



Automaatiojärjestelmän asennuskannan seurannan hyödyntäminen

Oona Silvonen

OPINNÄYTETYÖ
Huhtikuu 2019

Sähkö- ja automaatiotekniikan koulutus
Automaatiotekniikka

TIIVISTELMÄ

Tampereen ammattikorkeakoulu
Sähkö- ja automaatiotekniikan koulutus
Automaatiotekniikka

SILVONEN, OONA:

Automaatiojärjestelmän asennuskannan seurannan hyödyntäminen

Opinnäytetyö 47 sivua, joista liitteitä 11 sivua
Huhtikuu 2019

Opinnäytetyö käsittelee Valmet Automationin asiakkaiden automaatiojärjestelmistä kerättävää tietoa ja sen hyödyntämistä. Automaatiojärjestelmiä integroivalla ja testaavalla osastolla oli tarve vähentää manuaalista tiedonhakua ja nopeuttaa näin järjestelmän testaamista. Tarkoituksena oli luoda järjestelmästä kerätyistä tiedoista yksi uusi raportti, joka korvasi kaikki käsin tehdyt testit. Tähän raporttiin oli tarkoitus sisällyttää myös jo käytössä olevista raporteista käytetyt tiedot. Uuden raportin kartoittamiseen käytettiin järjestelmätestausohjetta ja sitä verrattiin olemassa oleviin raporteihin. Tämän lisäksi apuna käytettiin työntekijöiden kokemuksia raportin tarpeista sekä haastateltiin tiedonkeruuohjelman kehittäjää tietojen saatavuudesta.

Työn tuloksena luotiin yksi uusi raportti, joka korvasi yhden manuaalisen testisarjan. Tämän lisäksi testausohjeesta löydettiin yksi testisarja, joka korvattiin jo olemassa olevalla raportilla. Uuteen raporttiin ei merkitty kaikkia testauksessa käytettyjä tietoja, sillä raporttiin olisi tullut liian monta tietosaraketta ja sen luettavuus olisi kärsinyt. Tämän vuoksi päädyttiin luomaan yksi uusi raportti ja lisäämään järjestelmätestausohjeeseen maininta uuden raportin käytöstä. Tämän lisäksi ohjeeseen lisättiin maininta olemassa olevasta raportista. Yhdessä näillä muutoksilla korvattiin kaksi manuaalista testisarjaa.

Työn aikana pääpaino työssä siirtyi raporttien tekemisestä niiden sisältöjen määrittelyyn. Kuitenkin työssä päästiin tavoitteisiin eli onnistuttiin vähentämään järjestelmätestauksen manuaalista testausta. Varsinaiset hyödyt työstä tulevat esiin vasta, kun raportti otetaan käyttöön osana testausta. Oletettavasti työllä saavutettiin säästöjä testauksen ajankäytössä.

Asiasanat: asennuskannan seuranta, raportointi, automaatiojärjestelmä

ABSTRACT

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

SILVONEN, OONA:
The Utilizing of Installed Base Managing of Automation Systems

Bachelor's thesis 47 pages, appendices 11 pages
April 2019

This thesis was made for Valmet Automation and it covers the collection of information from automation systems and the utilizing of said information. The objective of this study was to reduce the amount of manual testing and thus reduce the time used in system testing by creating a new report of the collected data. This study was carried out by comparing the system testing instruction and the existing reports and finding correlating data in them. Interviews of the employees who test the system and the creator of the data query were carried out as well.

As a result, one new report was made which replaced one set of manual tests. Because of the study another manual set of tests was also replaced by an existing report. These changes were added to the system testing instructions. Most of the manual testing was impossible to add to the report due to technical obstacles. Also, initially the objective was to create a new report which contained all the used data from the old and new reports, but this idea was discarded because the report would have become too big.

In conclusion, two manual sets of tests were eliminated from the system testing. The time savings caused by it will be seen after the report will be taken into use in the testing. However, it can be assumed that some time will be saved.

Key words: installed base managing, reporting, automation system

SISÄLLYS

1	JOHDANTO	7
2	VALMET AUTOMATION.....	8
3	AUTOMAATIOJÄRJESTELMÄ.....	10
	3.1 Automaatioprojekti	12
	3.2 Asennuskanta ja sen seuranta.....	14
4	AUTOMAATIOJÄRJESTELMIEN ASENNUSKANNAN SEURANTA..	15
	4.1 Automaatiojärjestelmän toimitusprojekti Valmetilla	15
	4.2 Asennuskannan seuranta Valmetilla	17
	4.2.1 Asennuskannan tiedonkeruuhjelma.....	19
	4.2.2 Asennuskannan tietokanta	20
5	RAPORTTITARPEIDEN KARTOITUS.....	22
	5.1 Järjestelmätestauksen tarpeet raportille.....	23
	5.2 Tarvittavien tietojen saatavuus.....	24
6	UUDEN RAPORTIN EHDOTUS	27
	6.1 Raportin luominen	29
7	POHDINTA	33
	LÄHTEET.....	35
	LIITTEET	37
	Liite 1. System and network testing - IBM.....	37
	Liite 2. IO Structure -raportti Report Builderissa	46
	Liite 3. IO Structure IBM-raportti	47

LYHENTEET JA TERMIT

ACN	Valmetin sovellus- ja ohjausasema, Application and Control Node
AMID	Asennuskanta kohtainen tunnuskoodi, Activity Management Identity code
AM-IS	Projektin asennuskannan testiajo, Activity Management Installation Survey
Asennuskanta	Yrityksen asiakkaille myymät ja käytössä olevat tuotteet
Auditointi	Selvitys, jolla tarkistetaan automaatiojärjestelmän tila
Automaatioprojekti	Projekti, joka tuottaa asiakkaalle automaatiojärjestelmän
Automaatioverkko	Ethernet-verkko, joka yhdistää DCS-järjestelmän asemat toisiinsa
DamaticXD(i)	Valmetin aikaisemman sukupolven järjestelmä
Dataluokka	Hakee, käsittelee ja säilöö tietokannan tietoja
DCS	Hajautettu automaatiojärjestelmä, Distributed Control System
Debugger	Valmetin järjestelmän ja sovellusten testaamiseen tarkoitettu työkalu
DNA	Valmetin automaatiojärjestelmä, Dynamic Network of Applications
Logistiikkaosasto	Valmetin osasto, joka suunnittelee, integroi, alustaa ja testaa automaatiojärjestelmän
Metso DNA noodi	Valmetin aikaisemman sukupolven järjestelmä DNA-järjestelmän prosessiasema
Ethernet	Lähiverkkotekniikka, joka lähettää tiedot paketteina
FAT	Tehdaskelpoisuustesti, Factory Acceptance Test
FBC	Kenttäväyläohjain, Field Bus Controller
IBC	I/O-signaalien ohjain, I/O Bus Controller
IBM	Asennuskannan tietokannan hallintatyökalu, Installed Base Manager
I/O	Tulo ja lähtö, input/output
I/O-kaappi	Kaappi, johon I/O-kortit on asennettu

I/O-kehikko	I/O-kaapissa oleva kehikko, johon kortit kiinnitetään
I/O-kortti	Fyysinen komponentti, joka yhdistää tulot ja lähdöt automaatiojärjestelmään
Järjestelmätestaus	Automaatiojärjestelmän laitteiston, ohjelmistojen ja lisenssien testaus
Ohjelmoitava logiikka	Tietokone, jota käytetään prosessin ohjaamiseen
PCS	Prosessiohjainpalvelin, Process Control Server
PIC	I/O-kehikon väyläosoite, Process Interface Controller
PI-kaavio	Putkitus- ja instrumentointikaavio
Proseduuri	SQL-tietokantaan luotu sarja toimintoja, joka korvaa yleisesti tehtyjä kyselyjä
Prosessiasema	Palvelin tai tietokone, johon automaatiosovellus on ladattu
QCS	Laatujärjestelmä, Quality Control System
Service	Valmetin huolto- ja asiakaspalveluosasto
SQL	Tietokantojen käsittelyä varten kehitetty kieli, Structured English Query Language
Teollinen internet	Teollisuuden älykkäitä laitteita yhdistävä verkko
Verkkolaite	Erilaiset tietoverkkoa jakavat, välittävät ja muuntavat laitteet.

1 JOHDANTO

Automaatioprojekti tuottaa asiakkaalle automaatiojärjestelmän, johon asennetaan erilaisia laitteita, ohjelmistoja ja lisenssejä. Järjestelmää ja kaikkia sen sisältämiä osia kutsutaan yhdessä asennuskannaksi. Automaatioprojektin aikana järjestelmätestauksen yhteydessä kerätään ensimmäinen otanta asennuskannan tilasta. Automaatioprojektin päätyttyä Valmet jatkaa asiakkaiden automaatiojärjestelmien asennuskannan seuranta huoltosopimuksen mukaisesti ja tarjoaa järjestelmän tilan perusteella kunnossapitoa. Valmet käyttää asennuskantojen tietoja ohjatakseen tutkimusta ja kehitystä, sekä suhteuttamaan varaosien tuotantoa. Asennuskannan seurantaan käytetään järjestelmässä ajettavaa testiajtoa, jonka tulosten pohjalta luodaan erilaisia raportteja edellä mainittuja käyttötarpeita varten.

Tarve tälle opinnäytetyölle ilmeni Valmetin järjestelmätestaus- sekä huolto-osaston tahoilta. Järjestelmätestauksessa käytettiin erilaisia työkaluja ja niiden lisäksi kahta raporttia asennuskannasta, mikä teki testauksesta työlästä. Tämän työn tavoitteena on luoda mahdollisimman kattava raportti järjestelmätestauksen käyttöön ja vähentää näin manuaalista testausta sekä käytettäviä työkaluja. Raporttiin oli tarkoitus sisällyttää kaikki testauksessa käytettävät asennuskannasta kerättävät tiedot. Lisäksi tavoitteena on selvittää Valmetin kunnossapitopalveluita tarjoavan osaston tarpeet uudelle raportille. Näiden tarpeiden pohjalta luotaisiin uudet raportit järjestelmätestausta ja kunnossapitoa varten.

Raporttien tekeminen vaatii ymmärrystä asennuskannasta, sen seurannasta sekä kerätyn tiedon käsittelystä. Näitä asioita käsitellään työn teoriaosuudessa ja niihin tutustumiseen varattiin aikaa viikko. Koko työn kestoksi arvioitiin kymmenen viikkoa ja tästä yli puolet varattiin raporttien luomiselle. Raporttitarpeiden kartoitukselle jäi aikataulun mukaan kaksi viikkoa, ja siihen suunniteltiin käytettäväksi olemassa olevia raportteja, testiajon tuloksia sekä tulevien raporttien käyttäjien kokemuksia ja tarpeita.

2 VALMET AUTOMATION

Työn tilaaja Valmet Automation Oy on osa Valmet-konsernia, jonka toimialoihin kuuluvat automaation lisäksi sellu, energia, erilaiset paperit ja näiden toimituksiin liittyvät palvelut. Yrityksellä on pitkä historia teollisuudessa, sen tuotantoon on mahtunut laivojen rakennusta, höyrykattiloiden valmistamista ja tykkien osien korjaamista. Yrityksen nimi Valmet tulee sanoista Valtion Metallitehtaat ja Valmet-nimi otettiin ensimmäisen kerran käyttöön vuonna 1951. Yrityksen nimi vaihtui välillä Metsoksi, mutta palautettiin entiselleen vuonna 2013, kun Metso ja Valmet jakautuivat omiksi pörssiyhtiöikseen.

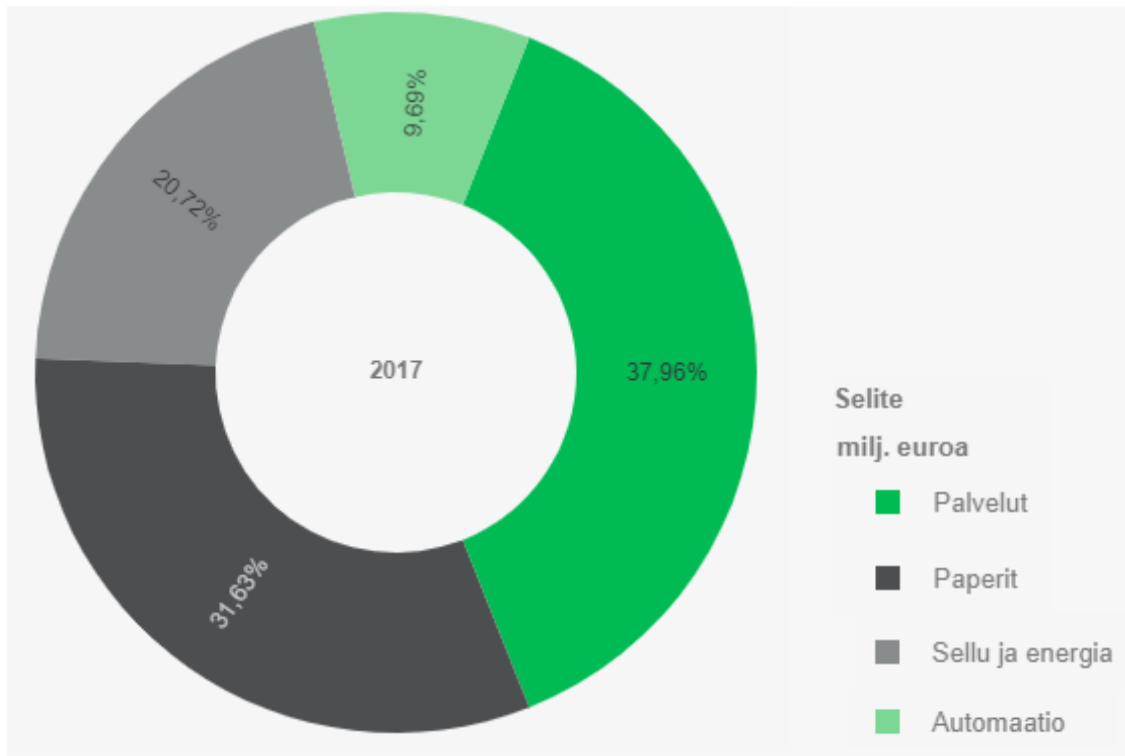
(220 vuotta teollista historiaa 2019.)

Vuonna 2019 Valmet työllistää yli 12 000 ihmistä ympäri maailmaa. Yrityksen toiminta keskittyy sellutehtaiden ja erilaisten paperivalmistelinjojen tuottamiseen ja ylläpitoon. Toiminta on asiakaspalvelulähtöistä ja ekologisesti kestävää. Yrityksen päätoimipaikka on Espoossa ja Valmet Automationin toimipaikka sijaitsee Tampereella. Valmetin toimitusjohtaja on Pasi Laine ja Automaatio-liiketoimintalinjan johtajana toimii Sami Riekkola. (Valmet yrityksenä 2019.)

Valmet Automationin tuotteisiin kuuluvat hajautetut automaatiojärjestelmät (DCS), laatuja järjestelmät (QCS), analysointirit, mittaukset, suoritusratkaisut sekä turvallisuuteen ja teolliseen internetiin liittyvät ratkaisut. Vuonna 2015 automaatiojärjestelmiä oli toimitettu yli 4500 (Laine 2015, 23) eri aloille, joihin kuuluvat energiatuotanto, risteilylaivat, prosessiteollisuus, sellu, paperi ja jäteveden puhdistus. Järjestelmiä asennetaan uusiin tehtaisiin, päivitetään vanhojen Valmetin järjestelmien tilalle sekä niillä korvataan kilpailevien yritysten järjestelmiä. Valmet Automationin tuotteet tukevat muiden Valmet-konsernin toimialojen myyntiä. (Valmet presentation 2017, 15.)

Vuonna 2017 Valmetin liikevaihto oli 3058 miljoonaa euroa. Varsinaisen automaation osuus oli alle kymmenyksen koko liikevaihdosta (kuvio1). Automaatio-toimitusprojekti on kuitenkin vain hyvin pieni osa automaation elinkaaresta. Toimituksen jälkeen projekti siirtyy huollon ja asiakaspalvelun alaiseksi ja sitä kautta automaatiojärjestelmien tuottavuus yritykselle nousee. Vuonna 2017 palveluiden liikevaihto oli 1178 miljoonaa euroa eli lähes 38% koko yrityksen liikevaihdosta.

Kuviossa 1 näkyy Valmetin liikevaihdon jakautuminen toimialoittain. (Taloudellista tietoa 2019.)



KUVIO 1. Valmetin toimialojen liikevaihdot prosentteina vuonna 2017 (Taloudellista tietoa 2019)

3 AUTOMAATIOJÄRJESTELMÄ

Automaatiojärjestelmä voi olla yksittäinen ohjelmoitava logiikka tai hajautettu automaatiojärjestelmä, jossa eri toiminnot on jaettu omille tietokoneilleen eli asemille. Tässä työssä keskitytään hajautettuihin automaatiojärjestelmiin eli DCS-järjestelmiin. Omille asemille voidaan jakaa automaatio-sovellusten suunnittelu, historian keruu, laitteiden ohjaimet ja prosessin valvominen. Eri asemat yhdistyvät toisiinsa Ethernet-verkon avulla ja muodostavat näin automaatioverkon. Fyysinen automaatiojärjestelmä koostuu antureista, toimilaitteista, ohjaimista, käyttöliittymälaitteista ja tiedonsiirtolaitteista. Automaatiojärjestelmän ohjelmallinen puoli taas koostuu automaatio-sovelluksista, jotka ohjaavat laitteita, säätävät prosessia ja lukevat mittauksia. (Strömman, Hirvonen, Hukki & Tommila 2017, 10-12.)

Mittaustiedot tuodaan järjestelmään tulokorttien kautta ja laitteiden ohjaustiedot viedään järjestelmästä laitteille lähtökorttien avulla. Yhdessä näitä kutsutaan I/O-korteiksi (input/output). I/O-kortit sijaitsevat I/O-kaapissa, joka hajautetussa automaatiojärjestelmässä on viety lähelle prosessia, jotta kaapelivedot kaapilta toimilaitteelle olisivat mahdollisimman lyhyitä. I/O-kortit on asennettu kaappiin kerroksittain ja yhtä tällaista kerrosta kutsutaan kehikoksi. Kehikolle on annettu PIC-numero, jolla yksilöidään kehikot. Yhteen kehikkoon mahtuu 16 I/O-korttia ja jokaisen korttirivin edessä on I/O-signaaleja ohjaava prosessiväyläohjain eli IBC-kortti. Termit PIC ja IBC sekoittuvat toisinaan, sillä niitä molempia käytetään erottamaan kehikot toisistaan. Erona on se, että PIC on ohjelmallinen muuttuja, kun taas IBC viittaa fyysiselle kortille annettuun numeroon. (Oulun ammattikorkeakoulu n.d.)

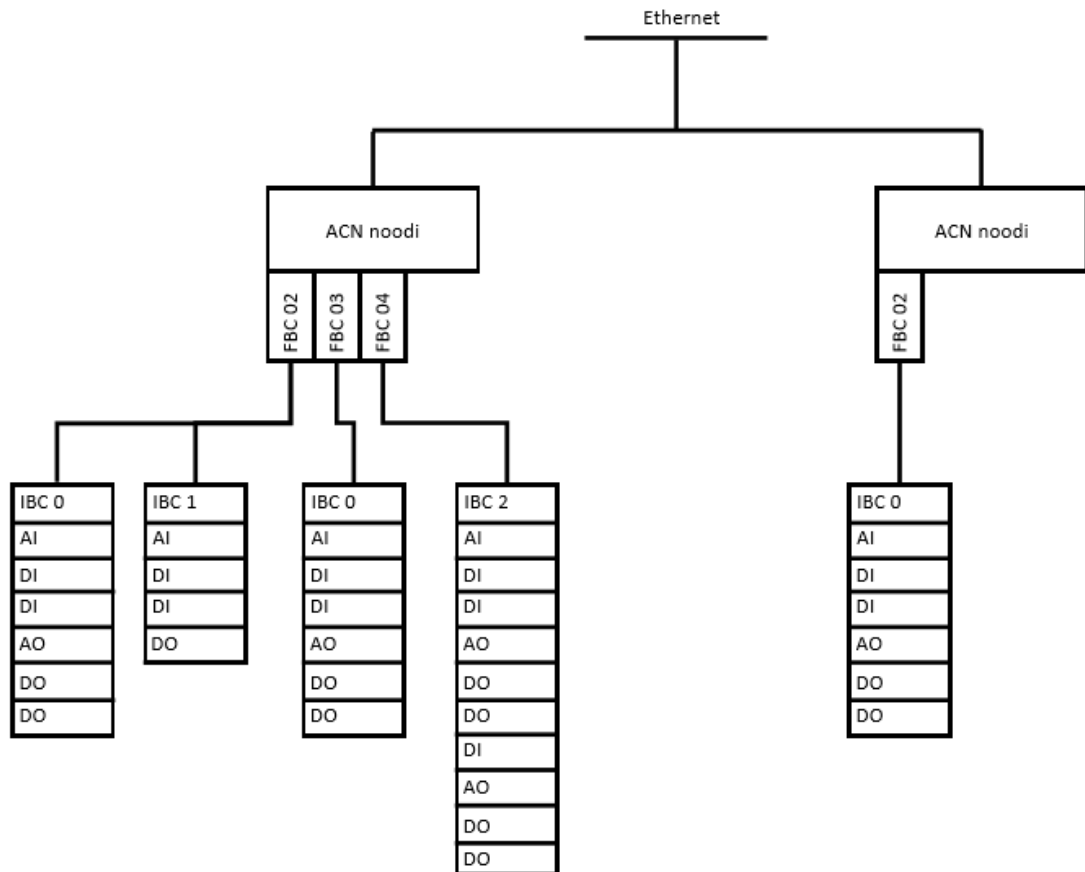
Kuvassa 1 näkyy, miten eri kortit on asennettu kaappiin. Rivin alussa vasemmalla on prosessiasema (ACN SR1), jossa automaatio-ohjelma pyörii. Valmetilla prosessiasemia nimitetään ACN:ksi eli sovellus- ja ohjaussolmuksi eli noodeiksi. Sen vieressä oikealla on virtalähde, joka antaa käyttöjännitteen kehikon korteille. Seuraavana rivissä on IBC-kortti ja sitä seuraavat kahdeksan korttia ovat I/O-kortteja eli tulo- ja lähtökortteja. Kehikkoon mahtuu tyypillisesti 16 I/O-korttia. Yh-

dessä kortissa on enintään 8 kanavaa, joihin voidaan jokaiseen liittää yksi signaali kentältä. Kuvassa 1 näkyy I/O-korttien alapuolella liitynnät, joista viedään johdotukset kentällä oleville laitteille. (Heikkinen 2019.)



KUVA 1. Automaatiojärjestelmän I/O-kaapin kehikko (Control point n.d.)

I/O-kaapissa oleva ACN-noodi on liitetty automaatiojärjestelmään kenttäväylällä. Suurissa prosesseissa voi olla tarve useammalle prosessiasemalle, jolloin järjestelmä voidaan jakaa osajärjestelmiin. Tämän lisäksi yhdellä prosessiasemalla voi olla useampi FBC eli kenttäväyläohjain, jonka perässä taas voi olla enintään 16 IBC:tä. FBC ohjaa tiedon liikennöintiä väylässä. Tällaisessa suuressa järjestelmässä ACN-noodi ei ole kuvan 1 tapaan kehikossa, sillä suuremmille järjestelmille tarkoitetut prosessiasemat ovat kooltaan huomattavasti suurempia. Kuviossa 2 on havainnollistettu I/O-rakennetta, kun järjestelmä on jaettu useammalle ACN-noodille. (Heikkinen 2019.)



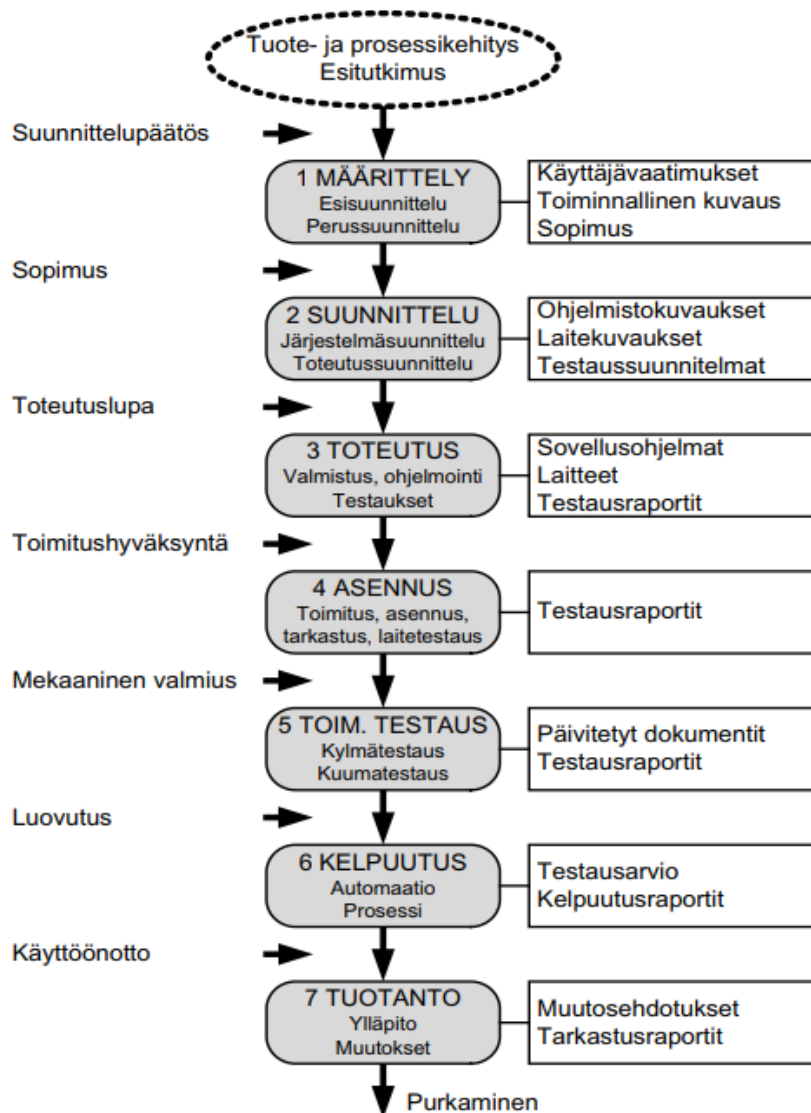
KUVIO 2. Osajärjestelmiin jaetun järjestelmän I/O-rakenne

Kuviossa 2 automaatiojärjestelmä on jaettu kahdelle prosessiasemalle, joista toisella on käytössä 3 FBC:tä ja toisella yksi. Vasemman noodin FBC 02:ssa on kaksi IBC:tä, FBC 03:ssa ja FBC 02:ssa on yhdet IBC:t. ACN-noodit ovat liittyneinä järjestelmän automaatioverkkoon Ethernetilla. IBC:iden perässä näkyvät asennetut I/O-kortit. Korttien kirjainyhdistelmät kertovat onko kyseessä analoginen (A) vai digitaalinen (D) tieto ja onko kortti tulo (I) vai lähtö (O).

3.1 Automaatioprojekti

Automaatioprojekti määritellään seuraavasti: ”Automaatioprojektin tarkoitus on tuottaa tehtaan automaatiojärjestelmän toteuttamiseen, käyttämiseen ja ylläpitoon tarvittavat tiedot sekä toteuttaa itse järjestelmä.” (Strömman, Hirvonen, Hukki & Tommila 2017, 8) Eli projektin tarkoituksena on tuottaa automaatiojärjestelmä ja kaikki tiedot sen elinkaaresta huolehtimiseen. Automaatioprojekti päättyy, kun vastuu järjestelmästä siirtyy projektilta asiakkaalle.

Automaatiojärjestelmän elinkaari alkaa, kun syntyy suunnittelupäätös eli yritys päättää haluavansa toteuttaa tuotantonsa automaatiolla. Elinkaari loppuu järjestelmän tai koko laitoksen purkamiseen. Automaatiojärjestelmän elinkaari on jaettu seitsemään vaiheeseen (kuva 2), joista ensimmäinen on järjestelmän tarpeiden ja toiminnan määrittely. Määrittelyjen pohjalta luodaan sopimus siitä, mitä järjestelmän tekijä ja asiakas lupautuvat toimittamaan toisilleen projektin aikana. Sopimus pitää sisällään muun muassa järjestelmän toimintakuvauksen, PI-kaaviot ja projektin budjetin. (Strömman, Hirvonen, Hukki & Tommila 2017, 8.)



KUVA 2. Automaatiojärjestelmän elinkaari (Strömman, Hirvonen, Hukki & Tommila 2007, 16)

Sopimuksen allekirjoittamisen jälkeen voidaan siirtyä suunnitteluun ja siitä itse järjestelmän toteutukseen, johon kuuluu sekä fyysisen laitteiston kokoaminen ja

alustaminen että automaation eli sovellusten luominen. Kun sovellusohjelmat on saatu valmiiksi ja laitteisto on koottu, siirrytään järjestelmätestaukseen (Factory Acceptance Test, FAT). Hyväksytyn järjestelmätestauksen jälkeen järjestelmä toimitetaan asiakkaan tehtaalle ja asennetaan. Työmaalla suoritetaan uudelleen järjestelmätestaus (Site Acceptance Test, SAT), ja kun kaikki on kunnossa, siirrytään tuotantoon. Tuotannon pituutta ja tehokkuutta lisätään huolloilla, kunnes on aika purkaa järjestelmä. (Strömman, Hirvonen, Hukki & Tommila 2017, 8.)

3.2 Asennuskanta ja sen seuranta

Asennuskannalla tarkoitetaan yrityksen myymiä asiakkaille käytössä olevia tuotteita. Näiden tuotteiden seuranta alkoi yleistyä yritysten välisen kilpailun seurauksena. Erottuakseen joukosta yritysten tuli tarjota kokonaisvaltaisempaa palvelua asiakkailleen ja tämä edellytti kattavaa tietoa asiakkaille käytössä olevista tuotteista. Myydyistä tuotteista alettiin pitää kirjaa ja tätä kutsutaan asennuskannan seurannaksi. (Saccani, Alghisi, & Borgman 2012, 415-417.)

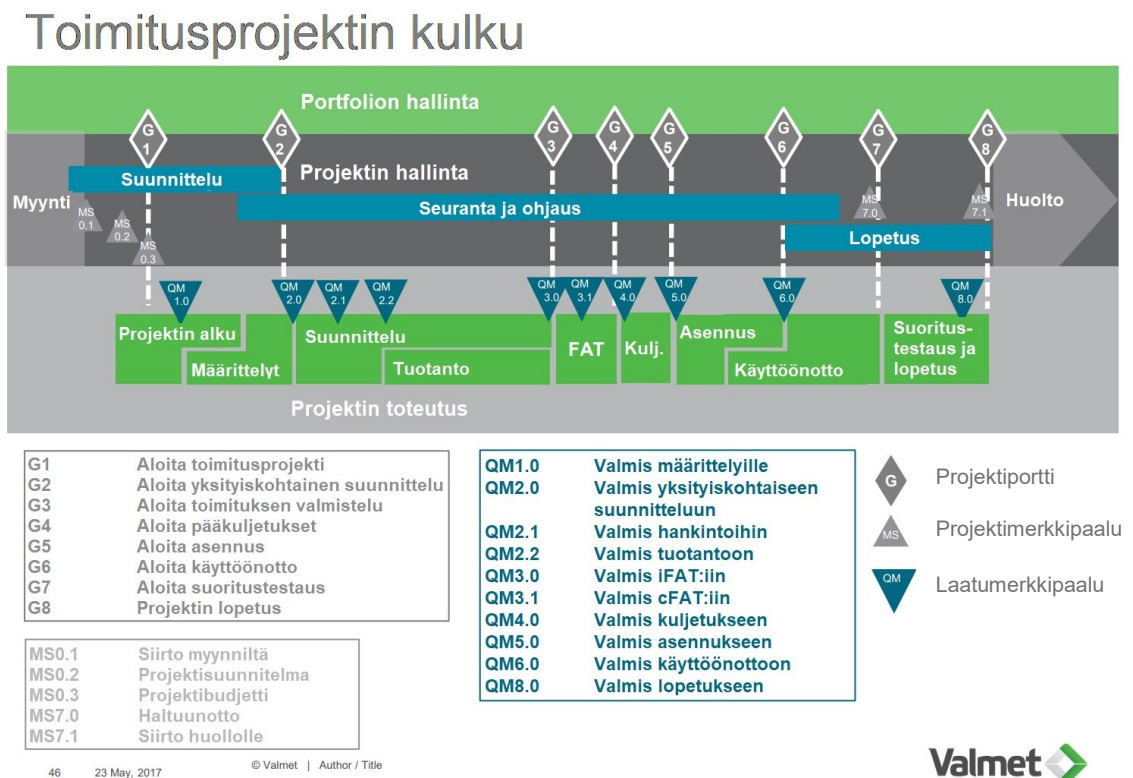
Asennuskannalla tarkoitetaan tässä asiayhteydessä myytyyn automaatiojärjestelmään asennettuja laitteita, ohjelmistoja ja lisenssejä. Asennuskannan seuranta on osa automaatiojärjestelmään kohdistuvaa tiedonkeruuta, toinen osa on tuotannon seuranta. Asennuskannan seuranta on yrityksillä melko laajasti käytössä ja kehityksen alla. Kuitenkin se on tuotannon seurantaan harvinaisempaa ja kirjallisuudessa sitä harvoin käsitellään itsenäisenä ilmiönä. Seurantojen yleisyyden ero näkyy myös niihin kohdistuvissa standardeissa; tuotannon seurantaan on luotu standardi, asennuskannan seurantaan ei. (Ala-Ruisku 2009, 11-19.)

Asennuskannan seurantajärjestelmien tarkoituksena on jäljittää myytyjen tuotteiden sijainti, omistaja ja käyttäjä, käyttötarkoitus, missä olosuhteissa tuotteita on sovellettu, elinkaaristatus, mitä huoltotoimenpiteitä ja teknisiä muutoksia on tehty, mitkä osat on huollettu tai vaihdettu sekä tuotteiden nykyinen tekninen tila (Ala-Ruisku 2009, 5-6). Kerättyä tietoa käytetään hyväksi tuotteiden elinkaaren ajan kunnonvalvonnassa, asiakaspalvelussa ja huollossa.

4 AUTOMAATIOJÄRJESTELMIEN ASENNUSKANNAN SEURANTA

4.1 Automaatiojärjestelmän toimitusprojekti Valmetilla

Automaatiojärjestelmän toimitusprojektin malli (kuva 3) on yksityiskohtainen kuvaus Valmetin automaatioprojekteista (Tyynelä 2017, 46, muokattu). Mallissa on yleisen automaatioprojektin vaiheet sekä niiden lisäksi projektiportteja, projekti-merkkipaaluja ja laatumerkkipaaluja. Projektia seurataan myös eri näkökulmista: portfolion hallinta, projektin hallinta ja projektin toteutus. Näiden toimintamallien avulla saavutetaan automaatioprojektien kesken tasainen laatu.



KUVA 3. Valmetin automaatiojärjestelmän toimitusprojektin kulku (Tyynelä 2017, 46, muokattu)

Projektiportit ovat portfolion ja projektin hallinnan työkaluja. Portfolion hallinnalla tarkoitetaan kaikkien Valmetin automaatioprojektien hallintaa kokonaisuutena. Projektiportit ovat virallisia projektikatselmuksia, joissa tarkastellaan projektin tilannetta ja suunnitelmissa pysymistä (aikataulu, budjetti ja resurssit). Jos porttien

kohdalla huomataan suuria muutostarpeita, voidaan vaatia projektisuunnitelman päivitystä. (Tyynelä 2019.)

Projektimerkkipaalut ovat puhtaasti projektihallinnan työkaluja, joilla seurataan edistymistä. Niiden avulla projektia tarkastellaan, kun mukana on projektin työryhmän lisäksi muita tahoja: myyntiosasto, huolto-osasto tai asiakas. Laatumerkkipaalujen avulla seurataan sisäisesti toteutuksen edistymistä keskittyen työryhmän edistymiseen. Laatumerkkipaalut ovat erityisen tärkeitä projektin laadun kannalta, sillä niiden avulla varmistetaan, etteivät työtehtävät mene päällekkäin silloin kun se olisi haitallista projektin toteutukselle. Esimerkiksi järjestelmän testausta ei voida aloittaa ennen kuin tuotanto on lopetettu ja järjestelmä on valmis. (Tyynelä 2019.)

Yleensä automaatioprojektin työryhmä huolehtii kaikesta järjestelmään liittyvästä, mutta Valmetilla on oma osasto, joka suunnittelee ja tilaa laitteiston, josta automaatiojärjestelmä rakentuu. Tämä osasto on System and Networks tai logistiikkaosasto, joka toimii osana automaatioprojektia, mutta on oma yksikkönsä. Logistiikkaosasto tekee tilaukset järjestelmän osille, kokoaa ja alustaa järjestelmän, eli asentaa siihen kaikki tarvittavat käyttöjärjestelmät ja ohjelmistot sekä konfiguroi palvelimet. Palvelimella tarkoitetaan tässä yhteydessä kaikkia tietokoneita, jotka liittyvät Valmet DNA-järjestelmään. Logistiikkaosasto testaa valmiin järjestelmän ennen kuin se siirtyy automaatioprojektin haltuun. (Jokinen 2019a.)

Valmetin mallin mukaan automaatioprojekti päättyy, kun korjausvastuu järjestelmästä siirtyy Servicelle eli Valmetin huolto-organisaatiolle ja automaatioprojektin työntekijät siirtyvät uusien projektien pariin. Huolto-organisaatio ylläpitää järjestelmää palvelusopimuksen ajan ja tarjoaa automaatiojärjestelmien asiakkaille auditointeja järjestelmän kunnosta, joiden avulla voidaan ennakoida huoltotarpeita sekä ohjelmistojen ja laitteiden vanhentumisia. (Jokinen 2019a.)

Automaatioprojektin jokainen vaihe on oleellinen projektille, mutta tämän työn kannalta tärkein osuus on järjestelmätestaus, sekä toimitusprojektia seuraava palvelusopimus. Tämä työ käsittelee asennuskannan seuranta, joka alkaa, kun laitteisto on fyysisesti koottu ja logistiikkaosasto alkaa testaamaan sitä ja jatkuu järjestelmän elinkaaren loppuun saakka. Järjestelmätestauksen tulee olla valmis

ennen FAT:ia, eli laatumerkkipaalu QM3.0:aa. Näin saadaan testauksen yhteydessä ensimmäinen otanta asennuskannasta, jota voidaan käyttää myöhemmin vertailukohtana vikojen ilmaantuessa ja nopeuttaa näin ongelmanratkaisua.

4.2 Asennuskannan seuranta Valmetilla

Valmet seuraa automaatioprojektinsa laitteiston ja laitteistoon asennettujen ohjelmien tilaa Valmet DNA-järjestelmän sisältämien sensorien avulla. Sensorit ovat järjestelmän koodissa määritellyjä ohjelmallisia pisteitä, joiden kautta järjestelmästä saadaan Valmet DNA Activity Management Installation Surveyllä eli AM-IS-testiajolla otettua kuvakaappaus-tyylinen tilannetietoraportti. Käsittelemättöminä testiajon tulokset ovat lähes käyttökelvottomia muille kuin järjestelmän asiantuntijoille. Tästä syystä AM-IS-ajon parametreista on luotu kohdennettuja raportteja eri käyttäjien tarpeita vastaamaan. Nämä raporttipohjat ovat tallennettuina IBM, eli Installed Base Manager -tietokantaan, johon tallennetaan myös AM-IS-testiajojen tulokset. (Jokinen 2019b.)

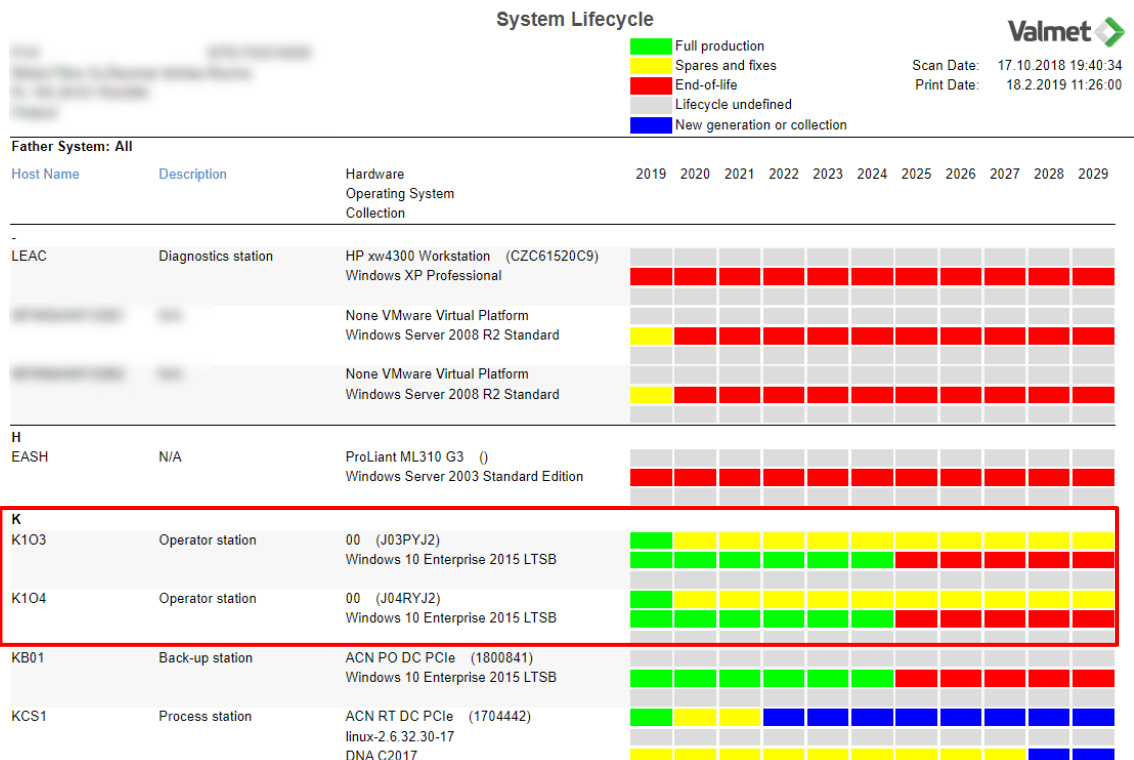
AM-IS-ajon pohjalta tehtyjä raportteja käyttävät Valmetin huolto- ja logistiikkaosastot. Logistiikkaosasto käyttää raportteja muun testauksen tukena testatesaansa fyysistä järjestelmää, kun se on alustettu. Huolto-osasto käyttää raportteja järjestelmän kunnon seurantaan ja tarjoaa asiakkaille raporttien perusteella kunnossapitoa ja päivityksiä. Raportit ovat myös automaatiojärjestelmien asiakkaiden nähtävissä huolto-osaston kautta. Raporttien avulla asiakkaiden järjestelmät pystytään pitämään ajan tasalla kuten varmistamaan, ettei käytössä olevien käyttöjärjestelmien tuki lopu yllättäen. Valmet käyttää asennuskannan tietoja hyväkseen myös varaosien tuotannon sekä tutkimuksen ja kehityksen ohjaamisessa. Kun on tiedossa mitä laitteistoa asiakkailta on käytössä, voidaan varaosien valmistus suhteuttaa tarpeisiin ja kehittää niitä laitteita ja ohjelmia, joita on eniten käytössä. (Kleemola 2016.)

Automaatioprojektia seuraavaan palvelusopimukseen kuuluu huolto-osaston tarjoamat järjestelmäauditoinnit ja niiden osana asennuskannan testiajot. Auditoinneilla tarkistetaan käyttöjärjestelmien tuen jatkuminen, virustorjuntajärjestelmien tila, kriittiset turvallisuusmuutostiedostot, lokikansiot, käyttöjärjestelmäpäivitykset

tietokoneisiin, palvelimiin ja kytkimiin sekä kytkinten uusintatarpeet. Auditointien pohjalta järjestelmän eri osille voidaan luoda elinkaarisuunnitelma. Se kertoo, kauanko järjestelmän osa on tuotannossa, milloin valmistetaan vain varaosia ja milloin se tulisi uusia, mikäli toimintaa halutaan jatkaa. (Tyynelä 2017, 52.)

Kuva 4 on kaappaus erään asiakkaan System Lifecycle -raportista, josta on sensuroitu asiakkaaseen viittaavat tiedot. Kuvaa on rajattu selkeyden vuoksi. Raportin sarakkeissa vasemmalta oikealle lukee noodin nimi, kuvaus ja laite sekä laitteen alla käyttöjärjestelmä ja kokoelma, jossa laite on julkaistu. Oikeassa laidassa on kuvattu eri järjestelmän osien elinkaaristatusta väreillä:

- vihreä = täysi tuotanto
- keltainen = varaosia ja korjauksia
- punainen = elinkaaren loppu
- harmaa = elinkaari määrittelemättä
- sininen = uusi sukupolvi tai kokoelma.



KUVA 4. Valmetin asiakkaan järjestelmän System Lifecycle -raportti, rajattu, osa tiedoista sensuroitu asiakkaan suojelemiseksi

Elinkaariraportista nähdään, että käytössä olevien operointiasemien tuotanto lopetetaan vuoden 2019 jälkeen, mutta varaosia on saatavilla ainakin vuoteen

2029 asti. Operointiasemien käyttöjärjestelmien tuki taas loppuu vuonna 2024, mikä tarkoittaa, että yrityksen on päivitettävä asemien käyttöjärjestelmät ennen sitä. Asennuskannan seurannan tuloksena saatu System Lifecycle -raportti siis auttaa Valmetin huolto-osastoa ja asiakasta ennakoimaan huolto- ja päivitystarpeita.

4.2.1 Asennuskannan tiedonkeruuohjelma

AM-IS on Valmetin kehittämä ohjelma, joka kerää dataa asiakkaiden järjestelmiin asennetusta laitteistosta. Ensimmäinen versio AM-IS:sta julkaistiin vuonna 2006 ja vuonna 2007 ohjelma asennettiin ensimmäistä kertaa onnistuneesti asiakkaan tehtaalle. AM-IS-ohjelmasta on tehty useita versioita, ja on tärkeää käyttää aina viimeisintä versiota, jotta kaikki testiajon pohjalta luodut raportit toimivat. (Lahtinen 2014, 2-3; Kleemoja 2016, 6.)

AM-IS asennetaan tyypillisesti suunnitteluasemalle ja ajetaan ensimmäisen kerran ennen FAT:ia. Ohjelma kaappaa tilannetietoja asennetusta kannasta sekä muuta huoltojärjestelmä dataa Valmet DNA-, Metso DNA- ja DamaticXD(i)-automaatiojärjestelmistä. Tiedonkeruu suoritetaan kahdella toisistaan riippumattomalla tavalla:

1. DNA-noodien julkaisemaa dataa kerätään diagnostiikka antureista. Tämä vaatii DNA-kommunikointiyhteyden ja Valmet DNA-tietoverkon.
2. PC-datakysely kerää tietoa raudasta, käyttöjärjestelmistä ja ei-DNA-ohjelmistoista. (Kiviniemi 2016, 3.)

Yhtä testiajon tuloksena saatua tiedostoa kutsutaan datasetiksi. Tiedosto tallennetaan .xml muotoon ja lähetetään Valmetille Tampereelle siirrettäväksi asennuskannan tietokantaan.

4.2.2 Asennuskannan tietokanta

IBM on selainpohjainen Valmetin luoma ja ylläpitämä asennuskannan seurannan tietokantajärjestelmä, johon tallennetaan AM-IS-testiajon tuloksena saadut tiedostot. Asennuskannan seuranta on tehty Valmetilla jo ennen IBM:ää, tällöin käytössä oli Oraclen tietokantaa käyttävä SoftLife-järjestelmä. SoftLifen kehitys alkoi vuonna 2005, ja se otettiin käyttöön vuonna 2007. Järjestelmä koettiin kankeaksi, mikä johti tietokannan uusimiseen vuonna 2014. Tilalle tuli IBM, joka käyttää ketterämpää Microsoftin tietokantaa. Kaikki SoftLifeen tallennetut tiedostot siirrettiin IBM:ään järjestelmän vaihdon yhteydessä. (Kleemola 2016, 4-5.)

IBM:stä voi etsiä projekteja asiakkaan tehdaskohtaisella koodilla, AMID:llä (asennuskantakohtainen tunnus), asiakkaan nimellä tai tehtaan sijainnilla. IBM itsessään ei luo raportteja, vaan toimii käyttöliittymänä tietokannan ja tiedonhakijan välillä. Raportit hakevat tietoa Tampereella sijaitsevasta SQL Serverin tietokannasta, ja palauttavat ne näytölle tulosteena. (Kleemola 2016, 5, 8.)

SQL Server on Microsoftin tietokantojen hallintajärjestelmä, johon voidaan tallettaa tietoja taulujen muodossa ja suorittaa tallennettuun dataan kohdistuvia kyselyjä. Kyselyt luodaan SQL-kyselykielellä (Structured English Query Language), joka on tietokantojen käsittelyä varten kehitetty kieli. Usein käytettyjä kyselyjä voidaan myös tallentaa proseduureina, jotka toimivat ikään kuin raporttipohjina. (McGehee, Kraft & Shepker 2000.)

Nämä proseduurit liitetään dataluokkiin, joita IBM:n raportit lukevat. Proseduurien ja dataluokkien liittäminen tapahtuu Visual Studioissa C#-ohjelmointikielellä. Kun proseduuuri ja dataluokka on liitetty, voidaan tietokannan dataa käyttää raportissa. Tietokantaan voidaan tallentaa jokaista asennuskantaa eli AMID:tä kohden enintään 15 tiedostoa kerralla. Tämän jälkeen tallennetut uudet tiedostot poistavat samalla vanhimman. Poikkeuksena on FAT:sta saatu aivan ensimmäinen tiedosto, joka ei poistu kannasta sillä sitä käytetään vertailukohtana vikaantuneen järjestelmän tilalle. (Elfving 2019.)

Olemassa olevat raportit

IBM:ssä on saatavilla asiakkaiden järjestelmäkohtaisia raportteja sekä globaaleja eli koko Valmetin asennuskantaa käsitteleviä raportteja. Globaalit raportit antavat suuntaa tuotekehitykselle ja tutkimukselle sekä varaosien tuotannolle. Järjestelmäkohtaiset raportit toimivat huolto-organisaation työkaluina ja logistiikkaosasto käyttää osaa raporteista järjestelmätestauksen tukena. Tämän työn tuloksena tehtävä raportti ovat järjestelmäkohtainen raportti.

5 RAPORTTITARPEIDEN KARTOITUS

Työn alkumäärittelyjen mukaan raportit tehtäisiin logistiikka- ja huolto-osaston tarpeisiin. Logistiikkaosastoa varten oltiin määritelty järjestelmätestausta tukeva raportti ja huolto-osastolle esiin nousevien tarpeiden mukainen raportti. Työn alettua kävi ilmi, että molemmilla tahoilla saattaisi olla tarvetta useammalle raportille, jolloin työtä rajattiin yhteen raporttiin osastoa kohden.

Työ aloitettiin tutustumalla asennuskannan seurantaan käytettäviin työkaluihin ja lähettämällä sähköpostitse kyselyjä huolto- ja logistiikkaosastolle raporttitarpeista. Järjestelmiin tutustumiselle oltiin varattu aikaa noin viikko, mutta todellisuudessa järjestelmien toiminnan opettelu jatkui koko työn ajan. Huolto-osaston raporttitarpeiden kartoitusta siirrettiin myöhemmäksi työntekijöiden lomien takia. Oikeiden ihmisten tavoittamisessa oli myös haasteita huoltotoimipisteiden hajautuneen rakenteen vuoksi. Logistiikkaosaston raportin määrittely voitiin aloittaa, kun testausohje saatiin käyttöön. Aikataulun lykkääntyminen osoittautui kuitenkin hyväksi asiaksi, sillä järjestelmään tutustuminen vei odotettua kauemmin ja näin aikaa jäi myös raporttien tekemiseen käytettäviin ohjelmiin tutustumiselle.

Ennen työn alkua oli määritelty, että työhön kuuluu järjestelmiin tutustuminen, raporttitarpeiden kartoitus sekä raporttien luominen. Työn edettyä kartoitusvaiheeseen oli aikataulusta kulunut noin puolet, mikä johti työn uudelleenrajaamiseen. Pääpaino työssä oli koko ajan ollut raporttiin haluttujen tietojen määrittelyssä. Tämä ei kuitenkaan näkynyt alkuperäisessä aikataulussa, jossa raporttien luomiseksi oltiin varattu yli puolet työn ajasta. Työn uudelleenrajauksessa sovittiin, että itse raporttien luominen hoituisi pitkälti Valmetin raporteja käsittelevän henkilöstön puolesta ja loput jäljellä olevasta työajasta käytettäisiin raporttien määrittelemiseen.

Raporttitarpeiden kartoitus alkoi, kun logistiikkaosaston järjestelmätestausohje saatiin käyttöön. Ohjeesta kerättiin erilliseen dokumenttiin kaikki Valmet DNA Debuggerilla tehtävät testit sekä logistiikkaosaston esimiehen ilmaisemat muut manuaalisesti tehtävät testit, jotka toivottiin korvattavan raportilla. Debugger on Valmetin työkalu, jolla testataan DNA-järjestelmän sovellusten ja itse järjestelmän

toimintaa. Työkalulla voidaan tutkia ja muuttaa Valmetin DNA-järjestelmän sisäisiä tietoja. Tavoitteena oli korvata mahdollisimman moni Debuggerilla tehtävä testi AM-IS-ajosta saatavalla tiedolla IBM-raportissa.

Järjestelmätestauksenohjetta oli tarkoitus verrata AM-IS-kyselyn tuloksiin. Pian kävi kuitenkin ilmi, ettei tämä ole mahdollista, sillä käsittelemättömistä AM-IS-tulosteista oli mahdotonta paikallistaa tarvittavaa tietoa. Tämän takia testaus suunnitelmaa alettiin verrata olemassa oleviin raportteihin. Sopivat raportit päätettiin raporttien nimien perusteella ja sen jälkeen tutkimalla tarkemmin raporttien sisältöä. Osaa halutuista tiedoista ei löytynyt suoraan olemassa olevista raporteista. Tällöin siirryttiin haastattelemaan AM-IS-testiajon tekijää raporttiin haluttujen tietojen saatavuudesta.

Huolto-osaston raporttia varten ei työn alussa oltu määritelty selkeää tarvetta. Eri huolto-osastojen toimipisteille lähetettiin kyselyjä mahdollisista tarpeista, mutta niitä ei ilmennyt. Myöhemmin työn loppuvaiheilla huolto-osastolta ilmaistiin muutostarpeita yhteen olemassa olevaan raporttiin. Aikataulullisista syistä tätä ei kuitenkaan enää otettu osaksi opinnäytetyötä.

5.1 Järjestelmätestauksen tarpeet raportille

Järjestelmätestauksessa käytettiin useita eri työkaluja, joiden määrää haluttiin vähentää testauksen selkeyttämiseksi. Lisäksi valmiin raportin käyttäminen testauksessa vähentäisi manuaaliseen tiedonhakuun kuluvaan aikaa. Automaatiojärjestelmän toimitusprojektissa logistiikkaosaston järjestelmätestaukselle on varattu yleensä yhdestä kahteen päivää, joten kaikki säästetty aika on hyväksi.

Järjestelmätestauksessa testattiin seuraavat asiat järjestelmästä Debuggerilla:

- I/O-korttien rakenne
- FBC:n eli kenttäväyläohjainten vikalaskurit
- multicast eli ryhmälähetyskommunikaatio
- virta- ja laiteviat
- PCS:n eli prosessiohjainpalvelimen varmuuskopiointi
- ajan synkronointi.

Edellä mainittujen lisäksi toiveena oli saada tietoverkossa olevien verkkolaitteiden vikalaskurit osaksi raporttia. Lisäksi testaussuunnitelmassa pyydetään tarkistamaan tietoja Versio ja Asiakaslisenssit IBM-raporteista. Näiden tarpeiden pohjalta lähdettiin selvittämään mitä tietoja olisi mahdollista saada AM-IS-ajon kautta IBM-raporttiin.

Liitteenä 1 olevaan System and Networks testing – IBM -dokumenttiin kerättiin kaikki IBM-raporteilla korvattavat testit sekä kirjattiin raportti, josta testattavat tiedot löytyvät, mikäli sellainen oli olemassa. Taulukkoon on merkattu raportin tietojen tarpeellisuus logistiikkaosaston testaukselle. I/O-korttien ja ryhmälähetyksen testauksesta tulostettiin testiohjeen mukaiset tiedot Debuggerilla Valmetin testijärjestelmästä. Debuggerin tulosteita verrattiin olemassa oleviin raporteihin. Liitteestä poistettiin Debuggerilla tulostetut testit tietosuojan säilyttämiseksi.

5.2 Tarvittavien tietojen saatavuus

Versiotestaus

Järjestelmätestauksessa tarkistetaan, että asennettujen ohjelmistojen versiot ovat automaatioprojektilta saadun määrittelyn mukaiset. Tähän on käytetty IBM:n Versio-raporttia. Raportista käytettiin tietoja viidestä sarakkeesta ja neljän sarakkeen tiedot olivat logistiikkaosaston testaukselle tarpeettomia. Myöhemmin kävi ilmi, että kahta versiotestaukselle tarpeetonta tietoa käytetään ryhmälähetyksen testauksessa.

Lisenssitestaus

Lisenssitestauksessa tarkistetaan, että järjestelmään asennetut DNA-lisenssit vastaavat projektin määrittelyjä. Testauksessa on käytetty Asiakas lisenssit -raporttia. Raportin kaikki sarakkeet olivat tarpeellisia, eikä niihin tarvittu lisäyksiä.

I/O-korttien rakenne

I/O-korttien rakenteen testauksessa varmistetaan, että järjestelmän I/O-korttien asennus vastaa määrittelyjä eli tulo ja lähtökortit on aseteltu suunnitelman mukaisesti. I/O-korttien tietoja käytettiin I/O-kanavat IBM-raportissa, mutta siinä oli paljon tarpeetonta tietoa testauksen käyttöön nähden. Raportin asettelu oli myös

epälooginen logistiikkaosaston käyttöä varten. Sisältöä verrattiin Debuggerilta saatuun I/O-korttien rakenteen tulosteeseen ja halutut tiedot löytyivät raportista, ne piti vain asetella Debuggerin tulostetta vastaavaksi. I/O-korttien ja määrittelyjen yhteneväisyyttä testattiin myös pistokokeilla poistamalla I/O-kortteja ja tarkistamalla, että oikea kortti häviää Debuggerin tulosteesta. Tätä ei ole mahdollista toteuttaa IBM-raporttien avulla, joten jatkossa pistokokeet toteutetaan kuten ennenkin.

FBC-vikalaskurit

Testauksessa FBC-vikalaskureiden tilat nollattiin ja tulostettiin Debuggerilla, jotta saatiin selville kasvavatko lukemat, mikä olisi merkki kaapelointiin liittyvästä ongelmasta. Kävi kuitenkin ilmi, että tätä testiä ei voida korvata AM-IS-ajosta saatavilla tiedoilla, sillä AM-IS-ajolla ei ole mahdollista nollata laskureita. Laskureiden tilat pystyttäisiin kyllä tulostamaan, mutta tieto olisi käyttökelvotonta, sillä virheiden kerääntymiseen kulunutta aikaa ei saataisi selville. Näin ollen FBC-vikalaskureiden testaus tehdään jatkossa kuten ennenkin.

Tietoverkon verkkolaitteiden vikalaskurit

Tietoverkossa olevien verkkolaitteiden vikalaskurit tarkistetaan testauksen yhteydessä katsomalla niiden lukemat kahdesti. Lukemien kasvaminen viittaisi yhteysongelmiin. AM-IS-ajo hakee jokaisen tiedon kerran, joten lukemien tarkistaminen kahdesti vaatisi testiajon ajamista kahdesti. Nykyisellään AM-IS-ajo voidaan konfiguroida keräämään tietoja minimissään yhden päivän välein (Roihupalo 2019). Tästä syystä vikalaskureita ei lisätty IBM-raporttiin.

Ryhmälähetyskommunikaation testaus

Ryhmälähetyskommunikaatio tarkoittaa tiedon lähettämistä kaikille verkossa oleville laitteille, jotka on määriteltävä kohderyhmäksi. Verrattuna normaaliin unicastiin eli täsmälähetykseen, jolloin tieto lähetetään vain yhdelle vastaanottajalle. Testauksessa tarkistetaan, että kaikki DNA-noodit, joissa on ryhmälähetystoiminto, on konfiguroitu oikein. Sama asia saadaan tarkistettua Versio-raportista noodin, merkin ja hardware osoitteen perusteella.

Virta- ja laitevikojen testaus

Logistiikkaosasto testaa prosessiohjainpalvelimen eli PCS:n kahdennuksen toiminnan virran katketessa tai laitevikojen ilmaantuessa. Testaus edellyttää pää PCS:n sammuttamista, mitä ei pystytä mallintamaan AM-IS-ajolla eli virta- ja laitevikojen testaus suoritetaan jatkossa kuten ennenkin.

PCS-varmuuskopioinnin testaus

PCS-varmuuskopioinnin testauksella tarkistetaan, että operaattorin tekemät muutokset prosessin ohjausarvoihin varmuuskopioidaan vara-asemalle. Testauksessa on tulostettu Debuggerilla viimeisin kopiointiaika ja päivitysvirheiden määrä. Samat tiedot ovat saatavilla AM-IS-ajon tuloksista. Testaus lisätään myöhemmin osaksi olemassa olevaa raporttia.

Ajan synkronoinnin testaus

Ajan synkronoinnin testauksessa tulostetaan Debuggerilla kaikkien DNA-noodien kellot. Tulostetuista kellonajoista yksi on isäntä-kello (master clock), joka synkronoi ajat kaikkiin muihin noodeihin. Isäntä-kellon aikaa verrataan muiden noodien kellonaikoihin ja tarkistetaan ettei kellonaikojen välillä ole merkittäviä eroja. Heittoa kellonaikojen välillä saa olla enintään 100 ms. Aikojen synkronointia ei pystytty lisäämään IBM-raporttiin, sillä AM-IS-kyselyllä ei saada tietoon DNA-noodien kellonaikoja. Kyselyn tuloksiin tulostuu kyllä kellonaika, mutta se kertoo vain, milloin kyseinen tieto on tulostettu. Tämän takia ajan synkronoinnin testaus suoritetaan jatkossakin Debuggerilla.

Saatavuuskartoituksen tuloksena selvisi, että versiotestaus, lisenssien tarkistus, I/O-korttien rakenteen tarkistus ja ryhmälähetyskommunikaation testaus on mahdollista tuoda AM-IS-ajolla IBM-raporttiin. PCS-varmuuskopioinnin tarkistus saadaan myös tuotua IBM-raporttiin, mutta se lisätään myöhemmin osaksi olemassa olevaa raporttia. Muita tietoja ei ollut teknisistä syistä mahdollista tuoda AMIS-ajosta IBM-raporttiin.

6 UUDEN RAPORTIN EHDOTUS

Työn alkumäärittelyjen mukaan logistiikkaosastolle tehtäisiin yksi raportti, joka sisältäisi kaikki järjestelmätestauksen kannalta tarpeelliset tiedot. Näin testauksessa tarvitsisi tulostaa vain yksi raportti, josta tiedot löytyisivät. Raporttityökaluun tutustumisen myötä kävi kuitenkin selväksi, että raporttiin mahtuu sarakkeita rajallinen määrä, ennen kuin sen luettavuus alkaa kärsiä. Raportit luodaan Valmetin kehittämällä DNA Report Builder ohjelmalla. Raporttiehdotukseen tuli tietojen saatavuuskartoituksen jälkeen 33 tietosaraketta, kun taas olemassa olevissa raporteissa on keskimäärin alle kymmenen saraketta. Tässä vaiheessa työtä alettiin harkita myös muita mahdollisuuksia raporttien toteutukselle.

Työn tarkoituksena ei ollut tehdä muutoksia olemassa oleviin raporteihin logistiikkaosaston käyttöä varten. Tämä siksi, että raportit ovat muidenkin osastojen käytössä, eivätkä heidän käyttötarpeensa raporteille olleet tiedossa. Mahdolliset muutokset olemassa oleviin raporteihin olisivat voineet olla haitallisia muiden käyttäjien näkökulmasta. Siksi keskityttiin luomaan ratkaisuja, joilla voitaisiin joko hyödyntää olemassa olevia raporteja tai luoda uusia niiden pohjalta.

System and Networks testing – IBM -dokumenttiin (liite 1) kerättiin yhteen taulukoon kaikki 33 tietoa ja testit, joihin niitä käytetään. Taulukosta huomattiin, että tarpeiden kartoituksen jälkeen järjestelmätestauksen käyttöön sopisi kaksi jo olemassa olevaa raporttia; Versio ja Asiakaslisenssit. Näiden kahden lisäksi tarvittaisiin uusi raportti I/O-korttien rakenteen testausta varten. Tietojen keräämistä yhteen suureen raporttiin harkittiin, mutta ideasta luovuttiin raporttien selkeyden ja luettavuuden säilyttämiseksi. Lisäksi hyvin suuren raportin sisältöjen haku tietokannasta on hidasta. Harkinnassa otettiin huomioon myös raportin luomiseen käytettävä aika verrattuna sen aikaansaamiin etuihin.

Versiotestauksessa käytettävässä Versio-raportissa oli testaukselle neljä ylimääräistä tietoa. Koska kahta näistä tiedoista voitiin käyttää ryhmälähetyksen testaukseen, ei nähty syytä tehdä uutta raporttia kahden ylimääräisen sarakkeen poistamisen takia. Asiakaslisenssit-raportti taas oli hyvä sellaisenaan, eikä vaatinut mitään muutoksia. PCS-varmuuskopioinnin testaukseen käytettäviä tietoja ei

saatu lisättyä uuteen raporttiin loogisesti, joten ne lisätään myöhemmin osaksi olemassa olevaa raporttia. Ainoa tarve uudelle raportille syntyi siis I/O-korttien rakenteen testauksesta. Lopulta päätettiin luoda uusi raportti I/O-korttien rakenteen testausta varten ja jatkaa kahden olemassa olevan raportin käyttöä sellaisinaan.

I/O-korttien rakenteen testausta varten tehtävään raporttiin käytettiin mallina Debuggerin tulosteen asettelua ja sen sisältämiä tietoja. Testaajat olivat jo tottuneet käyttämään Debuggerin tulostetta, joten työn sujuvuuden kannalta oli järkevintä käyttää samaa mallia raportissa. Debuggerin tuloste myös muistuttaa I/O-korttien fyysistä järjestystä, mikä on testauksen kannalta hyvä. Taulukossa 1 näkyy raportin suunniteltu rakenne. Taulukko tulostuu raporttiin jokaisen aseman jokaista käytössä olevaa FBC:tä kohden kerran. Yhdessä FBC:ssä voi olla enintään 16 kehikollista I/O-kortteja, joten taulukossa on IBC:n alla numerointi nolasta viiteentoista.

TAULUKKO 1. I/O-korttien rakenteen raportti

NODE	FBC															
IBC	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	Ai8h	Do8	Di8	Di8			Fi4	Ao4h								
1	Not Exists															
2	Not Exists															
3	Not Exists															
4	Not Exists															
5	Di8	Di8				Do8										
6	Not Exists															
7	Not Exists															
8	Not Exists															
9	Not Exists															
10	Not Exists															
11	Not Exists															
12	Not Exists															
13	Not Exists															
14	Not Exists															
15	Not Exists															

Raportin yläreunassa näkyy aseman tunnus ja FBC:n numero. Näiden jälkeen tulostetaan varsinainen I/O-korttien rakenne eli korttien tyypit IBC-kohtaisesti. Taulukon toisella rivillä olevat numerot kertovat kortin paikan kehikossa. Raporttiin haluttiin näkyviin myös kehikot, joissa IBC ei ole käytössä. Näin ollen, kun raporttia verrataan todelliseen I/O-kaapissa olevaan korttirakenteeseen nähdään

heti, jos suunnitelma ja testattava järjestelmä eroavat toisistaan. Raportin sisältö ja muoto hyväksyttiin logistiikkaosaston esimiehellä sekä IBM-kehitystiimillä. Tämän jälkeen siirryttiin luomaan itse raporttia.

6.1 Raportin luominen

Työssä kului odotettua enemmän aikaa raporttitarpeiden kartoitukseen, mikä johti siihen, että raporttien luomiselle jäi vähemmän aikaa. Samalla kun työn pääpaino rajattiin raporttien kartoitukseen, sovittiin että raporttien luomisessa saadaan hyödyntää Valmetin raportteihin erikoistuneiden ammattilaisten apua niin paljon kuin tarpeen.

IBM:n kehitystiimissä on pääasiallisesti kaksi raporttien tekemiseen erikoistunutta työntekijää; tietokannan proseduurien tekijä sekä raporttien ohjelmoija. Proseduurien tekeminen vaatii kattavaa ymmärrystä SQL-tietokannan kyselyistä, joihin tutustumiseen ei jäänyt aikaa tämän työn aikataulun puitteissa. Tästä syystä sovittiin, että Valmetin asiantuntija luo proseduurit. Raporttien luomiseen käytettävä työkalu Report Builder taas on selkeä ja hyvin intuitiivinen käyttää eikä vaadi paljon opettelua. Työssä päästiinkin osallistumaan raportin suunnitteluun Report Builderillä.

Uusien IBM-raporttien luomiseen käytetään seuraavia ohjelmia:

- SQL Server
- Visual Studio
- DNA Report Builder

SQL Serverillä tehdään proseduurit, jotka hakee halutut tiedot AM-IS-ajon tuloksista. Jokaiselle raportille on hyvä luoda oma proseduurit, joka hakee ainoastaan raportissa käytettävät tiedot. Näin kysely tuo vain tarvittavat tiedot ja tulosten haakuun kuluva aika saadaan minimoitua.

Kun proseduurit oli valmis, voitiin aloittaa raportin ulkoasun suunnittelu Valmetin DNA Report Builderillä. Tässä apuna toimi Valmetin raporttien ohjelmoija. Ra-

porttiin kerättiin kaikki halutut tiedot ja niiden asettelu tehtiin taulukon 1 mukaisesti. Ennen kuin raportti haki tietoja, piti sen käyttämät dataluokat linkittää Visual Studiossa proseduurin tuloksiin koodaamalla. Kun tiedot oltiin linkitetty, voitiin raportin toimintaa testata asettamalla se näkyviin IBM:ssä.

Kuva 5 on kaappaus Report Builderin suunnittelunäkymästä. Kuvaa on rajattu loppumaan yhdeksännen kortin tietojen jälkeen, jotta kuva säilyy selkeänä. Rajaamaton raportti on työn liitteenä (liite 2). Raportti on jaettu ylä- ja alatunnisteisiin. Ylimpänä on koko raportin ylätunniste ReportHeader1, joka tulostuu vain kerran raportissa. Aivan alimpana on raportin ylätunnistetta vastaava alatunniste, joka myös tulostuu vain kerran. Raportin ylätunnisteessa tulostetaan raportin nimi, asiakkaan tunnus, työmaakohtainen koodi, asiakkaan nimi, osoite, maa ja isäntäjärjestelmä. Isäntäjärjestelmä tulostuu vain, jos tuloksista suodatetaan vain yhden osajärjestelmän tiedot.

ReportHeader1									
									ReportName
ID	SiteCode								
CLIENT									
ADDRESS									
COUNTRY									
Father System: fatherSyst									
GroupHeader1									
NODE	NODE								
GroupHeader2									
FBC	fbc								
IBC	C0	C1	C2	C3	C4	C5	C6	C7	C8
Detail1									
pic	C0	C1	C2	C3	C4	C5	C6	C7	C8
GroupFooter2									
GroupFooter1									
ReportFooter1									

KUVA 5. IO Structure -raportti Report Builder -ohjelmassa, rajattu, asiakkaaseen viittaavat tiedot sensuroitu

Seuraava ylä- ja alatunniste pari on GroupHeader1 ja GroupFooter1. Ne tulostuvat aina kun siirrytään uuteen noodiin. Raportissa tulostuu otsikko **NODE** ja sen jälkeen tulostetaan tietokannasta haettu noodin nimi, joka korvaa tekstin NODE. Alatunnisteessa ei ole sisältöä.

Kolmantena on GroupHeader2 ja GroupFooter2. Ylä- ja alatunnisteet tulostetaan aina kun FBC:n numero vaihtuu. Ylätunnisteessa tulostuu otsikko FBC ja sen

viereen tietokannasta haettu FBC:n numero, joka korvaa tekstin fbc. Seuraavalle riville tulostuu IBC:n ja korttipaikkojen otsikot. Alatunnisteessa tulostetaan viiva, joka erottaa FBC:t toisistaan ja selkeyttää raportin ulkoasua.

Viimeisenä tulostetaan raportin varsinainen sisältö kohdassa Detail1. Tälle kentälle ei ole ylä- ja alatunnisteparia, vaan raporttiin tulostuu vain kannasta haetut tiedot. Kenttä pic korvataan IBC:n numerolla, jotka tulostetaan nolasta viiteentoista, vaikkei IBC olisi käytössä. Korttipaikat C0:sta C15:een korvataan tietokannasta haetuilla korttitiedoilla. Jos korttipaikka on tyhjä, näkyy se raportissa tyhjänä. Jos taas IBC ei ole käytössä, tulostuu rivin alkuun teksti Not Exists.

Kaikki tämä tehtiin Valmetin kehityspalvelimella. Näin varmistettiin, etteivät mahdolliset virheet vaikuta tuotantopalvelimen toimintaan. Lisäksi tuotantopalvelimen tietoliikenne on pysäytettävä, kun IBM:ään ladataan uusia raportteja. Raportin luomiseen kului noin 6 tuntia, mikä tarkoittaa, että IBM olisi ollut pysäytettynä koko sen ajan, jos kehitystyö olisi tehty suoraan tuotantopalvelimelle. Kun raportin ulkoasu ja toiminta oli varmistettu kehityspalvelimella, voitiin se siirtää tuotannon puolelle, jolloin se on kaikkien IBM:ää käyttävien työntekijöiden saatavilla.

Jos raporttiin olisi tarvinnut hakea tietoa, jota ei vielä nykyisellä AM-IS-ajolla saatu, olisi raportti näkynyt vasta uuden AM-IS-ajon version tullessa käyttöön. Koska raportti ei vaatinut muutoksia AM-IS-ajoon, voidaan se tulostaa lähes kaikille tehtaille. Raportti vaatii silti viimeisimmän AM-IS-version ajamista, eli jos tehtaalla on ajettu hyvin aikainen versio AM-IS-ajosta, ei raportti välttämättä näy, mutta tällöin sama ongelma koskee kaikkia uudempia raportteja.

Kuva 6 on kaappaus erään asiakkaan I/O-rakenteen raportista. Kuvasta on sensuroitu asiakkaaseen viittaavat tiedot salassapitosopimuksen mukaisesti. Kuva rajattiin loppumaan yhdeksännen I/O-kortin jälkeen, jotta raportin luettavuus säilyisi. Rajaamaton kuvakaappaus on työn liitteenä (liite 3). Kuvassa näkyy vain järjestelmän aakkosjärjestyksessä ensimmäinen noodi ja sen kaksi käytössä olevaa FBC:tä. Kuvasta saa kuitenkin idean, miltä raportti näyttää ja kuinka tiedot siinä asettuvat. Raportissa tulostuu kaikkien noodien kaikki käytössä olevat FBC:t.

IO Structure

Father System:									
NODE	HP01								
FBC	2								
IBC	C0	C1	C2	C3	C4	C5	C6	C7	C8
0	BIU8	BIU8	BIU8	BIU8	BIU8			BOU8	BOU8
1	BOU8	BIU8	BIU8	BIU8	BOU8	BIU8	BIU8	BIU8	BOU8
2	BOU8	BIU8	BIU8	BOU8	BIU8	BIU8	BOU8	BIU8	BIU8
3				AIU8		AIU8	AIU8	AIU8	AIU8
4	AIU8	AIU8	AIU8	AIU8	AIU8			AOH4	AOH4
5	AOH4	AOH4	AOH4	AOH4	AOH4				
6	BOU8	BIU8	BIU8				BIU8	BIU8	BIU8
7	BOU8	BIU8	BIU8	BOU8	BIU8	BIU8	BOU8	BIU8	BIU8
8	AIU8	AIU8	AIU8	AIU8	AIU8	AIU8			AOH4
9	BOU8	BIU8	BIU8	BIU8	BOU8	BIU8	BIU8	BIU8	BOU8
10	BOU8	BIU8	BIU8	BIU8	BOU8	BIU8	BIU8	BIU8	BOU8
11	BOU8	BIU8	BIU8	BIU8	BOU8	BIU8	BIU8	BIU8	BOU8
12	BOU8	BIU8	BIU8	BIU8	BOU8	BIU8	BIU8	BIU8	BOU8
13	BOU8	BIU8	BIU8	BIU8	BOU8	BIU8	BIU8	BIU8	BOU8
14	BOU8	BIU8	BIU8	BIU8	BOU8	BIU8	BIU8	BIU8	BOU8
15	Not Exists								
FBC	8								
IBC	C0	C1	C2	C3	C4	C5	C6	C7	C8
0	BOU8	BIU8	BIU8	BOU8	BIU8	BIU8	BOU8	BIU8	BIU8
1	BOU8	BOU8	BOU8			BIU8	BIU8	BIU8	BIU8
2	BOU8	BOU8	BOU8			BIU8	BIU8	BIU8	BIU8
3	AIU8	AIU8	AIU8	AIU8	AIU8		AIU8	AIU8	AIU8
4	AOH4	AOH4	AOH4	AOH4	AOH4	AOH4	AOH4	AOH4	AOH4
5	AIU4	PLUN	AIU4	PLUN		PLUN	PLUN	PLUN	
6	BOU8	BIU8	BIU8				BIU8		AIU8
7	Not Exists								
8	BOU8	BIU8	BIU8	BIU8	BOU8	BIU8	BIU8	BIU8	
9	BOU8	BIU8	BIU8	BIU8	BOU8	BIU8	BIU8	BIU8	
10	BOU8	BIU8	BIU8	BIU8					
11	BOU8	BIU8	BIU8	BIU8	BIU8	BIU8	BIU8	BOU8	BIU8
12	Not Exists								
13	Not Exists								
14	Not Exists								
15	Not Exists								

KUVA 6. I/O-korttien rakenteen IBM-raportti, rajattu, asiakkaaseen viittaavat tiedot sensuroitu

Kuvasta 6 nähdään, että raportti vastaa hyvin taulukon 1 suunnitelmaa. Muutoksena taulukon rakenteeseen vaihdettiin NODE ja FBC tiedot omille riveilleen. Näin saatiin aikaiseksi tuloste, jossa saman noodin FBC:t tulostuvat peräkkäin järjestyksessä. Tulokset järjestyvät ensin aakkosjärjestyksessä NODE:n mukaan ja sitten numerojärjestyksessä FBC:n mukaan. Tällä asettelulla NODE ei tulostu kuin kerran jokaista noodia kohden kuten kuvasta 6 nähdään.

Valmis raportti hyväksyttiin vielä logistiikkaosaston esimiehellä ja todettiin, että se vastaa testauksen tarpeita. Raporttiin tehdään jatkossa muutoksia ja päivityksiä testauksen yhteydessä ilmaantuvien tarpeiden mukaan.

7 POHDINTA

Ennen työn alkua määriteltiin, että työn aikana luotaisiin kaksi raporttia, logistiikka osastolle sekä huolto-osastolle. Lopputuloksena oli raporttiehdotus vain logistiikkaosastoa varten. Tähän suurimpana syynä oli huolto-osaston jakautuminen useaan toimipisteeseen, mikä aiheutti hankaluuksia raporttitarpeiden kartoitukselle. Huolto-osastolta oltiin aikaisemmin ilmaistu tarpeita raportille, mutta työn alettua näitä tahoja ei saatu tavoitettua ennen kuin vasta aivan työn loppumetreillä. Jälkeenpäin katsottuna oli hyvä asia, ettei huolto-osaston raporttitarpeisiin käytetty liikaa aikaa. Näin logistiikkaosaston raportin määrittelylle jäi enemmän aikaa ja myöhemmin kävikin ilmi, että suuri osa huolto-osaston tarpeista oli jo Valmetin IBM-järjestelmää kehittävän ryhmän työlistalla.

Työn tavoitteena oli myös pienentää logistiikka osaston työkuormaa vähentämällä järjestelmätestauksen manuaalista testausta IBM-raportin avulla. Testauksessa oli yhteensä yhdeksän manuaalista tiedonhakua vaativaa testiä. Uuden raportin avulla niistä pystyttiin korvaamaan yksi. Prosentuaalisesti testien väheneminen ei ole kovin vakuuttava, mutta käytännön työn kannalta kaikki ajansäästö testauksessa on pelkästään hyvä asia. Lisäksi aikaisemmin testauksessa jouduttiin toistamaan I/O-rakenteen testaus yksitellen jokaisen FBC:n kohdalla, kun taas raportti tulostaa ne kaikki kerralla. Uuden raportin lisäksi yksi manuaalinen testi saatiin korvattua olemassa olevalla raportilla, joten oikeastaan työn avulla saatiin karsittua kaksi manuaalista testiä. Varsinaiset tulokset ajansäästöissä nähdään, kun uusi raportti ja päivitetty testausohje tulevat testauksessa käyttöön.

Kokonaisuutena työ oli onnistunut, vaikka kaikkiin tavoitteisiin ei päästykään. Alkuperäinen suunnitelma kahdesta uudesta raportista oli kunnianhimoinen, mutta olisi ollut toteutettavissa, jos työhön olisi ollut käytettävissä enemmän aikaa. Aikataulu olikin yksi työn osa, joka muuttui työn aikana huomattavan paljon. Aikataulua tehdessä luultiin, että raporttien ohjelmoiminen veisi suurimman osan työhön varatusta ajasta, kun todellisuudessa raporttien sisältöjen määrittely oli työn

aikaavievin osa. Samoin raportointi ja tiedonkeruujärjestelmiin tutustumiseen kuluva aikaa aliarvioitiin aikataulua laatiessa. Loppujen lopuksi raportin ohjelmoimiseen kului työssä vähiten aikaa.

Jatkossa raporttiin tehdään muutoksia käytössä ilmenevien tarpeiden mukaan IBM-kehitysryhmän toimesta. Jos raportissa ei ilmene puutteita, jatkuu sen käyttö sellaisenaan niin kauan kuin järjestelmätestaus jatkuu testausohjeen mukaisesti. On kuitenkin mahdollista, että järjestelmän testaus uudistuu täysin ja muuttuu kokonaan automatisoiduksi lähivuosina. Varsinaista suunnitelmaa tälle muutokselle ei vielä ole, mutta mahdollisia testausmenetelmiä on jo alettu etsimään. Tavoitteena olisi poistaa kaikki manuaalinen testaus ja lisätä testauksen tehokkuutta ja varmuutta automaattisella testauksella, jonka tuloksista nähdään heti, jos jokin on pielessä.

LÄHTEET

Ala-Ruisku, T. 2009. INSTALLED BASE INFORMATION: Ensuring Customer Value and Profitability after the Sale. Tuotantotalous. Helsingin teknillinen korkeakoulu. Väitöskirja.

Control Point. N.d. Control point si Valmet. Luettu 26.3.2019.
<https://www.control-point.ro/proiectare-executie-instalatie-automatizare-valmet>

Elfving, J. Energiatekniikan diplomi-insinööri opiskelija. Raporttityökälusta. Sähköpostiviesti. julius.elfving@valmetpartners.com. Luettu 13.2.2019.

Heikkinen, T. Oulun ammattikorkeakoulu. Valmet DNA (Metso DNA) How-to. Päivitetty 31.1.2019. Luettu 3.4.2019. http://www.tekniikka.oamk.fi/~timohei/?p=20opintojaksot/0100TL6031/35howto&t=86pic_fi.html

Jokinen, S. Logistiikkaosaston esimies. 2019a. Haastattelu 20.2.2019. Haastattelija Silvonen, O. Tampere. Valmet.

Jokinen, S. Logistiikkaosaston esimies. 2019b. Haastattelu 15.3.2019. Haastattelija Silvonen, O. Tampere. Valmet.

Kiviniemi, T. 2016. AM-IS installation manual. Julkaistu 1.4.2016. Vaatii tunnustautumisen. Luettu 10.2.2019
<https://valmet.sharepoint.com/teams/dnard/DNALifecycle/SitePages/Home.aspx>

Kleemoja, E. 2016. IB Manager – Presentation. Julkaistu 22.3.2016. Vaatii tunnustautumisen. Luettu 10.2.2019.
<https://valmet.sharepoint.com/teams/dnard/DNALifecycle/SitePages/Home.aspx>

Lahtinen, S., Gaubusseau, H. & Elfving, J. 2017. IBM Developer's notes. Julkaistu 29.8.2014. Päivitetty 26.7.2017. Vaatii tunnustautumisen. Luettu 12.2.2019.
<https://valmet.sharepoint.com/teams/dnard/DNALifecycle/SitePages/Home.aspx>

Laine, P. 2015. Valmet becomes stronger as a result of acquiring Process Automation Systems. Julkaistu 15.1.2015. Luettu 7.2.2019.
<https://www.valmet.com/globalassets/investors/valmet-as-an-investment/acquisition-of-process-automation-systems/valmet---process-automation-systems-acquisition-2.pdf>

McGehee, B., Kraft, R. & Shepker, M. 2000. Microsoft SQL Server 7.0. Tehokäyttäjän opas. Suom. Saxberg, P. Jyväskylä: Gummerus Kirjapaino Oy. Alkuperäinen teos 1999.

Oulun ammattikorkeakoulu. N.d. Prosessiautomaatio. Pdf. Luettu 26.3.2019.
http://www.oamk.fi/~kurki/automaatiolabrat/TTT/24_Prosessiautomaatio.pdf

Roihupalo, K. Suunnitteluinsinööri. AM-IS-keräilyn tiedoista. Sähköpostiviesti. kimmo.roihupalo@valmet.com. Luettu 7.3.2019.

Saccani, N., Alghisi, A., & Borgman, J. 2012. The Value and Management Practices of Installed Base Information in Product-Service Systems. APMS. https://link.springer.com/chapter/10.1007%2F978-3-642-40361-3_53

Strömman, M., Hirvonen, J., Hukki, K. & Tommila, T. 2007. Automaatiosuunnittelun prosessimalli. Yhteiset käsitteet verkottuneen suunnittelun perustana. Helsinki. Verkkojulkaisu 2010. www.automaatioseura.fi

Tyynelä, M. 2017. AUT integration: customer presentation. Julkaistu 23.5.2017. Luettu 7.2.2019. https://www.automaatioseura.fi/site/assets/files/1603/sas_asaf-teemapaiva_10-5-2017_markku_tyynela_automaatiojarjestelman_tietoturvan_tekninen_toteutus.pdf

Tyynelä, M. Ohjelmistopäällikkö. Automaatioprojektin malli. Sähköpostiviesti. markku.tyynela@valmet.com. Luettu 12.3.2019.

Valmet Oyj. 2017. Valmet presentaiton. Julkaistu 20.10.2017. Luettu 7.2.2019. https://www.valmet.com/globalassets/about-us/valmet-in-brief/general-presentation_2017_10_eng_final.pdf

Valmet Oyj. 2019a. 220 vuotta teollista historiaa. Luettu 17.1.2019. <https://www.valmet.com/fi/campaign/220vuotta/>

Valmet Oyj. 2019b. Valmetin Johtoryhmä. Luettu 30.1.2019. <https://www.valmet.com/fi/valmet-yrityksena/valmetin-johto/johtoryhma/>

Valmet Oyj. 2019c. Valmet yrityksenä. Luettu 17.1.2019. <https://www.valmet.com/fi/valmet-yrityksena/>

Valmet Oyj. 2019d. Taloudellista tietoa. Luettu 31.1.2019.- <https://www.valmet.com/fi/sijoittajat/taloudellista-tietoa/>

LIITTEET

Liite 1. System and network testing - IBM

1 (9)

System and Networks testing – IBM

Report

TABLE OF CONTENTS

Revision History	38
Appendixes.....	38
1. INTRODUCTION.....	39
1.1 Purpose.....	39
1.2 General.....	39
2. System testing procedures	39
2.1 Version report	39
2.2 Licenses	40
2.3 I/O-channel structure	40
2.4 FBC fault counters (NO).....	41
2.5 Network device fault counters (NO).....	41
2.6 Multicast communication	41
2.7 Power or hardware failure (NO)	42
2.8 PCS data backup	42
2.9 Time synchronization (NO)	42
3. Report proposal	Error! Bookmark not defined.
3.1 I/O Channel testing	45

Revision History

Descrip- tion	Date	Modifier	Revi- sion	Comment
First draft	22.2.2019	OSi	0.0	

Appendixes

Appendix	Description	Document ID
Appendix 1	System Testing procedure plan	..
Appendix 2	IO Channel structure	

1 INTRODUCTION

1.1 Purpose

This document describes the correlations in the System and Networks System testing procedure and the reports in the IBM and gives a proposal for new report(s).

1.2 General

System and Networks team runs multiple tests on the built system before handover to Automation system delivery project. Some of the tests require the use of different IBM reports and DNA Debugger. The purpose of this document is to find out if the information from different reports can be summarized into one report, also if the data checked via Debugger can be also added to the IBM report.

2 SYSTEM TESTING PROCEDURES

The System testing procedure – Plan describes the steps of the testing procedure. The testing plan requires the tester to run the AM-IS-test run to use the different IBM reports. Listed below are all the tests that could possibly be added to a IBM report. Quotes from the testing plan are under each heading to give an idea of the test and its purpose. The correlating reports are referenced. If there is no correlating report, the tested data will be searched from the AM-IS test report. If the data is not on the AM-IS test report, the possibility of adding the data on the test will be researched.

2.1 Version report

“Verify that all version information is according to specification for all nodes. Discuss with project if there are differences in version, e.g. pilot versions etc.”

IBM report: Version Report

Report Column data	Necessary (yes/no)
Node	y
HW Address	y (used for multicast testing)
Token	y (used for multicast testing)
NCU Code	n
QEMGR	y
EA Version	y
IA Version	y
Collection	y
Setup	n

2.2 Licenses

“Verify that all DNA licenses are according to system design and sales specification i.e. what has been agreed. Also verify that all 3rd party licenses are installed accordingly. In case of temporary DNA licenses, confirm from project if they are valid.”

IBM report: Customer Licenses

Report Column data	Necessary (yes/no)
License Information	y
License ID	y
License Type	y
Code	y
Product	y
Number Licensed	y

2.3 I/O-channel structure

“The purpose of this test is to ensure that the I/O allocation, which is online in the DCS system, is according to definitions.

Open debugger in system mode and print the I/O structure from all FBC.“
. Debugger print capture was deleted from here.

The report should follow the form of the print from debugger

IBM report: I/O Channels

Report Column data	Necessary (yes/no)
Node	y
FBC	y
IBC	y
Version	n
IBC Serial	n
Channels	n
Card	y
ROM	n
Type	y
Card Serial	n

“Verify that I/O definitions matches to I/O cards. Spot check few I/O cards by removing them and verifying that correct system alarm is triggered.”

Spot checking is not possible via IBM reports.

2.4 FBC fault counters (NO)

“The purpose of this test is to ensure that there are no cabling related problems, which could cause malfunctioning of the fieldbus communication.”

The counters are cleared and then printed on the debugger.

Requires the resetting of the counters which cannot be done using AM-ISq.

2.5 Network device fault counters (NO)

“Verify that the fault counters are not increasing by checking them twice at different times. For Moxa -devices “MoxaDiag” -tool can be utilized, other devices need to be verified via telnet or http.”

Depending on the time between the two prints this may or may not be possible. The minimum difference between two AM-ISq’s is one day. If the counters need to be printed closer to each other, it is not possible to check the network device fault counters via IBM reports.

2.6 Multicast communication

“Verify that all nodes are connected to DNA System in terms of multicast communication (NCU). Open debugger in system mode and print DNA system structure:

p(rint) s(tructure)

All nodes in DNA system, that have NCU configured, should be listed.”

Debugger print capture was deleted from here.

IBM report: Version report

Only need to check Node, token and hardware address, which are shown in the Version report. Add to testing plan that Multicast communication can be checked from Version report.

2.7 Power or hardware failure (NO)

“The purpose of this test is to verify that the PCS redundancy works properly in case of physical ACN node DC power failure or fatal hardware failure. Failure should also trigger correct system diagnostic alarms. Do following tests for all redundant PCS’

The time between the active PCS becoming faulty and passive PCS activating (i.e. switchover) must be less than 5 seconds. Verify switchover time with debugger in system mode. Repeat for all FBC in the PCS node.”

Test requires the powering down of main PCS node, which cannot be replicated via AM-ISq.

2.8 PCS data backup

“Verify backup operation from all PCS diagnostic sensors with the debugger (system mode):

p(rint) v(riable) :e:di:<station>:backup

‘lastscs’ -member should be quite close to current time and ‘upderr’ should be ‘0’.”

This information can be added to the IBM reports, the AM-ISq already retrieves it from the system. Necessary information would be node, lastscs and upderr members. This information could be added to the new report of the I/O-channel structure, or make modifications to an existing report to add them in.

2.9 Time synchronization (NO)

The clocks of all nodes are printed and checked if they differ from the “master clock”. There should not be more than 100ms deviations between nodes.

AM-ISq cannot print the clocks of the nodes. It only prints the time of last change or the time of the query.

3 REPORT PROPOSAL

Report proposals were made based on the information in the System testing procedures chapter. The ideal would be to fit all the required information into one report, but the Report tool might have some limits which will be taken into consideration. This report proposal is meant for a new report, the existing reports are not affected by it.


In the table below is listed all the used data and the tests they are used for.

Report Column data	Test
Node	Version report & I/O-channel structure & PCS data backup
HW Address	Multicast testing
Token	Multicast testing
QEMGR	Version report
EA Version	Version report
IA Version	Version report
Collection	Version report
License Information	Licenses
License ID	Licenses
License Type	Licenses
Code	Licenses
Product	Licenses
Number Licensed	Licenses
FBC	I/O-channel structure
IBC	I/O-channel structure
Card 0	I/O-channel structure
Card 1	I/O-channel structure
Card 2	I/O-channel structure
Card 3	I/O-channel structure
Card 4	I/O-channel structure
Card 5	I/O-channel structure
Card 6	I/O-channel structure
Card 7	I/O-channel structure
Card 8	I/O-channel structure
Card 9	I/O-channel structure
Card 10	I/O-channel structure
Card 11	I/O-channel structure
Card 12	I/O-channel structure
Card 13	I/O-channel structure
Card 14	I/O-channel structure
Card 15	I/O-channel structure
lastscs	PCS data backup
upderr	PCS data backup

Because the Licenses test can be done using the existing Customer Licenses report there is no need to make a new report for it. Also, the Version report is mainly used in the Version testing, and two of the unnecessary parameters are used in the Multicast testing. The PCS data backup testing will be added to an existing report later. The biggest changes concern the I/O-channel testing.

The objective was to fit all necessary information into one big report. But because there would be so many columns in the report the readability would suffer. Therefore, only one new report will be made for the I/O-channel structure testing.

Liite 2. IO Structure -raportti Report Builderissa

<input type="checkbox"/> ReportHeader1		ReportName																
ID	SiteCode																Scan Date:	dataSetDate
CLIENT																	Print Date:	PrintDate
ADDRESS																		PrintTime
COUNTRY																		
Father System: fatherSyst																		
<input type="checkbox"/> GroupHeader1																		
NODE NODE																		
<input type="checkbox"/> GroupHeader2																		
FBC	fbc																	
IBC	C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15		
<input type="checkbox"/> Detail1																		
pic	C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15		
<input type="checkbox"/> GroupFooter2																		
<input type="checkbox"/> GroupFooter1																		
<input type="checkbox"/> ReportFooter1																		

Liite 3. IO Structure IBM-raportti

IO Structure



Scan Date: 9.11.2018 11:41:05
Print Date: 2.4.2019 14:35:00

Father System:																
NODE HP01																
FBC 2																
	C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15
0	BIU8	BIU8	BIU8	BIU8	BIU8	BIU8	BOU8	BOU8	BOU8	BIU8	BIU8	AIU8	AIU8	AIU8	AIU8	C15
1	BOU8	BIU8	BIU8	BIU8	BOU8	BIU8	BOU8	BIU8	BOU8	BOU8	BIU8	BIU8	BOU8	BIU8	BIU8	BIU8
2	BOU8	BIU8	BIU8	BOU8	BIU8	BOU8	BOU8	BIU8	BOU8	BOU8	BIU8	BIU8	BOU8	BIU8	BIU8	BIU8
3	AIU8	AIU8	AIU8	AIU8	AIU8	AIU8	AIU8	AIU8	AIU8	AIU8	AIU8	AIU8	AIU8	AIU8	AIU8	AIU8
4	AIU8	AIU8	AIU8	AIU8	AIU8	AIU8	AOH4	AOH4	AOH4	AOH4	AOH4	AOH4	AOH4	AOH4	AOH4	AOH4
5	AOH4	AOH4	AOH4	AOH4	AOH4	AOH4	AOH4	AOH4	AOH4	AOH4	AOH4	AOH4	AOH4	AOH4	AOH4	AOH4
6	BOU8	BIU8	BIU8	BOU8	BIU8	BOU8	BOU8	BIU8	BIU8	BIU8	BIU8	BIU8	BIU8	BIU8	BIU8	BIU8
7	BOU8	BIU8	BIU8	BOU8	BIU8	BOU8	BOU8	BIU8	BIU8	BIU8	BIU8	BIU8	BIU8	BIU8	BIU8	BIU8
8	AIU8	AIU8	AIU8	AIU8	AIU8	AIU8	AIU8	AIU8	AIU8	AIU8	AIU8	AIU8	AIU8	AIU8	AIU8	AIU8
9	BOU8	BIU8	BIU8	BIU8	BOU8	BIU8	BIU8	BIU8	BOU8	BIU8	BIU8	BIU8	BOU8	BIU8	BIU8	BIU8
10	BOU8	BIU8	BIU8	BIU8	BOU8	BIU8	BIU8	BIU8	BOU8	BIU8	BIU8	BIU8	BOU8	BIU8	BIU8	BIU8
11	BOU8	BIU8	BIU8	BIU8	BOU8	BIU8	BIU8	BIU8	BOU8	BIU8	BIU8	BIU8	BOU8	BIU8	BIU8	BIU8
12	BOU8	BIU8	BIU8	BIU8	BOU8	BIU8	BIU8	BIU8	BOU8	BIU8	BIU8	BIU8	BOU8	BIU8	BIU8	BIU8
13	BOU8	BIU8	BIU8	BIU8	BOU8	BIU8	BIU8	BIU8	BOU8	BIU8	BIU8	BIU8	BOU8	BIU8	BIU8	BIU8
14	BOU8	BIU8	BIU8	BIU8	BOU8	BIU8	BIU8	BIU8	BOU8	BIU8	BIU8	BIU8	BOU8	BIU8	BIU8	BIU8
15	Not Exists															
FBC 8																
0	BOU8	BIU8	BIU8	BOU8	BIU8	BIU8	BOU8	BIU8	BIU8	BOU8	BIU8	BIU8	BIU8	BIU8	BIU8	BIU8
1	BOU8	BOU8	BOU8	BOU8	BIU8	BIU8	BIU8	BIU8	BIU8	BIU8	BIU8	BIU8	BIU8	BIU8	BIU8	BIU8
2	BOU8	BOU8	BOU8	BOU8	BIU8	BIU8	BIU8	BIU8	BIU8	BIU8	BIU8	BIU8	BIU8	BIU8	BIU8	BIU8
3	AIU8	AIU8	AIU8	AIU8	AIU8	AIU8	AIU8	AIU8	AIU8	AIU8	AIU8	AIU8	AIU8	AIU8	AIU8	AIU8
4	AOH4	AOH4	AOH4	AOH4	AOH4	AOH4	AOH4	AOH4	AOH4	AOH4	AOH4	AOH4	AOH4	AOH4	AOH4	AOH4
5	AIU4	PLUN	AIU4	PLUN	PLUN	PLUN	PLUN	PLUN	PLUN	PLUN	PLUN	PLUN	PLUN	PLUN	PLUN	PLUN
6	BOU8	BIU8	BIU8	BOU8	BIU8	BIU8	BOU8	BIU8	BIU8	BIU8	BIU8	BIU8	BIU8	BIU8	BIU8	BIU8
7	Not Exists															
8	BOU8	BIU8	BIU8	BOU8	BIU8	BIU8	BOU8	BIU8	BIU8	BOU8	BIU8	BIU8	BOU8	BIU8	BIU8	BIU8
9	BOU8	BIU8	BIU8	BOU8	BIU8	BIU8	BOU8	BIU8	BIU8	BOU8	BIU8	BIU8	BOU8	BIU8	BIU8	BIU8
10	BOU8	BIU8	BIU8	BOU8	BIU8	BIU8	BOU8	BIU8	BIU8	BOU8	BIU8	BIU8	BOU8	BIU8	BIU8	BIU8
11	BOU8	BIU8	BIU8	BOU8	BIU8	BIU8	BOU8	BIU8	BIU8	BOU8	BIU8	BIU8	BOU8	BIU8	BIU8	BIU8
12	Not Exists															
13	Not Exists															
14	Not Exists															
15	Not Exists															