusteisten raporttien toteuttamisen, muutoin liitännäiset vaikuttavat identtisiltä. Molemmat liitännäiset täyttävät vaatimuksen prosessitiedon ja hälytyksien raportoinnista. Raportointiajo tehdään PM-SERVER-sovelluksen Excel-ohjelmointilohkolla, jolla raportit voidaan tallentaa haluttuun sijaintiin useassa eri formaatissa tai lähettää sähköpostilla.

Rajapintatoteutukset

Sovellusohjelmointi on rajattu tämän työn tarkastelun ulkopuolelle, joten kartoitustyötä varten määriteltyjen ominaisuuksien tarkastelu on huomattavasti suppeampaa rajapintojen osalta. Samassa toteutuksessa voidaan periaatteessa hyödyntää kahtakin eri rajapintaa, täten rajapintoja ei vertailla tässä kappaleessa toisiinsa. Rajapinnat ovat käytettävissä suoraan WinCC Unified -valvomosovelluksen lisenssin kautta ja niiden skaalautuvuus periytyy suoraan myös tältä lisenssiltä.

Kaikilla rajapinnoilla on pääsy tietoon, jolla voidaan täyttää asetettu minimivaatimus prosessitiedon ja hälytyksien esittämisestä. Jokaisessa rajapinnassa on myös kaksisuuntaisia toimintoja. Siemensin rajapinnoilla voidaan manipuloida vähintään tagien arvoja valvomosovelluksessa. ODBC-rajapinnalla voidaan ainoastaan manipuloida valvomosovelluksen arkistoja, minkä kanssa on huomioitava, pystyykö valvomosovellus tulkitsemaan tällaisia muutoksia. Toiminnallisesti isoin rajoite on Open Pipe -rajapinnassa, jolla ei ole pääsyä valvomosovelluksen arkistoihin. Vastaavasti Runtime Openness -rajapintaa voidaan hyödyntää sekä raportointiin että valvomosovelluksen hallintaan. ODBC-

rajapinnan kautta voidaan luonnollisesti hallinnoida vain arkistoja, ei kirjoittaa tai lukea valvomosovelluksen tagien arvoja.

Open Pipe- ja Runtime Openness -rajapintoja hyödyntävät isäntäsovellukset ajetaan paikallisesti WinCC Unified -valvomosovelluksen yhteydessä. Raportointi voidaan suorittaa erillisellä ohjelmalla ja mahdollinen tiedonsiirto tähän toteuttavaan sovellukseen on osa sovellusohjelmointia, joka on rajattu tämän työn ulkopuolelle. ODBC-rajapinnan tiedonsiirto riippuu käytettävästä tietokantajärjestelmästä ja sen eri vaihtoehdot esiteltiin työn kartoitusvaiheen yhteydessä.

Markkinoilta löytyy lukuisia prosessiteollisuudelle tarkoitettuja raportointi- ja analyysisovelluksia, jotka hyödyntävät eri rajapintoja. OPC- ja ODBC-rajapinnat ovat käytössä yleisemmin, mutta myös valmistajien omia rajapintoja hyödyntäviä sovelluksia on saatavilla. Ontecilla jo käytössä oleva Ocean Data Systemsin Dream Report on yksi tällainen. Muita vaihtoehtoja olisivat esimerkiksi Open Automation Software Platform tai SyTech XLReporter. Näiden sovelluksien yhteydessä ei suoraan nimetä, mitä Siemensin rajapintoja ne hyödyntävät. Mainintana on yleisluontoisesti Siemens SDK tai pelkästään Siemens. Esimerkiksi Unified Runtime Openness -rajapintaa ei löytynyt eriteltyinä. Opinnäytetyön laajuuden rajaamiseksi näiden sovelluksien käsittely on rajattu sen ulkopuolelle, mutta niistä löytyy myös kattavia toimintoja tiedonjalostukseen, jotka pystyvät kattamaan laitos- ja yritystason tietotarpeita. Periaatteellisesti Runtime Openness -rajapintaa hyödyntävällä kolmannen osapuolen raportointisovelluksella voidaan saavuttaa samat analysointi- ja raportointimahdollisuudet kuin PM-lisäosilla.

Raportointilisäosa WinCC Unified Report Execution

Raporttien toteuttaminen on suoraviivaisinta Report Execution -työkalulla. Se toimitetaan ilmaiseksi WinCC Unified -valvomosovelluksen mukana ja skaalautuu valvomosovelluksen lisenssin mukaisesti. Raportointipohjia on helppoa ja nopeaa toteuttaa työkalun Excel-liitännäisellä, vaikkakin sen käyttöliittymä on hieman intuitiivinen. Raporttiajon konfigurointi onnistuu suoraan valvomokäyttöliittymän Reports Control -objektilla. Täten raportoinnin toteuttaminen on täysin loppukäyttäjän tehtävissä. Tältä pohjalta voidaan harkita, tarjotaanko

tätä alustaa hyödyntäen varsinaista raportointia vaiko pelkkää sen käyttökoulutusta.

Raporttipohjien toteuttaminen ei häiritse valvomotyöskentelyä, koska työkalu voidaan asentaa erilleen valvomoasemasta. Valvomonsovelluksen ja työkalun välinen tiedonsiirto on suojattu HTTPS-yhteydellä. Raportit voidaan tallentaa haluttuun sijaintiin PDF- tai XLSX-formaatissa tai lähettää sähköpostitse Reports Control -objektin toiminnallisuuksilla.

Raporteissa voidaan esittää hälytyksiä sekä prosessitietoa, mikä täyttää tarkastelulle asetetut minimivaatimukset. Raporttien tietosisältöä on mahdollista jalostaa pidemmälle ja raporttien visuaalista ilmettä parantaa Excelin omilla toiminnoilla, kuten kaavoilla ja kuvaajilla. Työkalun ominaisuuksia testattaessa kuitenkin havaittiin, että tämä tapahtuu useimmiten yrityksen ja erehdyksen kautta. Esimerkiksi muutokset raporttipohjan segmenteissä poistavat alueen Excelin välilehdeltä, jolloin tähän tehdyt soluviittaukset häviävät. Siemens (2022d, 518–519) tarjoaa joitakin vinkkejä raporttien muotoiluun, muttei Excelin toimintojen hyödyntämiseen.

Yleiset

Yleisen skaalautuvuuden näkökulmasta PM-lisäosat ja Unified Runtime Openness -rajapinta yhdessä kolmannen osapuolen raportointisovelluksen soveltuvat ison mittakaavankin projekteihin. Rajapintoja hyödyntävät itsetoteutetut sovellukset ovat myös vaihtoehtoja. Huomioiden sovelluskehityksen vaatiman ajan sekä vastuun ylläpidosta ja tietoturvasta, niiden voidaan olettaa olevan parempia pienemmän mittakaavan projekteihin. Report Execution -työkalu on yhtä lailla pienemmän mittakaavan projekteihin. ODBC-rajapinta on kahden edellisen väliltä. Sen osalta itsetoteutettu sovellus vastaa em. huomioita. Kolmannen osapuolen raportointisovelluksien yhteydessä pitää tarkastella, osaatko ne tulkita suoraan WinCC Unifiedin arkistointirakennetta tai kuinka haastavaa se on konfiguroida.

Tiedon hierarkiaan vasten tarkasteltuna kaikilla alustoilla voidaan toteuttaa hierarkkisesti alemman tason raportointia. PM-lisäosista löytyvät valmiit toiminnot myös korkeamman hierarkiatason raporttien toteuttamiseen, mutta ne

nojaavat vahvasti ihmisten toteuttamiin määrittelyihin ja laskentapohjiin, eikä niissä ole esimerkiksi koneoppimista sisältäviä sovellusosia. Verkottuneisuus on samoin keskitettyä tiedonkeruuta PM-SERVER-sovelluksen alle. Täten voidaan katsoa, että alustoilla saavutetaan ennemminkin Datasta liiketoiminta-osaamiseen -mallin palvelutasoja kuin että ne olisivat osana kyberfyysistä tuotantojärjestelmän ylempiä tasoja.

6 YHTEENVETO

Opinnäytetyö oli itsenäinen projekti, jonka keskeisin sisältö on kappaleen 4. kartoitustyö raportointialustojen ominaisuuksista. Työn sisältö muodostui kappaleeseen 4. saatavilla olevan lähdemateriaalin pohjalta sekä kirjallisuusaiheista, joiden koettiin oleellisesti liittyvän raportoinnin toteuttamiseen. Tehtävänäntoon sisältynyt suunnitelmaosuus on katettu kappaleen 5. yhteenvedolla ja siinä itsenäisesti tehdyillä johtopäätöksillä.

Kirjallisuusosiossa käsitellään asioita, joilla ei välttämättä ole lopulta näkyvää roolia työn keskeisessä sisällössä, mutta ovat hyvin oleellisia kokonaisuuden ymmärtämisen näkökulmasta. Osio kuvaa myös sitä oppimisprosessia, mikä on kuljettu tämän projektin aikana. Ymmärryksen syvyyttä on pyritty osoittamaan asioiden lähestymiskulmalla sekä itse luoduilla havainnollistavilla kuvilla.

Osuus, joka ei suoranaisesti näy raportin sisällössä, on itsenäisesti toteutettu valvomosovelluksen käyttöönotto. Tällä oli kuitenkin ehdoton lisäarvo työn uskottavuuden kannalta, koska alustoista Open Pipe -rajapinnan ja Report Execution -lisäosan toimintoja päästiin testaamaan. Lisäksi valvomosovelluksen toimintaa ja arkistointia voitiin tarkastella omakätisesti.

Opinnäytetyötä tehdessä syntyneen vaikutelman pohjalta PM-QUALITY-sovelluksen voi varauksetta hankkia kokeilukäyttöön. Se on tarkastelluista alustoista kattavin kokonaisratkaisu niin asiakirjamuotoiseen kuin taulukkopohjaiseen raportointiin ylittäen alustoille asetetut minimivaatimukset. Report Execution -raportointilisäosan tarjoaminen loppukäyttäjälähtöisenä raportointialustana on myös tarkastelemisen arvoinen vaihtoehto.

Opinnäytetyö oli hyvä oppimisprosessi, ei ainoastaan siinä käsiteltyjen asioiden ymmärtämisen kannalta, mutta myös vapaasti rajatun kartoitusprojektin hallinnan ja koordinoinnin kannalta. Työn eri aihealueita olisi mahdollista vielä syventää ottamalla mukaan kolmannen osapuolen raportointisovellukset tai toteuttamalla alustojen soveltuvuustarkastelun Datasta liiketoimintaosaamiseen -mallin pohjalta.

LÄHTEET

Ackoff, R. 1989. From data to Wisdom. *Journal of applied systems analysis* 16, 3–9. PDF-dokumentti. Saatavissa: <http://www-public.imtbs-tsp.eu/~gibson/Teaching/Teaching-ReadingMaterial/Ackoff89.pdf> [viitattu 11.5.2022].

Ahonen, T., Kortelainen, H. & Kunttu, S. 2017. Teollisen internetin palveluista liiketoimintaa. Teoksessa Kärri, T. & Martinsuo, M. (toim.) *Teollinen internet uudistaa palveluliiketoimintaa ja kunnossapitoa*. 1. painos. Helsinki: Kunnossapitoyhdistys Promaint Oy, 15–84. PDF-dokumentti. Saatavissa: <https://urn.fi/URN:NBN:fi:tyy-201706061586> [viitattu 12.5.2022].

Australian Bureau of Statistics s.a. Statistical Language. WWW-dokumentti. Saatavissa: <https://www.abs.gov.au/websitedbs/D3310114.nsf/Home/Statistical+Language> [viitattu 11.5.2022].

AVEVA. 2020. AVEVA InTouch HMI Datasheet. PDF-dokumentti. Saatavissa: https://www.aveva.com/content/dam/aveva/documents/datasheets/Datasheet_AVEVA_InTouchHMI_2020_EN.pdf [viitattu 22.7.2022].

Awadh, E. & Ghaziri H. 2003. *Knowledge Management*. Lontoo: Pearson Education.

Bagheri, B., Kao, H. & Lee, J. 2014. A Cyber-Physical Systems architecture for Industry 4.0 -based manufacturing systems. *Manufacturing Letters* 3, 18–23. PDF-dokumentti. Saatavissa: <http://doi.org/10.1016/j.mfglet.2014.12.001> [viitattu 13.5.2022].

Berge, J. 2005. *Software for Automation: Architecture, Integration, and Security*. Durham: ISA - International Society of Automation.

Bowne-Anderson, H. 2018. Your Data Literacy Depends on Understanding the Types of Data and How They're Captured. WWW-dokumentti. Saatavissa: <https://hbr.org/2018/10/your-data-literacy-depends-on-understanding-the-types-of-dataand-how-theyre-captured> [viitattu 18.5.2022].

Carter, P. 2018. *SQL Server Advanced Data Types. JSON, XML and beyond*. New York: Apress. E-kirja. Saatavissa: <https://doi.org/10.1007/978-1-4842-3901-8> [viitattu 24.5.2022].

Damm, M., Leitner, S.H. & Mahnke, W. 2009. *OPC unified architecture*. Berliini: Springer-Verlag. E-kirja. Saatavissa: <https://doi.org/10.1007/978-3-540-68899-0> [viitattu 22.5.2022].

Dey, C. & Sen, S.K. 2020. *Industrial Automation Technologies*. 1. painos. Florida: CRC Press. E-kirja. Saatavissa: <https://doi.org/10.1201/9780429299346> [viitattu 15.5.2022].

Difference between an API and a device driver. 2015. Keskusteluryhmän artikkeli. Päivitetty: 1.7.2015. Saatavissa: <https://stackoverflow.com/questions/31167526/difference-between-an-api-and-a-device-driver> [viitattu 19.5.2022].

Fall, K.R. & Stevens W.R. 2012. TCP/IP Illustrated. Volume 1: The Protocols. 2. painos. Boston: Addison-Wesley.

Geiger, K. 1995. Inside ODBC. Washington: Microsoft Press.

Gobinath, V.M. 2021. An Overview of Industry 4.0 Technologies and Benefits and Challenges That Incurred While Adopting It. Teoksessa Arockiarajan, A., Duraiselvam, M. & Ramesh, R. (toim.) *Advances in Industrial Automation and Smart Manufacturing*. Berliini: Springer Nature, 1–12. E-kirja. Saatavissa: <https://doi.org/10.1007/978-981-15-4739-3> [viitattu 12.5.2022].

Goller, V. 2019. Looking Inside Real-Time Ethernet. PDF-dokumentti. Saatavissa: <https://www.analog.com/media/en/analog-dialogue/volume-53/number-1/looking-inside-real-time-ethernet.pdf> [viitattu 20.5.2022].

Ibrahim, R.N., Lochert P.B. & Tahera, K. 2008. A fuzzy logic approach for dealing with qualitative quality characteristics of a process. *Expert Systems with Applications* 34, 2630–2638. PDF-dokumentti. Saatavissa: <https://doi.org/10.1016/j.eswa.2007.05.025> [viitattu 11.5.2022].

Industrial Automation Systems SIMATIC s.a. Siemens. WWW-dokumentti. Saatavissa: <https://new.siemens.com/global/en/products/automation/systems/industrial.html> [viitattu 19.6.2022].

Jasperneite, J., Sauter, T. & Wollschlaeger, M. 2017. The Future of Industrial Communication. Automation Networks in the Era of the Internet of Things and Industry 4.0. *IEEE Industrial Electronics Magazine*. 17–27. PDF-dokumentti. Saatavissa: <https://doi.org/10.1109/MIE.2017.2649104> [viitattu 16.5.2022].

Kärri, T., Marttonen-Arola, S., Kinnunen, S-K., Kortelainen, H., Kunttu, S., Hanski, J., Ahonen, T., Valkokari, P., Uusitalo, T. & Horn, S. 2017. Fleet-based industrial data symbiosis. *DIMECC Publications series* 19, 129–140. PDF-dokumentti. Saatavissa: <https://cris.vtt.fi/en/publications/> [viitattu 12.5.2022].

Käyhty, E. 2022. Automaatioasiantuntija. Henkilökohtainen tiedonanto 9.5.–27.6.2022. Ontec Oy.

Manoochehri, M. 2013. Data Just Right. Introduction to Large-Scale Data & Analytics. Boston: Addison-Wesley.

Microsoft s.a. Date systems in Excel. WWW-dokumentti. Saatavissa: <https://support.microsoft.com/en-us/office/date-systems-in-excel-e7fe7167-48a9-4b96-bb53-5612a800b487> [viitattu 7.6.2022].

Microsoft. 2022. Microsoft ODBC Driver for SQL Server. WWW-dokumentti. Päivitetty 15.2.2022. Saatavissa: <https://docs.microsoft.com/en-us/sql/connect/odbc/microsoft-odbc-driver-for-sql-server> [viitattu 10.6.2022].

Natri, P. 2019. Teollisten järjestelmien ja pilvipohjaisten sovellusten väliset integraatiomenetelmät. Tampereen yliopisto. Tekniikan ja luonnontieteiden tiedekunta. Diplomityö. PDF-dokumentti. Saatavissa: <https://urn.fi/URN:NBN:fi:tyy-201905211706> [viitattu 20.5.2022].

Ocean Data Systems. 2020a. Dream Report Main Brochure. PDF-dokumentti. Saatavissa: <https://dreamreport.net/wp-content/uploads/DR-Brochure-2020-En.pdf> [viitattu 22.7.2022].

Ocean Data Systems. 2020b. Dream Report Communication Drivers. Release 2. PDF-dokumentti. Saatavissa: <https://dreamreport.net/wp-content/uploads/Dream-Report-Communication-Drivers-2020-R2-1.pdf> [viitattu 22.7.2022].

Ontec Oy. 2022a. Yritys-esite. PDF-dokumentti. Saatavissa: <https://www.ontec.fi/app/uploads/2022/03/Ontec-Oy-FI-2022.pdf> [viitattu 23.5.2022].

Ontec Oy. 2022b. Tuotteet. PDF-dokumentti. Saatavissa: <https://www.ontec.fi/app/uploads/2022/03/Ontec-Oy-Tuotteet-FI-2022.pdf> [viitattu 17.6.2022].

Ontec Oy. 2022c. Tavoittemme on maksimoida tuotteen elinkaari asiakkaan käyttöön. WWW-dokumentti. Saatavissa: <https://www.ontec.fi/2022/04/04/yritysturvallisuus-hallussa-onteccybersecurity-2/> [viitattu 17.6.2022].

OPC CLASSIC s.a. IPCOMM. WWW-dokumentti. Saatavissa: <https://www.ipcomm.de/protocol/OPC/en/sheet.html> [viitattu 22.5.2022].

OPC UA s.a. IPCOMM. WWW-dokumentti. Saatavissa: <https://www.ipcomm.de/protocol/OPCUA/en/sheet.html> [viitattu 20.5.2022].

Pitkänen, H. 2016. Rajapintojen käyttö tiedonsiirrossa. Itä-Suomen yliopisto. Luonnontieteiden ja metsätieteiden tiedekunta. Pro gradu -työ. PDF-dokumentti. Saatavissa: <http://urn.fi/urn:nbn:fi:uef-20170032> [viitattu 20.5.2022].

PM Add-ons for WinCC s.a. Siemens. WWW-dokumentti. Saatavissa: <https://new.siemens.com/global/en/products/automation/industry-software/automation-software/scada/pm-add-ons.html> [viitattu 26.5.2022].

Profinet University s.a. PROFINET Communication Channels. WWW-dokumentti. Saatavissa: <https://profinetuniversity.com/profinet-basics/profinet-communication-channels/> [viitattu 20.5.2022].

Rinaldi, J. 2017. The End of Cyclic Communication... WWW-dokumentti. Saatavissa: <https://www.rtautomation.com/rtas-blog/the-end-of-cyclic-communication/> [viitattu 20.5.2022].

Rowley, J. 2007. The wisdom hierarchy: representations of the DIKW hierarchy. *Journal of Information Science* 33, 163–180. PDF-dokumentti. Saatavissa: <https://doi.org/10.1177/0165551506070706> [viitattu 12.5.2022].

Sanastokeskus. 2014. Rajapinta. WWW-dokumentti. Saatavissa: <http://www.tsk.fi/tsk/termitalkoot/fi/hakemistot-267.html> [viitattu 21.5.2022].

SFS-EN 13306. 2017. Kunnossapito. Kunnossapidon terminologia. Helsinki: Suomen Standardisoimisliitto.

- Siemens. 2014. SIMATIC S7-1200. System manual. PDF-dokumentti. Saatavissa: <https://support.industry.siemens.com/cs/document/91696622/> [viitattu 25.6.2022].
- Siemens. 2016. PM-QUALITY. Order/Batch-oriented Archiving and Reporting. 10. versio. PDF-dokumentti. Saatavissa: <https://support.industry.siemens.com/cs/document/109778976/> [viitattu 26.5.2022].
- Siemens. 2018. PM-ANALYZE. Water / Wastewater reporting. PDF-dokumentti. Saatavissa: <https://new.siemens.com/se/sv/branschspecifika-losningar/va/pm-analyze.html> [viitattu 26.5.2022].
- Siemens. 2019a. Calculating OEE using PM-ANALYZE. PDF-dokumentti. Saatavissa: <https://support.industry.siemens.com/cs/document/109769969/> [viitattu 28.5.2022].
- Siemens. 2019b. Totally Integrated Automation Portal. WWW- ja PDF-dokumentti. Saatavissa: <https://new.siemens.com/global/en/products/automation/industry-software/automation-software/tia-portal.html> [viitattu 26.6.2022].
- Siemens. 2020a. PM-ANALYZE. Analysis of Alarms and process values. 9. versio. PDF-dokumentti. Saatavissa: <https://support.industry.siemens.com/cs/document/109782856/> [viitattu 26.5.2022].
- Siemens. 2020b. PM-QUALITY. Getting Started. PDF-dokumentti.
- Siemens. 2020c. SIMATIC WinCC Unified System. PDF-dokumentti. Saatavissa: <https://assets.new.siemens.com/siemens/assets/api/uuid:cf0c75b3-32cb-4fdc-a226-2c1d379dfec7/simatic-wincc-unified-pc-webinar-110520.pdf> [viitattu 2.6.2022].
- Siemens. 2020d. Creating applications for WinCC Unified Runtime Openness. PDF-dokumentti. Saatavissa: <https://support.industry.siemens.com/cs/document/109781803/> [viitattu 3.6.2022].
- Siemens. 2020e. PM-MES Interface V1.7. What's New. PDF-dokumentti. Saatavissa: <https://support.industry.siemens.com/cs/document/109798673/> [viitattu 31.7.2022].
- Siemens. 2021a. Process Management System Redundancy scenarios. PDF-dokumentti. Saatavissa: <https://support.industry.siemens.com/cs/document/109779255/> [viitattu 28.5.2022].
- Siemens. 2021b. Delivery release PM-QUALITY V10.8. PDF-dokumentti. Saatavissa: <https://support.industry.siemens.com/cs/document/109793799/> [viitattu 28.5.2022].
- Siemens. 2021c. WinCC Unified Runtime API (ODK). System Manual. PDF-dokumentti. Saatavissa: <https://support.industry.siemens.com/cs/document/109803795/> [viitattu 3.6.2022].

Siemens. 2021d. WinCC Unified Open Pipe. Operating Manual. PDF-dokumentti. Saatavissa: <https://support.industry.siemens.com/cs/document/109803794/> [viitattu 3.6.2022].

Siemens. 2021e. Configuring Logging for SIMATIC WinCC Unified Systems. PDF-dokumentti. Saatavissa: <https://support.industry.siemens.com/cs/document/109782859/> [viitattu 7.6.2022].

Siemens. 2021f. STEP 7 and WinCC Engineering V17 System Manual. PDF-dokumentti. Saatavissa: <https://support.industry.siemens.com/cs/document/109798671/> [viitattu 21.6.2022].

Siemens. 2021g. SIMATIC S7-1500/ET 200MP Manual Collection. PDF-dokumentti. Saatavissa: <https://support.industry.siemens.com/cs/document/86140384/> [viitattu 25.6.2022].

Siemens. 2021h. Human Machine Interface Systems/PC-based Automation. PDF-dokumentti. Saatavissa: <https://new.siemens.com/global/en/products/automation/simatic-hmi.html> [viitattu 30.6.2022].

Siemens. 2022a. What system scope is available for the various PM-Add-ons? PDF-dokumentti. Saatavissa: <https://support.industry.siemens.com/cs/document/109779954/> [viitattu 26.5.2022].

Siemens. 2022b. Release notes for PM-ANALYZE Version 10.0. PDF-dokumentti. Saatavissa: <https://support.industry.siemens.com/cs/document/109806258/> [viitattu 28.5.2022].

Siemens. 2022c. Release notes for PM-QUALITY Version 11.0. PDF-dokumentti. Saatavissa: <https://support.industry.siemens.com/cs/document/109807767/> [viitattu 28.5.2022].

Siemens. 2022d. SIMATIC HMI WinCC Unified Engineering V17. PDF-dokumentti. Saatavissa: <https://support.industry.siemens.com/cs/document/109794204/> [viitattu 16.6.2022].

Siemens. 2022e. Product overview for SIMATIC S7-1200. WWW- ja PDF-dokumentti. Saatavissa: <https://new.siemens.com/global/en/products/automation/systems/industrial/plc/s7-1200.html> [viitattu 25.6.2022].

Siemens. 2022f. Product overview for SIMATIC S7-1500. WWW- ja PDF-dokumentti. Saatavissa: <https://new.siemens.com/global/en/products/automation/systems/industrial/plc/simatic-s7-1500.html> [viitattu 25.6.2022].

SQL ja relaatiotietokannat s.a. Metropolia Ammattikorkeakoulu. Luentosarja. Saatavissa: <https://vw4.viope.com/> [viitattu 25.5.2022].

Thompson, L.M. 2008. Industrial Data Communications. 4. painos. Durham: ISA - International Society of Automation.

Tietokantojen perusteet. 2019. Helsingin yliopisto. Saatavissa: <https://tietokantojen-perusteet-19.mooc.fi/> [viitattu 21.5.2022].

Totally Integrated Automation Portal s.a. Siemens. WWW-dokumentti. Saatavissa: <https://new.siemens.com/global/en/products/automation/industry-software/automation-software/tia-portal.html> [viitattu 19.6.2022].

Werner, C. 2020. SQLite ODBC Driver. WWW-dokumentti. Päivitetty: 20.6.2020. Saatavissa: <http://ch-werner.de/sqliteodbc/html/index.html> [viitattu 10.6.2022].

Åkerman, M. 2018. Implementing Shop Floor IT for Industry 4.0. Chalmersin teknillinen korkeakoulu. Teollisuuden- ja materiaalitieteen tiedekunta. Väitöskirja. PDF-dokumentti. Saatavissa: <https://www.researchgate.net/publication/326224890> Implementing Shop Floor IT for Industry 40 [viitattu 12.5.2022].

KUVALUETTELO

Kuva 1. Tiedon hierarkian perusmalli. Rowley, J. 2007. The wisdom hierarchy: representations of the DIKW hierarchy. *Journal of Information Science* 33, 163–180. PDF-dokumentti. Saatavissa:

<https://doi.org/10.1177/0165551506070706> [viitattu 12.5.2022].

Kuva 2. Tiedon käsitteet taulukkomuotoisessa rakenteessa. Australian Bureau of Statistics s.a. Statistical Language. WWW-dokumentti. Saatavissa:

<https://www.abs.gov.au/websitedbs/D3310114.nsf/Home/Statistical+Language> [viitattu 11.5.2022].

Kuva 3. Tiedon hierarkiamallin laajennus. Rowley, J. 2007. The wisdom hierarchy: representations of the DIKW hierarchy. *Journal of Information Science* 33, 163–180. PDF-dokumentti. Saatavissa:

<https://doi.org/10.1177/0165551506070706> [viitattu 12.5.2022].

Kuva 4. Datasta liiketoimintaosaamiseen -mallin yhteistyötasot visualisoituna. Ahonen, T., Kortelainen, H. & Kunttu, S. 2017. Teollisen internetin palveluista liiketoimintaa. Teoksessa Kärri, T. & Martinsuo, M. (toim.) Teollinen internet uudistaa palveluliiketoimintaa ja kunnossapitoa. 1. painos. Helsinki: Kunnossapitoyhdistys Promaint Oy, 15–84. PDF-dokumentti. Saatavissa:

<https://urn.fi/URN:NBN:fi:tty-201706061586> [viitattu 12.5.2022].

Kuva 5. ISA 95 -automaatiopyramidi ja kyberfyysinen tuotantojärjestelmä. Åkerman, M. 2018. Implementing Shop Floor IT for Industry 4.0. Chalmersin teknillinen korkeakoulu. Teollisuuden- ja materiaalitieteen tiedekunta. Väitöskirja. PDF-dokumentti. Saatavissa: <https://www.researchgate.net/publication/326224890> [Implementing Shop Floor IT for Industry 40](https://www.researchgate.net/publication/326224890) [viitattu 12.5.2022].

Kuva 6. Kyberfyysinen järjestelmän 5C-malli. Bagheri, B., Kao, H. & Lee, J. 2014. A Cyber-Physical Systems architecture for Industry 4.0 -based manufacturing systems. *Manufacturing Letters* 3, 18–23. PDF-dokumentti. Saatavissa: <http://doi.org/10.1016/j.mfglet.2014.12.001> [viitattu 13.5.2022].

Kuva 7. Standardoimattomat ja valmistajakohtaiset rajapinnat. Natri, P. 2019. Teollisten järjestelmien ja pilvipohjaisten sovellusten väliset integraatiomenetelmät. Tampereen yliopisto. Tekniikan ja luonnontieteiden tiedekunta. Diplomityö. PDF-dokumentti. Saatavissa: <https://urn.fi/URN:NBN:fi:tty-201905211706> [viitattu 20.5.2022].

Kuva 8. Avoimet rajapinnat. Natri, P. 2019. Teollisten järjestelmien ja pilvipohjaisten sovellusten väliset integraatiomenetelmät. Tampereen yliopisto. Tekniikan ja luonnontieteiden tiedekunta. Diplomityö. PDF-dokumentti. Saatavissa: <https://urn.fi/URN:NBN:fi:tty-201905211706> [viitattu 20.5.2022].

Kuva 9. OSI-malli. Thompson, L.M. 2008. Industrial Data Communications. 4. painos. Durham: ISA - International Society of Automation.

Kuva 10. OSI-, Internet- ja Kenttäväylä-mallit. Goller, V. 2019. Looking Inside Real-Time Ethernet. PDF-dokumentti. Saatavissa: <https://www.analog.com/media/en/analog-dialogue/volume-53/number-1/looking-inside-real-time-ethernet.pdf> [viitattu 20.5.2022].

Kuva 11. OSI-malli ja yleisiä teollisuusympäristön tiedonsiirtomenetelmiä. Goller, V. 2019. Looking Inside Real-Time Ethernet. PDF-dokumentti. Saatavissa: <https://www.analog.com/media/en/analog-dialogue/volume-53/number-1/looking-inside-real-time-ethernet.pdf> [viitattu 20.5.2022].

Kuva 11. OSI-malli ja yleisiä teollisuusympäristön tiedonsiirtomenetelmiä. OPC UA s.a. IPCOMM. WWW-dokumentti. Saatavissa: <https://www.ipcomm.de/protocol/OPCUA/en/sheet.html> [viitattu 20.5.2022].

Kuva 11. OSI-malli ja yleisiä teollisuusympäristön tiedonsiirtomenetelmiä. Profinet University s.a. PROFINET Communication Channels. WWW-dokumentti. Saatavissa: <https://profinetuniversity.com/profinet-basics/profinet-communication-channels/> [viitattu 20.5.2022].

Kuva 12. Väylätasot ja tiedonsiirron periaatteet automaatiopyramidissa. Dey, C. & Sen, S.K. 2020. Industrial Automation Technologies. 1. painos. Florida: CRC Press. E-kirja. Saatavissa: <https://doi.org/10.1201/9780429299346> [viitattu 15.5.2022].

Kuva 13. OPC DA ja OSI-malli. OPC CLASSIC s.a. IPCOMM. WWW-dokumentti. Saatavissa: <https://www.ipcomm.de/protocol/OPC/en/sheet.html> [viitattu 22.5.2022].

Kuva 14. OPC UA ja OSI-malli. Damm, M., Leitner, S.H. & Mahnke, W. 2009. OPC unified architecture. Berliini: Springer-Verlag. E-kirja. Saatavissa: <https://doi.org/10.1007/978-3-540-68899-0> [viitattu 22.5.2022].

Kuva 14. OPC UA ja OSI-malli. OPC UA s.a. IPCOMM. WWW-dokumentti. Saatavissa: <https://www.ipcomm.de/protocol/OPCUA/en/sheet.html> [viitattu 20.5.2022].

Kuva 15. Prosessitietoa CSV-formaatissa ja taulukkomuodossa.

Kuva 16. Taulukkomuotoinen rivi yhtenä XML-formaatin elementtinä.

Kuva 17. XML-formaatin elementti attribuuteilla.

Kuva 18. Taulukkomuotoinen rivi yhtenä JSON-formaatin elementtinä.

Kuva 19. JSON-formaatti hyödyntäen eri tietotyyppejä.

Kuva 20. Valmistajasidonnaisuuden ongelma rajapinnoissa. Geiger, K. 1995. Inside ODBC. Washington: Microsoft Press.

Kuva 21. ODBC-arkkitehtuuri. Geiger, K. 1995. Inside ODBC. Washington: Microsoft Press.

Kuva 22. Taulun rakenneosat ja pääavaimen määrittely.

Kuva 23. Tietojen yhdistäminen relaatiotietokannassa.

Kuva 24. Ontecin periaatemalli tiedonsiirrosta tuotemoduulien kanssa. Ontec Oy. 2022c. Tavoitteemme on maksimoida tuotteen elinkaari asiakkaan käyttöön. WWW-dokumentti. Saatavissa: <https://www.ontec.fi/2022/04/04/yritysturvallisuus-hallussa-onteccybersecurity-2/> [viitattu 17.6.2022].

Kuva 25. Periaatemalli tarkasteltavasta ohjausjärjestelmäympäristöstä. Käyhty, E. 2022. Automaatioasiantuntija. Henkilökohtainen tiedonanto 9.5.–27.6.2022. Ontec Oy.

Kuva 26. Logiikkajärjestelmän ohjelmasykli. Dey, C. & Sen, S.K. 2020. Industrial Automation Technologies. 1. painos. Florida: CRC Press. E-kirja. Saatavissa: <https://doi.org/10.1201/9780429299346> [viitattu 15.5.2022].

Kuva 27. Siemens Simatic S7-1200 ja laajennusmoduuleja. Siemens. 2022e. Product overview for SIMATIC S7-1200. WWW- ja PDF-dokumentti. Saatavissa: <https://new.siemens.com/global/en/products/automation/systems/industrial/plc/s7-1200.html> [viitattu 25.6.2022].

Kuva 28. Prosessin vuorovaikutustasot ja Siemensin käyttöpäätäratkaisut. Siemens. 2021h. Human Machine Interface Systems/PC-based Automation. PDF-dokumentti. Saatavissa: <https://new.siemens.com/global/en/products/automation/simatic-hmi.html> [viitattu 30.6.2022].

Kuva 29. WinCC Unified -alustojen ohjelmointi. Siemens. 2022d. SIMATIC HMI WinCC Unified Engineering V17. PDF-dokumentti. Saatavissa: <https://support.industry.siemens.com/cs/document/109794204/> [viitattu 16.6.2022].

Kuva 30. Arkistoinnin rakenne WinCC Unified -ympäristössä. Siemens. 2021e. Configuring Logging for SIMATIC WinCC Unified Systems. PDF-dokumentti. Saatavissa: <https://support.industry.siemens.com/cs/document/109782859/> [viitattu 7.6.2022].

Kuva 31. Kiertävä tietokantasegmentointi ja segmenttien seuranta. Siemens. 2021e. Configuring Logging for SIMATIC WinCC Unified Systems. PDF-dokumentti. Saatavissa: <https://support.industry.siemens.com/cs/document/109782859/> [viitattu 7.6.2022].

Kuva 32. Syklisen ja muutosperusteisen arkistoinnin ero. Siemens. 2021e. Configuring Logging for SIMATIC WinCC Unified Systems. PDF-dokumentti. Saatavissa: <https://support.industry.siemens.com/cs/document/109782859/> [viitattu 7.6.2022].

Kuva 33. Opinnäytetyötä varten luotu valvomokäyttöliittymä.

Kuva 34. Process Management -lisäosakokonaisuus. Siemens. 2022a. What system scope is available for the various PM-Add-ons? PDF-dokumentti. Saatavissa: <https://support.industry.siemens.com/cs/document/109779954/> [viitattu 26.5.2022].

Kuva 35. Process Management -sovelluksien tiedonsiirto infrastruktuuri. Siemens. 2022a. What system scope is available for the various PM-Add-ons? PDF-dokumentti. Saatavissa: <https://support.industry.siemens.com/cs/document/109779954/> [viitattu 26.5.2022].

Kuva 36. Tiedonsiirto Process Management -sovelluspinossa. Siemens. 2020b. PM-QUALITY. Getting Started. PDF-dokumentti.

Kuva 37. Asennuskonfiguraatiovaihtoehtoja PM-lisäosille. Siemens. 2019a. Calculating OEE using PM-ANALYZE. PDF-dokumentti. Saatavissa: <https://support.industry.siemens.com/cs/document/109769969/> [viitattu 28.5.2022].

Kuva 37. Asennuskonfiguraatiovaihtoehtoja PM-lisäosille. Siemens. 2022a. What system scope is available for the various PM-Add-ons? PDF-dokumentti. Saatavissa: <https://support.industry.siemens.com/cs/document/109779954/> [viitattu 26.5.2022].

Kuva 38. Calculations-toiminnon kokonaisuus. Siemens. 2020a. PM-ANALYZE. Analysis of Alarms and process values. 9. versio. PDF-dokumentti. Saatavissa: <https://support.industry.siemens.com/cs/document/109782856/> [viitattu 26.5.2022].

Kuva 39. Suorituskykyindikaattori laskenta Calculations-toiminnolla. Siemens. 2020a. PM-ANALYZE. Analysis of Alarms and process values. 9. versio. PDF-dokumentti. Saatavissa: <https://support.industry.siemens.com/cs/document/109782856/> [viitattu 26.5.2022].

Kuva 40. Sähköpostin ja tiedoston muodostaminen Calculations-toiminnolla. Siemens. 2022a. What system scope is available for the various PM-Add-ons? PDF-dokumentti. Saatavissa: <https://support.industry.siemens.com/cs/document/109779954/> [viitattu 26.5.2022].

Kuva 41. Tason asetukset ja tuloksien tallentuminen tietokantaan. Siemens. 2022a. What system scope is available for the various PM-Add-ons? PDF-dokumentti. Saatavissa: <https://support.industry.siemens.com/cs/document/109779954/> [viitattu 26.5.2022].

Kuva 42. Kaavoja DayValues-tasolla. Siemens. 2022a. What system scope is available for the various PM-Add-ons? PDF-dokumentti. Saatavissa: <https://support.industry.siemens.com/cs/document/109779954/> [viitattu 26.5.2022].

Kuva 43. Aikataulutoiminto PM-SERVER-sovelluksessa. Siemens. 2018. PM-ANALYZE. Water / Wastewater reporting. PDF-dokumentti. Saatavissa: <https://new.siemens.com/se/sv/branschspecifika-losningar/va/pm-analyze.html> [viitattu 26.5.2022].

Kuva 44. Seurattavat tekstimuuttujat Context acquisition -toiminnossa. Siemens. 2022a. What system scope is available for the various PM-Add-ons? PDF-dokumentti. Saatavissa: <https://support.industry.siemens.com/cs/document/109779954/> [viitattu 26.5.2022].

Kuva 45. Arvonmuutoksien arkistointia Context acquisition -toiminnossa. Siemens. 2022a. What system scope is available for the various PM-Add-ons? PDF-dokumentti. Saatavissa: <https://support.industry.siemens.com/cs/document/109779954/> [viitattu 26.5.2022].

Kuva 46. State Model -toiminto ja esimerkkitapaus Motor_1. Siemens. 2020b. PM-QUALITY. Getting Started. PDF-dokumentti.

Kuva 47. State Model PM-QUALITY-sovelluksen trendinäytössä. Siemens. 2022a. What system scope is available for the various PM-Add-ons? PDF-dokumentti. Saatavissa: <https://support.industry.siemens.com/cs/document/109779954/> [viitattu 26.5.2022].

Kuva 48. PM-QUALITY Standard- ja Professional-versioiden ero. Siemens. 2016. PM-QUALITY. Order/Batch-oriented Archiving and Reporting. 10. versio. PDF-dokumentti. Saatavissa: <https://support.industry.siemens.com/cs/document/109778976/> [viitattu 26.5.2022].

Kuva 49. Yhdistetyt tuotantoyksiköt PM-QUALITY Professional-versiossa. Siemens. 2016. PM-QUALITY. Order/Batch-oriented Archiving and Reporting. 10. versio. PDF-dokumentti. Saatavissa: <https://support.industry.siemens.com/cs/document/109778976/> [viitattu 26.5.2022].

Kuva 50. PM-QUALITY-sovelluksen Topology Manager -toiminnon pääikkuna ja selitteet. Siemens. 2016. PM-QUALITY. Order/Batch-oriented Archiving and Reporting. 10. versio. PDF-dokumentti. Saatavissa: <https://support.industry.siemens.com/cs/document/109778976/> [viitattu 26.5.2022].

Kuva 51. Aktiivisuustila ja perustiedot tuotantoyksikössä. Siemens. 2020b. PM-QUALITY. Getting Started. PDF-dokumentti.

Kuva 52. Vaiheistuksen luominen tuotantoyksikköön. Siemens. 2020b. PM-QUALITY. Getting Started. PDF-dokumentti.

Kuva 53. Vaiheistuksen arvojen tallennus otoksella. Siemens. 2020b. PM-QUALITY. Getting Started. PDF-dokumentti.

Kuva 54. Tallennettavien tagien asetukset otoksessa. Siemens. 2020b. PM-QUALITY. Getting Started. PDF-dokumentti.

Kuva 55. Arkistointi PM-QUALITY-sovelluksessa. Siemens. 2020b. PM-QUALITY. Getting Started. PDF-dokumentti.

Kuva 56. BatchTable-moduuli ja aikavälisuodatus PM-QUALITY-sovelluksessa. Siemens. 2020b. PM-QUALITY. Getting Started. PDF-dokumentti.

Kuva 57. Vaiheistuksia, hälytyksiä ja otoksia TrendCurve Control -moduulissa. Siemens. 2020b. PM-QUALITY. Getting Started. PDF-dokumentti.

Kuva 58. PM-QUALITY-sovelluksen Report Editor -toiminnon perusnäkyvä. Siemens. 2016. PM-QUALITY. Order/Batch-oriented Archiving and Reporting. 10. versio. PDF-dokumentti. Saatavissa: <https://support.industry.siemens.com/cs/document/109778976/> [viitattu 26.5.2022].

Kuva 59. Viittauksen liittäminen tekstikenttään. Siemens. 2020b. PM-QUALITY. Getting Started. PDF-dokumentti.

Kuva 60. Raporttipohjan automaattisen tulostuksen asetukset. Siemens. 2020b. PM-QUALITY. Getting Started. PDF-dokumentti.

Kuva 61. Aikaperusteisen raportoinnin tuotantoyksiköt PM-QUALITY-sovelluksessa. Siemens. 2016. PM-QUALITY. Order/Batch-oriented Archiving and Reporting. 10. versio. PDF-dokumentti. Saatavissa: <https://support.industry.siemens.com/cs/document/109778976/> [viitattu 26.5.2022].

Kuva 62. TimeBatch-ohjelmointilohko PM-SERVER-sovelluksen calculati-ons-toiminnossa. Siemens. 2016. PM-QUALITY. Order/Batch-oriented Archiving and Reporting. 10. versio. PDF-dokumentti. Saatavissa: <https://support.industry.siemens.com/cs/document/109778976/> [viitattu 26.5.2022].

Kuva 63. PM-MES-rajapinnan infrastruktuuri. Siemens. 2020e. PM-MES Interface V1.7. What's New. PDF-dokumentti. Saatavissa: <https://support.industry.siemens.com/cs/document/109798673/> [viitattu 31.7.2022].

Kuva 64. Työtiloja PM-ANALYZE-sovelluksessa. Siemens. 2022a. What system scope is available for the various PM-Add-ons? PDF-dokumentti. Saatavissa: <https://support.industry.siemens.com/cs/document/109779954/> [viitattu 26.5.2022].

Kuva 65. Server-määrittelyjen pääikkuna PM-ANALYZE-sovelluksessa. Siemens. 2020a. PM-ANALYZE. Analysis of Alarms and process values. 9. versio. PDF-dokumentti. Saatavissa: <https://support.industry.siemens.com/cs/document/109782856/> [viitattu 26.5.2022].

Kuva 66. Raportoinnin toimintaperiaate Excel-liitännäisen kanssa. Siemens. 2019a. Calculating OEE using PM-ANALYZE. PDF-dokumentti. Saatavissa: <https://support.industry.siemens.com/cs/document/109769969/> [viitattu 28.5.2022].

Kuva 67. Excel-ohjelmointilohkon asetukset. Siemens. 2019a. Calculating OEE using PM-ANALYZE. PDF-dokumentti. Saatavissa: <https://support.industry.siemens.com/cs/document/109769969/> [viitattu 28.5.2022].

Kuva 68. Liitännäisen muodostamien kaavojen muokkaaminen. Siemens. 2019a. Calculating OEE using PM-ANALYZE. PDF-dokumentti. Saatavissa: <https://support.industry.siemens.com/cs/document/109769969/> [viitattu 28.5.2022].

Kuva 69. PM-ANALYZE:n Excel-liitännäisen työkalurivi. Siemens. 2020a. PM-ANALYZE. Analysis of Alarms and process values. 9. versio. PDF-dokumentti. Saatavissa: <https://support.industry.siemens.com/cs/document/109782856/> [viitattu 26.5.2022].

Kuva 70. PM-QUALITY:n Excel-liitännäisen työkalurivi eräkohtaiseen raportointiin. Siemens. 2016. PM-QUALITY. Order/Batch-oriented Archiving and Reporting. 10. versio. PDF-dokumentti. Saatavissa: <https://support.industry.siemens.com/cs/document/109778976/> [viitattu 26.5.2022].

Kuva 71. Esimerkkiraportteja PM-lisäosan Excel-liitännäisellä. Siemens. 2016. PM-QUALITY. Order/Batch-oriented Archiving and Reporting. 10. versio. PDF-dokumentti. Saatavissa: <https://support.industry.siemens.com/cs/document/109778976/> [viitattu 26.5.2022].

Kuva 71. Esimerkkiraportteja PM-lisäosan Excel-liitännäisellä. Siemens. 2022a. What system scope is available for the various PM-Add-ons? PDF-dokumentti. Saatavissa: <https://support.industry.siemens.com/cs/document/109779954/> [viitattu 26.5.2022].

Kuva 72. Siemensin esimerkkiohjelma Runtime Openness -rajapinnalle. Siemens. 2020d. Creating applications for WinCC Unified Runtime Openness. PDF-dokumentti. Saatavissa: <https://support.industry.siemens.com/cs/document/109781803/> [viitattu 3.6.2022].

Kuva 73. Open Pipe -rajapinta ja tagin monitorointi Basic-tason komennolla.

Kuva 74. Open Pipe -rajapinta ja hälytyksien monitorointi Expert-tason komennolla.

Kuva 75. Raportointiesimerkki ODBC-rajapinnalle ja sen SQL-kysely. Siemens. 2021e. Configuring Logging for SIMATIC WinCC Unified Systems. PDF-dokumentti. Saatavissa: <https://support.industry.siemens.com/cs/document/109782859/> [viitattu 7.6.2022].

Kuva 76. Lisäosan Excel-työkalupalkki, tietolähteiden valinta ja tiedonsiirron suojauksen todennus.

Kuva 77. Aikasarjasegmentin määrittelyn vaiheet.

Kuva 78. Aikasarjasegmentti sijoittuneena raportointipohjalle.

Kuva 79. Segmenttien tietosisällön muokkaus konfiguraatioilla.

Kuva 80. Raportointipohjan koestaminen valmistelun yhteydessä.

Kuva 81. WinCC Unified -valvomosovelluksen Reports Control -objekti.

Kuva 82. Raportointiajon käynnistysehtoja Job parameters -välilehdellä.

Kuva 83. Raportointiajon yleiset asetukset Report jobs -välilehdellä.

Kuva 84. AlarmReport-raportointiajon tuottama raportti.