



# **TIETOKANTAOHJELMIEN KÄYTTÖ PROSESSITEOLLISUUDEN INSTRUMENTOINNISSA**

Anssi Lanamo

Opinnäytetyö  
Helmikuu 2014  
Sähkötekniikan ko.  
Automaatiotekniikka

TAMPEREEN AMMATTIKORKEAKOULU  
Tampere University of Applied Sciences

## TIIVISTELMÄ

Tampereen ammattikorkeakoulu  
Sähkötekniikan koulutusohjelma  
Automaatiotekniikka

LANAMO, ANSSI:

Tietokantaohjelmien käyttö prosessiteollisuuden instrumentoinnissa

Opinnäytetyö 50 sivua, joista liitteitä 4 sivua  
Helmikuu 2014

---

Neste Jacobs Oy toimii pääsuunnittelijana monissa Porvoon öljynjalostamolle tehtävistä projekteista. Prosessiteollisuuden instrumentoinnissa suuremmat projektit käsittelevät usein satoja instrumentteja. Jotta instrumenttien tietojen käsittely olisi mahdollisimman helppoa, on avuksi otettu tietokantapohjaisia suunnitteluohjelmia. Tässä opinnäytetyössä tutustutaan Neste Jacobs Oyn tapaan käyttää tietokantapohjaisia suunnitteluohjelmia instrumentoinnissa.

Tämän opinnäytetyön tarkoituksena on antaa noviisille pohjatiedot, mihin tietokantapohjaisia suunnitteluohjelmia käytetään instrumentoinnissa ja mitä niillä voidaan tehdä. Samalla yritetään kyseenalaistaa instrumentoinnin tiedon käsittelyn tehokkuutta ja käytössä olevia työtapoja. Opinnäytetyössä on esimerkkiohjelmana käytetty ALMA NG -suunnitteluohjelmaa, joka on Neste Jacobs Oyn instrumentointiosaston käytössä.

Työ toteutettiin yhteistyössä Neste Jacobs Oyn Instrumentointiosaston kanssa. Tehtävän ajaksi minut palkattiin osa-aikaiseksi työntekijäksi. Osallistuin Porvoon öljynjalostamolle tehtävään projektiin, jonka tarkoituksena on kahden jalostamon pääuunin uusiminen. Sain käyttööni tarvittavat toimistotyökalut ja oman työhuoneen.

Opinnäytetyö sisältää pohjatiedot tietokannoista ja niiden historiasta. Tämän lisäksi työssä on esitelty ALMA-suunnitteluohjelmaa ja sen mahdollisuuksia. Opinnäytetyö on Neste Jacobs Oyn käytössä, ja sen avulla voidaan perehdyttää uusia työntekijöitä tietokantojen käyttöön instrumentoinnissa.

## ABSTRACT

Tampere University of Applied Sciences  
Degree Programme in Electrical Engineering  
Automation technology

LANAMO, ANSSI:

The use of database programs in the process industry instrumentation

Bachelor's thesis 50 pages, appendices 4 pages

February 2014

---

Neste Jacobs Ltd is the chief designer at many projects done in Porvoo oil refinery. Bigger projects in the process industry instrumentation often deal with hundreds of instruments. In order to ease the handling of the data of these instruments, database-driven design programs were introduced. In this bachelor's thesis is introduced Neste Jacobs Ltd's use of database-driven software in the instrumentation design.

Purpose of this bachelor's thesis is to give novice base information about what are database-driven programs used for in the instrumentation design and what they can do. At the same time it is trying to call into question the efficiency of the information processing in instrumentation design and the existing ways of working. The example program used in this thesis is ALMA NG design program that Neste Jacobs Ltd's instrumentation department uses.

The work was carried out in cooperation with Neste Jacobs Ltd's Instrumentation department. I was hired for the task as a part-time worker. I took part in a project designated to Porvoo oil refinery, the purpose of which is the renewal of two of the refiners main ovens . I was given the necessary office tools and my own office room.

The thesis includes the base information about databases and their history. In addition it also introduces ALMA design program and what it can be used for. The thesis is in Neste Jacobs Ltd's disposal and can be used to introduce new employees to the use of databases in instrumentation.

---

Key words: process industries, instrumentation, instruments, databases, design programs

## SISÄLLYS

1	JOHDANTO .....	6
2	TYÖYMPÄRISTÖ .....	7
	2.1 Neste Jacobs Oy .....	7
	2.2 Työn suorittaminen .....	8
	2.3 Työn tavoitteet .....	9
3	NYKYAIKAINEN TIETOKANTA .....	10
	3.1 Tietokanta ja Suunnittelujärjestelmä .....	10
	3.2 Tietokantojen historiaa .....	12
	3.3 Relatiopohjaisen tietokannan perusteet .....	13
	3.3.1 Tiedon rakenne .....	14
	3.3.2 Tiedon käsittely .....	16
	3.3.3 Tiedon eheys .....	18
	3.4 Oliot ja ALMA .....	19
	3.5 SQL .....	21
4	ALMA .....	22
	4.1 Perustiedot .....	22
	4.2 Katsaus tuoteperheeseen .....	23
	4.3 Tietokannan malli .....	25
	4.4 Käyttöliittymä .....	26
	4.4.1 Käyttöliittymän näkymät .....	27
	4.4.2 Attribuutti-ikkuna .....	31
	4.5 Symbolikorvaus .....	34
5	INSTRUMENTOINTI ALMA:SSA .....	36
6	TIETOKANTOJEN KÄYTTÖ .....	40
	6.1 ALMA:n käyttäminen .....	40
	6.2 Excel-taulukot ja import .....	41
	6.3 Projektinhallinnan ongelmat .....	43
7	TYÖN TARKASTELU JA LOPPUSANAT .....	45
	LIITTEET .....	47
	Liite 1. ALMA käyttöliittymä .....	47
	Liite 2. Generointi .....	48
	Liite 3. Valmis instrumenttikuva Sivu 1(2) .....	49
	Liite 3. Valmis instrumenttikuva Sivu 2(2) .....	50

**ERITYISSANASTO**

ALMA	Alma Consulting Oy:n kehittämä suunnittelujärjestelmä.
ALMA NG	Neste Jacobs Oy:n käytössä oleva versio ALMA-järjestelmästä.
Kenttä	Pienin osa, vastaa taulukkolaskentaohjelman saraketta. Saman sarakkeen tiedot täytyy olla samaa tietotyyppiä.
NJ	Neste Jacobs Oy
Perusavain	Tietty kenttä (tai kentät), jonka arvot erottavat taulukon tietueet yksiselitteisesti toisistaan.
Relaatio	Tietoa yhdestä aiheesta, kuten esimerkiksi asiakkaat, tilaukset tai varasto, käyttäjä näkee relaatiot taulukoina.
SQL	Structured Query Language. Kieli, jolla voidaan muokata tietokannan rakennetta, kysellä ja päivittää tietokannan tietoja.
Taulu	Sisältää useita tietueita joista muodostuu taulukko. Voi olla myös tyhjä (ei yhtään tietuetta).
Tietue	Taulun yksi rivi, joka sisältää yhteen aiheeseen liittyvät tiedot. Tietue sisältää useita kenttiä.
Viiteavain	Tämä on taulun kenttä, jonka avulla kahden taulun välille muodostetaan yhteys.
Viite-eheys	Käsite tai menetelmä, jonka avulla useassa taulussa olevat tiedot pitävät keskenään paikkansa.

## 1 JOHDANTO

Ennen tietokoneiden yleistymistä kaikki yrityksen olennainen tieto laitettiin arkistoon talteen. Pienissäkin yrityksissä tämä tarkoitti hyllyittäin kansioihin lajiteltuja dokumentteja. Selataksaan kansioihin lajiteltua tietoa, käyttäjän oli mentävä arkistoon ja etsiä sieltä tarvittavat dokumentit. Tiedonetsinnän helpottamiseksi tiedot oltiin lajiteltu yhdellä tai useammalla tavalla, aakkosjärjestys, aikajärjestys, mallinumero.

Tietokanta on kokoelma toisiinsa liittyvää tietoa, kuten pankin asiakasrekisteri tai kirjaston kortisto tai tässä opinnäytetyössä käsiteltävä instrumenttilaitekanta. Tietokannan pohjalla on tietoa joka on taulukkomuotoista, esimerkiksi taulukkolaskentaohjelmalla luotu lista. Jos taulukko sisältää hyvin paljon tietoa, tuhansia rivejä, on tiedon etsiminen hidasta. Tietokantaohjelmalla tieto voidaan etsiä ns. kyselyn avulla, eikä taulukkoa tarvitse selata manuaalisesti.

Yrityksestä riippuen käytössä on erilaisia tietokantamalleja. Ohjelma, sen ylläpito ja käyttökoulutus ostetaan joltain ulkopuoliselta yritykseltä. Pienillä yrityksillä on monesti valmis, kaupallinen tietokanta, koska se on halvin vaihtoehto. Suuremmilla yrityksillä on varaa ostaa omaan käyttöön räätälöity tietokantapohja ja ohjelma. Kaikkia omia työntekijöitä ei välttämättä kannata kouluttaa ohjelman tehokäyttäjiksi, muutama riittää. Suurilla yrityksillä voi olla käytössä useampikin heille räätälöity kanta, joiden tiedot tulee voida yhdistää toisiinsa. Tietoa on voitava siirtää järjestelmästä toiseen ja muokata kohderyhmästä riippuen. Tästä esimerkkinä Neste Jacobs Oy, jonka laitossuunnittelijat käyttävät eri tietokantaa ja instrumenttisuunnittelijat eri tietokantaa. Tiedon on kuljetettava alojen välillä sujuvasti koska ne ovat toisistaan riippuvaisia. Putken koko täytyy tietää, jotta siihen osataan hankkia oikean kokoinen venttiili jne.

Suurella yrityksellä voi olla palkattuna useampi asiantuntija jokaista tietokantaohjelmaa varten. Peruskäytön lisäksi täytyy tietää tietokannan toimintaperiaate. Heidän on osattava suunnitella ja luoda tietokanta, joka toimii oikein. Pystyttävä varmistamaan että tieto tallennetaan oikeassa muodossa, jotta raportit tulostuvat myöhemmin oikein. Sekä heidän on osattava tehdä räätälöityjä lomakkeita, raportteja ja kyselyjä.

## 2 TYÖYMPÄRISTÖ

### 2.1 Neste Jacobs Oy

Työn toimeksiantaja Neste Jacobs Oy, oli alun perin vuonna 1956 perustettu Neste Oyn öljynjalostamon suunnitteluosasto. Suunnitteluosastosta tehtiin oma yrityksensä 1999 ja näin Neste Engineering Oy sai alkunsa. Jacobs Engineering Inc. osti yrityksestä osan vuonna 2004 ja nimi vaihdettiin Neste Jacobs:ksi (lyhennettynä NJ).

NJ on nykyään Pohjoismaiden suurin teknologiapalveluiden, suunnittelun ja projektipalveluiden tuottaja öljyn, kaasun, biopolttoaineiden, petrokemikaalien ja kemikaalien osalta.

NJ:lla on 50 vuoden kokemus laitossuunnittelussa ja suurissa projekteissa Euroopassa, Pohjois- ja Etelä-Amerikassa ja Lähi-idässä. Yrityksestä on 60 % Neste Oilin ja 40 % Jacobs Engineeringin omistuksessa.

Yrityksellä on toimistoja Suomessa, Ruotsissa, Hollannissa, Singaporessa sekä Yhdistyneissä Arabiemiraateissa.

Vuonna 2008 Rintekno Group (Rintekno Oy, Systecon Oy, Turun Sähkösuunnittelu Oy, Kotka Control Oy, Rintekno AB) liitettiin osaksi Neste Jacobsia.

NJ on erikoistunut tarjoamaan kilpailukykyisiä suunnittelu-, hankinta- ja rakennuttamispalveluita läheisessä yhteistyössä asiakkaidensa kanssa.

Organisaatio työllistää tällä hetkellä noin 700 henkilöä ja liikevaihto vuonna 2009 oli noin 100 miljoonaa euroa. Strategiansa mukaisesti NJ hakee kasvua uusilta alueilta, kuten teknologian kehitys ja konsultointi.

NJ:n päätoimisto sijaitsee Porvoossa, Kilpilahden teollisuusalueella, noin 40 km Helsingistä itään. Muut toimipisteet suomessa sijaitsevat Kotkassa, Naantalissa ja Turussa.

(Neste Jacobs in brief)

## 2.2 Työn suorittaminen

Ennen tämän työn aloittamista katsastetaan eräänlainen kulmakivi, josta päästään varsinaiseen opinnäytetyöprojektiin. Projekti liittyy Neste Oilin öljynjalostamon yhden tuotolinjan automaation uusintaan. Siinä saadaan ensi kosketus tietokantoihin ja yhteen työmenetelmiin liittyvään ongelmaan.

Tehtävänä oli kerätä tietoa useasta eri Excel-taulukosta ja kopioida sitä yhteen taulukkoon, ns. Indeksi-taulukko. Indeksi-taulukko on rakennettu niin, että se voidaan suoraan Import-komennolla ajaa tietokantaohjelmaan, tässä tapauksessa ALMA:an. Yksittäiset taulukot oli generoitu vanhasta instrumenttikannasta, yhteen oli koottu venttiilit ja toiseen lähettimet.

Tehtävästä suoriutuu helposti kohtalaisilla ATK-taidoilla. Siinä oli kuitenkin aiheutta kyseenalaistaa työn tehokkuus. Työ on käytännössä tiedon kopioimista yhdestä taulukosta ja liittämistä toiseen. Inhimillisten virheiden riski on olemassa kun taulukkoja muokataan, rivejä piilotetaan ja tietoa kopioidaan. Tätä on instrumenttisuunnittelun arki monen työntekijän osalta.

Varsinaisen opinnäytetyön alettua siirrytään Porvoon jalostamon teknologiakeskukseen ja aloitetaan työt uudessa projektissa. Projekti liittyy Neste Oilin öljynjalostamon kahden prosessiuunin uusimiseen. Työ suoritetaan osana NJ:n instrumentointiosastoa.

Työssä opetellaan käyttämään NJ:lle räätälöityä ALMA NG - automaatio suunnitteluohjelmaa ja tehdään sillä projektiin liittyviä instrumentointitöitä. Tästä pystytään ammentamaan tietoa ja aloittamaan opinnäytetyön kokoaminen.

Tuntityöntekijäksi palkkaaminen takaa sopivat puitteet sekä projektiin liittyvien töiden, että opinnäytetyön kirjoittamisen osalta. Töitä tehdään projektin parissa kolme päivää viikossa ja välipäivinä keskitytään kirjoittamiseen.



### 2.3 Työn tavoitteet

Neste Jacobs Oyn instrumentointiosaston tietokanta-asiantuntija antoi tämän aiheen. Hän on huomannut, kuinka tietojenkäsittely on vuosien mittaan hankaloitunut. Työn tehokkuus kärsii, kun projektien edetessä työntekijät luovat omia Excel-taulukoitaan tai muita dokumentteja, jotka eivät ole minkään yleisen ohjesäännön mukaisia. Instrumentoinnin projekteissa tämä tulee jokaisen työntekijän eteen ennemmin tai myöhemmin. Yhdeksi tekijäksi ovat tulleet tietokantapohjaiset suunnitteluohjelmat, jotka tuovat helpotusta osaan ongelmista, mutta samalla ne tuovat omat vaikeutensa.

Tämän opinnäytetyön tavoitteena on esittää katsaus tietokannoista ja niiden tarkoituksesta. Mihin niitä käytetään prosessiteollisuuden instrumentointi- ja automaatio suunnittelussa. Käydään läpi ALMA-suunnitteluohjelman käyttöä ja mitä sillä voi tehdä. Lopuksi tarkastellaan tietokantojen käytössä ilmeneviä tehokkuuskysymyksiä yrityksen ja työntekijöiden näkökulmasta.

Opinnäytetyö jää Neste Jacobsin käyttöön. Sen avulla voidaan uudet työntekijät tutustuttaa tietokantojen käyttöön instrumentoinnissa ja herättää ajatuksia työtapojen tehokkuudesta.

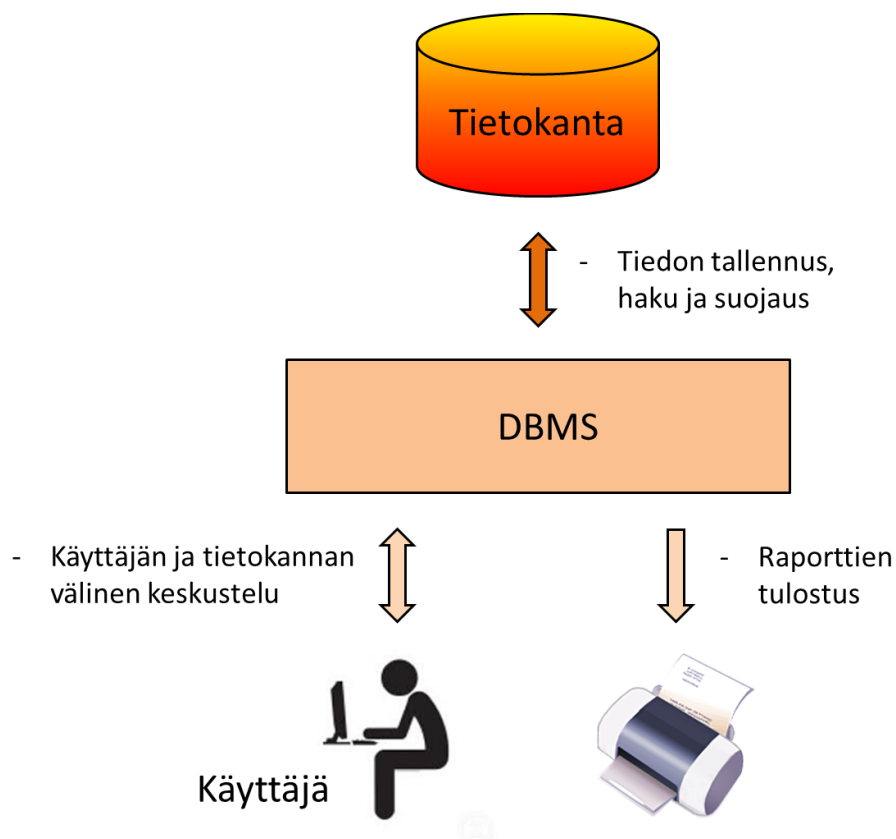
### 3 NYKYAIKAINEN TIETOKANTA

#### 3.1 Tietokanta ja Suunnittelujärjestelmä

Tietokanta (Database, DB) tarkoittaa yleisesti ottaen tietovarastoa, jonka sisältämällä tiedoilla on yhteys toisiinsa. Tiedot on järjestetty johonkin muotoon, josta sitä tulisi olla helppo käsitellä tietokantakielellä (kuten SQL). Esimerkiksi relaatiotietokannassa tieto on järjestetty taulukoihin jotka on luotu vaikka Microsoft Excelillä.

Tietokanta on eräänlainen sähköinen kortisto, jonka tietoja hallinnoi erityinen ohjelmisto, tietokannan hallintajärjestelmä (Database Management System, DBMS). Tunnettuja esimerkkejä ovat Oracle, DB2, Microsoft SQL Server, MySQL ja Access.

Tietokannan hallintajärjestelmät ovat erittäin monimutkaisia ja isoja ohjelmistoja. Ne toimivat rajapintoina käyttäjän ja tietokannan välillä. Tiedot tallennetaan tietokantaan jolloin suorituskyky pysyy hyvänä, tiedot pysyvät ehyinä, muutokset pystytään tekemään joustavasti ja sovellusten muokkaus helpottuu.



KUVA 1. Tietokantajärjestelmän toimintaperiaate

DBMS hallinnoi tietokantaa ja suorittaa tiedon tallentamisen, muokkaamisen ja raportoinnin, kuten kuvassa 1 on esitetty.

Nykyiset tietokannan hallintajärjestelmät ovat valtaosin SQL-pohjaisia relaatiotietokantoja (relational database, RDB). Aiemmin suosiossa olivat verkkomalliset (esim. Total, IDMS) ja hierarkkiset (esim. DL/1) järjestelmät.

NJ:n instrumentointiosaston käyttämä ALMA NG on tietokannan muokkaamiseen tarkoitettu suunnittelujärjestelmä. Se toimii periaatteessa kuin DBMS mutta varsinaisen tietokannan muokkauksen hoitaa siihen tarkoitettu API-sovelluspalvelin (Application Programming Interface).

Suunnittelujärjestelmän avulla tietokannasta saadaan:

- Reaaliaikainen, eli yhden käyttäjän tekemät päivitykset näkyvät heti kaikille muille sen jälkeen, kun ne on ajettu tietokantaan
- Ei-toistuva, tiedot on tallennettu vain kertaalleen
- Ehyt, tiedot ovat ristiriidattomia ja kuvaavat mahdollisimman tarkasti reaalia maailmaa

(Älykkään opastusjärjestelmän...)

(Tietokantojen suunnittelu ja...)

### 3.2 Tietokantojen historiaa

Ensimmäiset tietokoneet käsittelivät tietoa tiedostomuodossa. 1950-luvulla jokaisella tietokoneohjelmalla oli omat tiedostonsa, joita ne käsittelivät. Jos useampi ohjelma käytti samaa tiedostoa, piti tehdä oma kopionsa kullekin ohjelmalle. Kun huomattiin, että tiedostoja olisi kätevä jakaa muiden käyttäjien kesken, alettiin kehittää tietokanta-teknologiaa.

Tietokantojen hallintajärjestelmien ensimmäiset versiot syntyivät 1950-luvun jälkipuolella, kun tietokoneet yleistyivät liike-elämässä. Nämä olivat General Electric Companyn Mark I Report Generator vuonna 1956 ja IBM:n Information Retrieval vuonna 1958.

Tietojenkäsittely ja tietokantojen käyttö yleistyi liike-elämässä 1960-luvulla. Magneettinauhat korvasivat 50-luvulla käytetyt reikäkortit ja nauhakelat. Suurimman harppauksen eteenpäin tarjosi käynnissä ollut Yhdysvaltain ja Neuvostoliiton välinen avaruuskilpa. Apollo-ohjelman rahoitus oli niin yltäkylläinen, että IBM otettiin mukaan Apollo-projektin tietojärjestelmän kehitykseen vuonna 1965. Tämän tuloksena IBM kehitti järjestelmän nimeltä IMS. Siinä toteutettiin ensimmäisen kerran periaate, jossa ohjelma-koodi pidettiin erillään käsiteltävästä datasta.

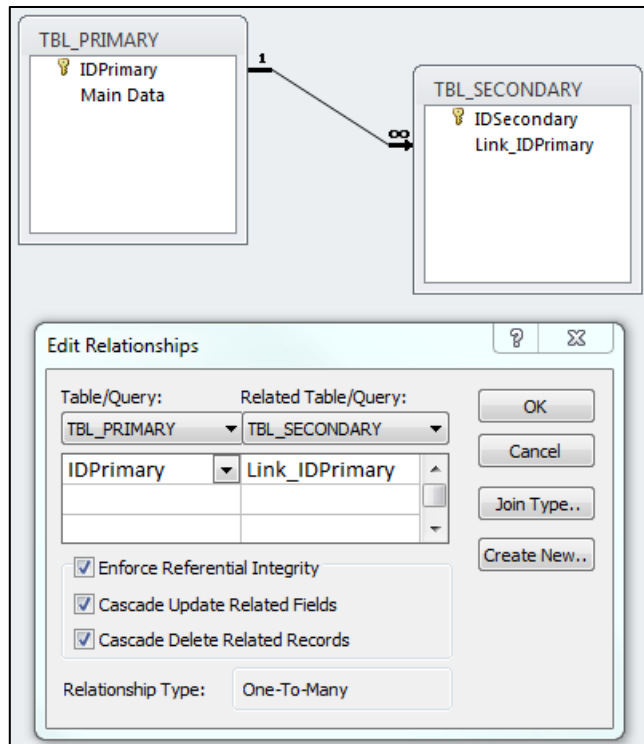
Nykyaikaisten tietokantojen näkökulmasta tärkein julkaisu oli IBM:llä työskentelevän matemaatikon, Edgar F. Coddin kirjoitus relaatiotietokannoista vuonna 1970. IBM kehitti relaatiomallin prototyypin, System R:n, jonka ensimmäinen versio julkaistiin 1978–1979. System R:n kyselykieli oli Structured Query Language, joka vakiintui muutamassa vuodessa standardiksi. IBM ei kuitenkaan huomannut kielen potentiaalia. Vuonna 1977 Larry Ellison perusti Oraclen ja alkoi kehittää ja myydä SQL-yhteensopivia tuotteita.

Vuonna 1980 IBM toi markkinoille ensimmäisen kaupallisen tietokantatuotteensa, SQL/DS (Structured Query Language/Data System). Samana vuonna se toi ensimmäiset PC:t markkinoille. Relaatiotietokannat olivat tulleet jäädäkseen ja vuonna 1985 julkaistiin ensimmäinen SQL-standardi. Relaatiomallin voittokulku on jatkunut 1980-luvulta tähän päivään.

(Tietokantojen historia ennen SQL:ää)

### 3.3 Relaatiopohjaisen tietokannan perusteet

Relaatiomallisessa tietokannassa tieto on jaettu useisiin tauluihin. Taulujen väliset relaatiot luodaan kuten kuvan 2 yksinkertaistetussa Access-esimerkissä.



KUVA 2. Access-relaatiomalli

Ylimpänä on kuvattu taulujen väli-  
nen yksi-moneen relaatio.

TBL\_PRIMARY on päätaulukko  
ja IDPrimary pääavain.

TBL\_SECONDARY liittyy pää-  
taulukkoon Link\_IDPrimary-  
viiteavaimen avulla.

Kuvan 2 alemmassa ikkunassa on esitetty, miten taulut ovat suhteessa toisiinsa. Esimerkiksi **Enforce Referential Integrity** tarkoittaa, että jos taulut viittaavat puuttuviin tietoihin niin ohjelma ilmoittaa siitä. Se siis varmistaa näin, ettei tietokanta pirstoudu.

Nykyaikaiset relaatiomalliset suunnittelujärjestelmät tarjoavat ainakin seuraavat toiminnot:

- Luo tietokannan, joka koostuu kentistä, taulukoista ja taulukoiden välisistä relaatioista
- Helposti tehtävä tiedon tallentaminen, muokkaaminen ja poistaminen
- Sisäinen kyselykieli, jonka avulla voit tehdä kyselyjä kannan tiedoista
- Raporttgeneraattori, jolla voi luoda asiallisen näköisiä raportteja tiedoista
- Turvaa tietokannan tiedot suojauksen, hallinnan ja palautuksien avulla

Relaatiotietokannat ovat helpompia käyttää ja muuttaa kuin perinteisemmät hierarkkiset ja verkkomalliset tietokannat. Suurikin määrä tietoa saadaan yksinkertaisilla kyselyillä. Tietokannan muutokset sujuvat vaivattomasti ilman monimutkaisia operaatioita. Lisäksi SQL on standardoituun lähes ainoaksi tietokantakieleksi.

(AB+ Trainer – Tietokannat Access)

(Relaatiotietokantojen peruskäsitteet)

Relaatiomallin tutkiminen voidaan jakaa kolmeen osaan:

- Tiedon rakenne
- Tiedon käsittely
- Tiedon eheys

### 3.3.1 Tiedon rakenne

Tietokannan rakenne on tehtävä sellaiseksi, että suunnittelujärjestelmä pystyy sitä tehokkaasti käsittelemään. Relaatiomallissa tiedot esitetään kaksiulotteisina tauluina, jotka koostuvat kentistä ja tietueista. Taululla on aina otsikkorivi ja sen alla ovat tietorivit. Yhdellä rivillä voi olla useita kenttiä. Kuvassa 3 on esimerkkitaulu.

ASIAKAS			
asiakasno	etunimi	maa	postinro
1126	Tellervo	SUOMI	48400
1175	Hjalmar	RUOTSI	830 13
2354	Peter	YHDYSVALLAT	99840

Relaatio

Tietue/Rivi

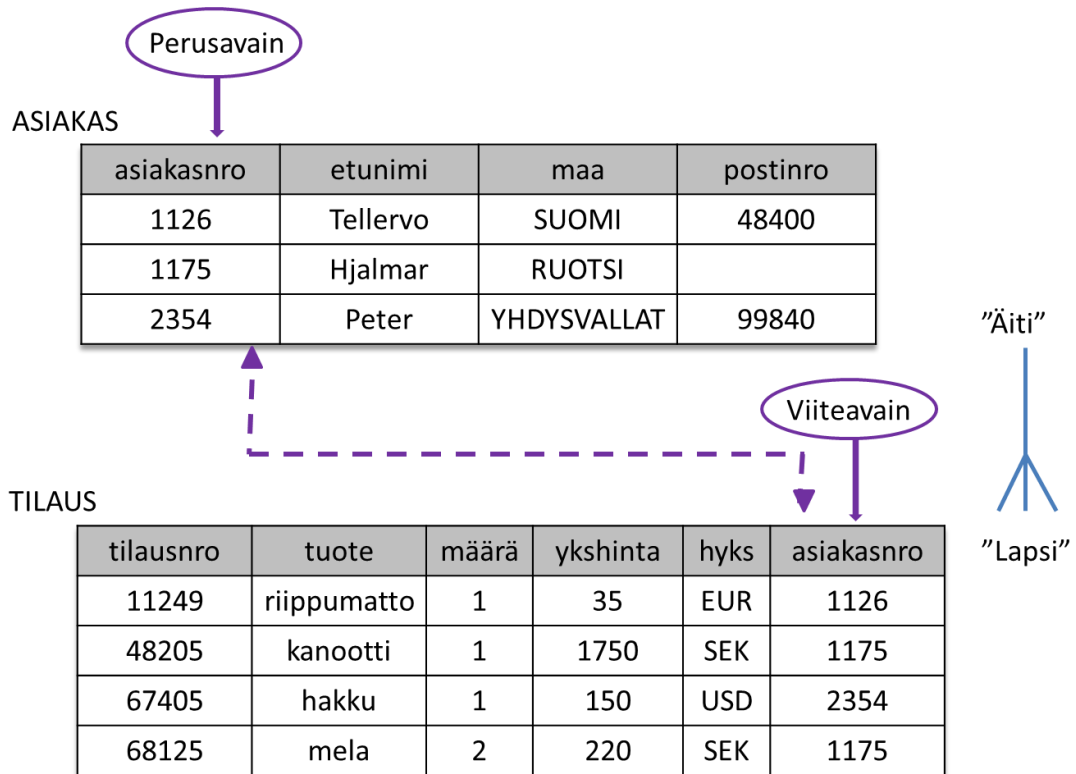
KUVA 3. Esimerkki taulusta

Otsikkorivi määrää sarakkeen nimen. Sarakkeen tietojen arvot kuuluvat samaan arvojoukkoon. Niillä on siis yhteinen tietotyyppi, kuten numeerinen tai merkkimuotoinen ja niille on varattu tietty pituus.

(AB+ Trainer – Tietokannat Access)

(Relaatiotietokantojen peruskäsitteet)

Relaatiomallin mukaan kussakin taulussa on tunnisteena perusavain. Kuvien 3 ja 4 ASIAKAS-taulun perusavain on sarake **asiakasno**. Tällä tarkoitetaan saraketta jonka on oltava yksilöivä eli uniikki. Tämä tarkoittaa, että sarakkeessa ei saa olla kahdella tai useammalla eri rivillä samaa arvoa. Perusavaimen tiedon avulla tehostetaan tiedon hakua ja varmistetaan tiedon yksikäsitteisyys.



KUVA 4. Kahden taulun välinen relaatio

TILAUS-tauluun on tallennettu asiakkaiden tekemät tilaukset. Taulu liittyy viiteavaimen asiakasno avulla ASIAKAS-tauluun.

ASIAKAS-taulu on ns. **Äiti** ja TILAUS on **Lapsi**. Tämä on yksi-moneen relaatio, äidillä voi olla monta lasta mutta lapsella vain yksi äiti. TILAUS-taulun lisäksi voisi olla olemassa toinen lapsitaulu. Tämä voisi olla esim. MAAT-taulu, jonka avulla saataisiin selville, montako tilaajaa mistäkin maasta yrityksellä on.

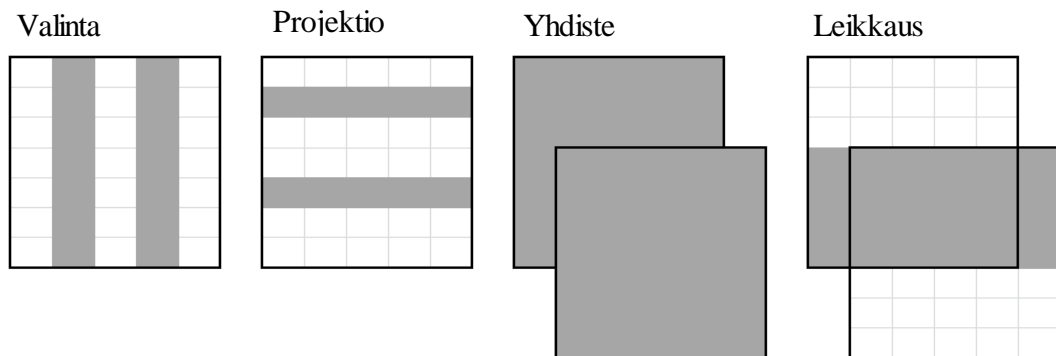
TILAUS-taulusta nähdään, että asiakasno 1175 on tehnyt kaksi eri tilausta, jotka molemmat tallentuvat oman yksilöivän tunnuksen **tilausno** mukaan.

(AB+ Trainer – Tietokannat Access)

(Relaatiotietokantojen peruskäsitteet)

### 3.3.2 Tiedon käsittely

Relaatiotietokannassa olevaa tietoa pitää pystyä hakemaan ja muokkaamaan. Tämä tapahtuu joukko-opin mukaisilla operaatioilla. Taulu on periaatteessa joukko rivejä ja joukkoon voidaan kohdistaa operaatioita, kuten ”hae kaikkien suomalaisten asiakkaiden tekemät tilaukset (valinta), niistä kaikki tilatut riippumatot (projektio)”. Operaatioita on kahdeksan kappaletta, kuvassa 5 esitetään niistä neljä.



KUVA 5. Joukko-opin perusoperaatioita

Alla on lyhyet selitykset kustakin perusoperaatiosta:

- Valinta Valitsee rivejä taulusta
- Projektio Valitsee sarakkeet taulusta
- Yhdiste Muodostaa relaation, joka sisältää kummankin taulun tai jompaankumpaan tauluun kuuluvat rivit
- Leikkaus Muodostaa relaation, johon kuuluvat kumpaankin tauluun kuuluvat rivit

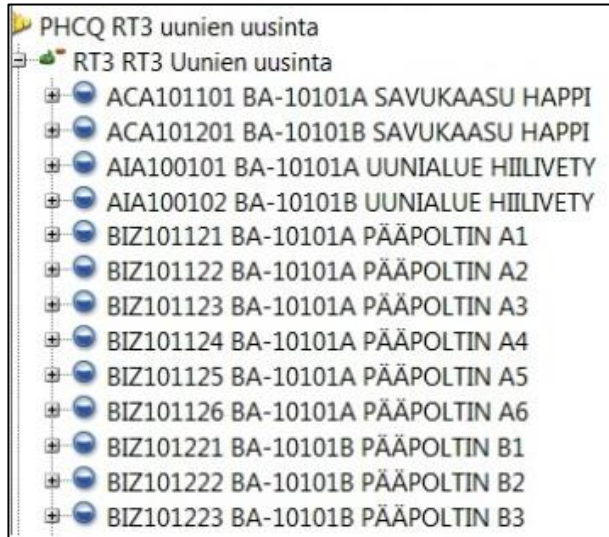
Lisäksi on vielä Erotus, Ristitulo, Liitos ja Jako.

Joukko-operaatioilla käsitellään yksi tai useampi taulu. Operaatiot toteutetaan SQL-kielellä.

(Lähiverkkopohjaisten relaatio...)



Seuraavaksi esimerkki siitä, kuinka operaatiot toimivat ALMA-ympäristössä. Tämä esimerkki on kuvitteellinen, koska itse tietokantaa ei pääse tarkastelemaan. Käytännössä se kuitenkin vastaa todellisuutta. Käyttöliittymässä tämä tapahtuu haku-toiminnoilla.



Otetaan tarkasteluun kuvassa 6 näkyvä automaatiopositioden joukko. Niistä halutaan valita vain ”BIZ1011”-alkuiset positiot. Käytännössä tämä menee niin, että haku-toimintoon laitetaan ehdoksi BIZ1011\* ja kohdistetaan haku koko joukkioon.

KUVA 6. Automaatiopositioden joukko

Automaatiopositioden joukko on olemassa taulukkona, jossa automaatiopositio on yksilöivä tunnus. Joukko-opin mukainen valinta-operaatio valitsee sarakkeen ja projektiio-operaatio valitsee rivit, joissa ovat ”BIZ1011”-alkuiset positiot (kuva 7).

Valinta	
automaatiopositio	nimi
ACA101101	BA-10101A SAVUKAASU HAPPI
ACA101201	BA-10101B SAVUKAASU HAPPI
AIA100101	BA-10101A UUNIALUE HIILIVETY
AIA100102	BA-10101B UUNIALUE HIILIVETY
BIZ101121	BA-10101A PÄÄPOLTIN A1
BIZ101122	BA-10101A PÄÄPOLTIN A2
BIZ101123	BA-10101A PÄÄPOLTIN A3
BIZ101124	BA-10101A PÄÄPOLTIN A4
BIZ101125	BA-10101A PÄÄPOLTIN A5
BIZ101126	BA-10101A PÄÄPOLTIN A6
BIZ101221	BA-10101B PÄÄPOLTIN B1
BIZ101222	BA-10101B PÄÄPOLTIN B2
BIZ101223	BA-10101B PÄÄPOLTIN B3

Projektiot

KUVA 7. Valinnan ja projektion yhdistelmä

Tuloksena haku antaa taulun, joka sisältää vain ”BIZ1011”-alkuiset positiot.

### 3.3.3 Tiedon eheys

Relaatiomallissa tiedon eheydellä tarkoitetaan sitä, ovatko tiedot todenmukaisia ja yksikäsitteisiä. Tarkoitus on varmistaa, ettei tieto mene ristiin tai tuhoudu, eikä tietokantaan jää kummittelemaan orpoja tietoja. Eheys vaarantuu, jos esim. sama asiakas tallennetaan kahteen kertaan tai asiakkaasta löytyy kaksi eri osoitetta eikä tiedetä, kumpi on oikein. Mallissa on kaksi yleistä eheyssääntöä, avaineheys ja viite-eheys.

Avaineheys tarkoittaa, että taulun perusavaimen osana olevan sarakkeen arvo ei saa olla tyhjä (NULL). Toisin sanoen perusavaimen arvo on pakollinen.

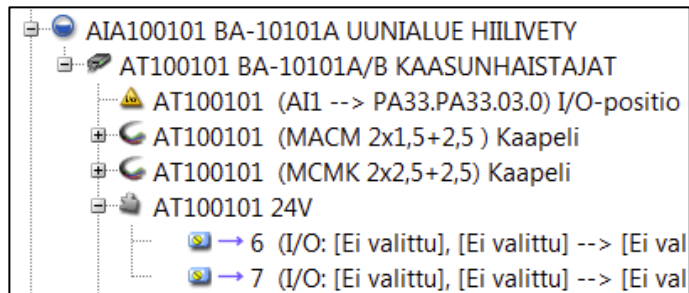
Viite-eheys tarkoittaa, ettei tietokannassa saa olla tietoja jotka eivät viittaa mihinkään eli ”orpoja” tietoja. Jos esim. tietokannasta poistetaan asiakas, jolla on vielä maksamattomia laskuja, jäävät laskut roikkumaan ja viite-eheys särkyä.

Tiedon eheyden vaatimukset kannattaa ottaa huomioon jo tietokannan suunnitteluvaiheessa. Täytyy tarkoin määritellä esimerkiksi mitä nimikkeitä käytetään millekin tuotteelle, eivätkä nämä saa mennä sekaisin. Useimmat relaatiokannat mahdollistavat nykyisin viite-eheyden valvomisen määrittämällä kantaan eheysrajoitteet, joita ei voida ohittaa.

(Lähiverkkopohjaisten relaatio...)

### 3.4 Oliot ja ALMA

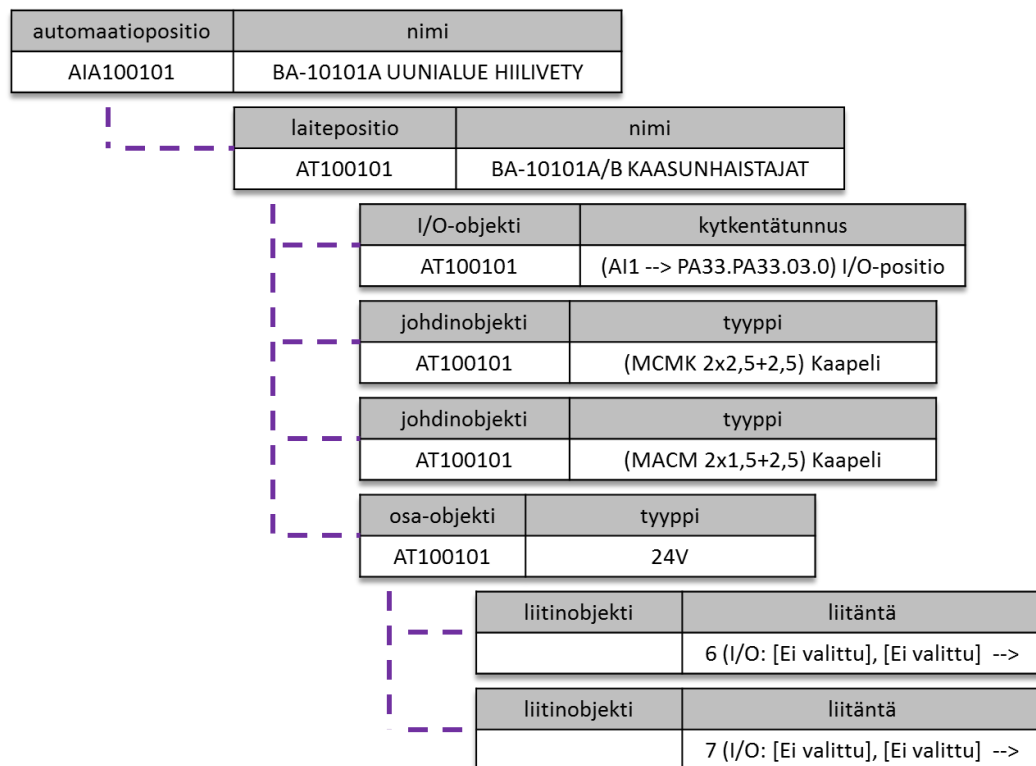
ALMA on Olio-orientoitunut relaatiotietokanta. Tämä tarkoittaa, että sen tietoja käsitellään olio-objekteina. Käyttöliittymän osalta se ilmenee siten, että mallinnettu reaali maailma nähdään myös olio-objekteina.



KUVA 8. Oliot

Kuvassa 8 on esillä ALMA:n olio-objekteja. Ylimpänä on automaatiopositio-objekti, sen alla laitepositio-objekti, sitten I/O-, johdin-, osa- ja liitinobjektit. Näihin jokaiseen objektiin on sidottu niiden omat attribuutit eli tiedot.

Kuva 8 voidaan esittää tauluina, jotta asiasta saa paremman käsityksen.



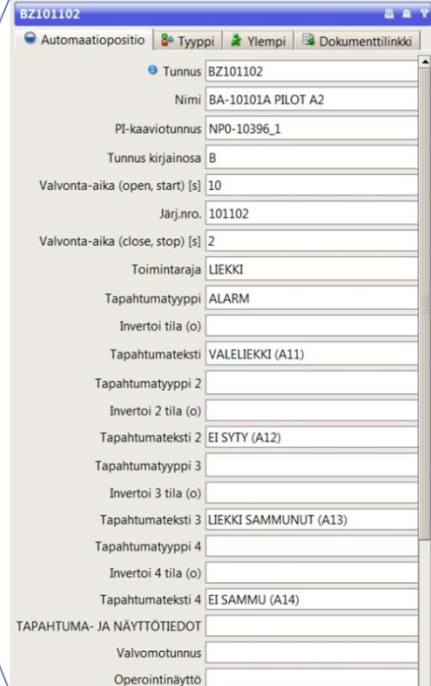
KUVA 9. Olio-käyttöliittymän taulukkoesitys

Automaatiopositio on tässä tapauksessa äiti ja siihen on sitten sidottuna useampi lapsi. Laitteen osiksi katsotut objektit, kuten sen johtimet ja liitokset, tulevat saman laitepositio-objektin alle.

Objekteilla on sama positiotunnus, koska ohjelma automaattisesti nimeää laitepositioiden alla olevat objektit samalla tunnuksella. Tunnus generoidaan automaattisesti automaatioposition mukaan. Kahta samaa automaatiopositiota ei voi olla, tämä on yksi monista keinoista varmistaa tietokannan eheys.

Attribuutit ovat liitoksissa vain tiettyyn objektiin. Kuten kuvassa 10 on havainnollistettu. Automaatiopositio-objektiin on sidottu siihen liittyvät tiedot ja ne ovat nähtävissä attribuutteina. Kuvassa 10 on BZ101102 PILOT-polttimen automaatioposition sidotut attribuutit.

automaatiopositio	nimi
BZ101102	BA-10101A PILOT A2



The screenshot shows a software window titled 'BZ101102' with a list of attributes for an 'Automaatiopositio'. The attributes include: Tunnus (BZ101102), Nimi (BA-10101A PILOT A2), PI-kaaviotunnus (NP0-10396\_1), Tunnus kirjainosa (B), Valvonta-aika (open, start) [s] (10), Järj.nro. (101102), Valvonta-aika (close, stop) [s] (2), Toimintaraja (LIEKKI), Tapahtumatyyppi (ALARM), Invertoi tila (o), Tapahtumateksti (VALELIEKKI (A11)), Tapahtumatyyppi 2, Invertoi 2 tila (o), Tapahtumateksti 2 (EI SYTY (A12)), Tapahtumatyyppi 3, Invertoi 3 tila (o), Tapahtumateksti 3 (LIEKKI SAMMUNUT (A13)), Tapahtumatyyppi 4, Invertoi 4 tila (o), Tapahtumateksti 4 (EI SAMMU (A14)), TAPAHTUMA- JA NÄYTTÖTIEDOT, Valvomotunnus, and Operointinäyttö.

**KUVA 10. Automaatioposition attribuutit**

Attribuutit ovat siis sidottu objektiin ja objektit ovat vuorovaikutuksessa keskenään. Tätä on Olio-orientoitunut relaatiotietokanta.

(Neste Jacobs Oy, asiantuntija)

### 3.5 SQL

Structured Query Language on tiedonhallintakieli, jota käytetään relaatiotietokantojen kyselyissä, päivityksessä ja hallinnassa. Sen avulla voidaan tietokantaan tehdä erilaisia hakuja, muutoksia ja lisäyksiä. Kielen ensimmäinen versio kehitettiin vuonna 1974 ja uusin 2011.

SQL ei ole sovellusten kehittämiseen tarkoitettu ohjelmointikieli, vaativampaan sovellukseen käytetään isäntäkielenä yleensä jotain kehittyntä ohjelmointikieltä, johon SQL upotetaan. SQL on ns. ei-proseduraalinen kieli. Tämä tarkoittaa, että sillä kerrotaan mitä tietoa haetaan, eikä sitä miten haetaan.

Seuraavaksi esimerkki SQL:n käskyistä. Tiedon käsittelyyn käytettävistä käskyistä tärkeimmät ovat:

**SELECT** haetaan valittu osajoukko yhdestä tai useammasta taulusta

**SELECT tilausnro FROM TILAUS WHERE kpl > 1**

Komennolla valitaan **TILAUS**-taulusta ne rivit, joissa **tilausnro** täsmää **where**-osassa annettuun ehtoon.

**UPDATE** käskyllä voidaan päivittää olemassa olevaa tietoa

**INSERT** lisää uusia tietoja tauluun

**DELETE** ylimääräisen tiedon poistaminen

Lisäksi on vielä tiedon määrittelykäskyt joilla voidaan luoda uusi taulu, muokata sen rakennetta tai poistaa taulu. On olemassa myös valvontakomennot, joilla voidaan hallita käyttäjien oikeuksia taulujen käyttöön.

SQL on edelleen suosituin tiedonhallintakieli vaikka se ei ole täydellinen. Lähes kaikki järjestelmän valmistajat tekevät kieleen laajennuksia. Eri valmistajien SQL-yhteensopivuus pätee siis vain perusteiden osalta ja sovellusten siirrettävyys kärsii tämän takia.

(Lähiverkkopohjaisten relaatio...)  
(SQL, Structured Query Language)

## 4 ALMA

### 4.1 Perustiedot

ALMA on nimitykseltään ns. tietämyshallintajärjestelmä teknisen tiedon ja tapahtumien linkkaarenaikaiseen hallintaan. Se sisältää suuret määrät toimintoja ja sitä voidaan käyttää teknisen suunnittelun monilla alueilla. Näitä ovat mm. työtapojen yhtenäistäminen, standardointi, suunnittelu, projektien ja dokumenttien hallinta sekä kunnossapito, reaaliaikaisesti.

Ohjelmiston kehittäjä on suomalainen Alma Consulting Oy. Yritys toimittaa ja ylläpitää järjestelmän ja siihen liittyvät palvelut. Kuten koulutuksen, käyttöönoton, tuen projekti-palvelut, pääkäyttäjäpalvelun ja tietokantahotellipalvelun.

ALMA:n käyttäjälisenssejä on prosessiteollisuudessa noin 7000 noin 300 yrityksessä ja 46 maassa. Se on ollut markkinoilla vuodesta 1986.

Neste Jacobs siirtyi vuonna 2009 käyttämään nykyaikaisille laitteille kehitettyä ALMA NG -järjestelmää. Tätä ennen käytössä oli vanhempi versio joka ainakin MS Windows -käyttöliittymältään näyttää olevan 1990-luvulta peräisin.

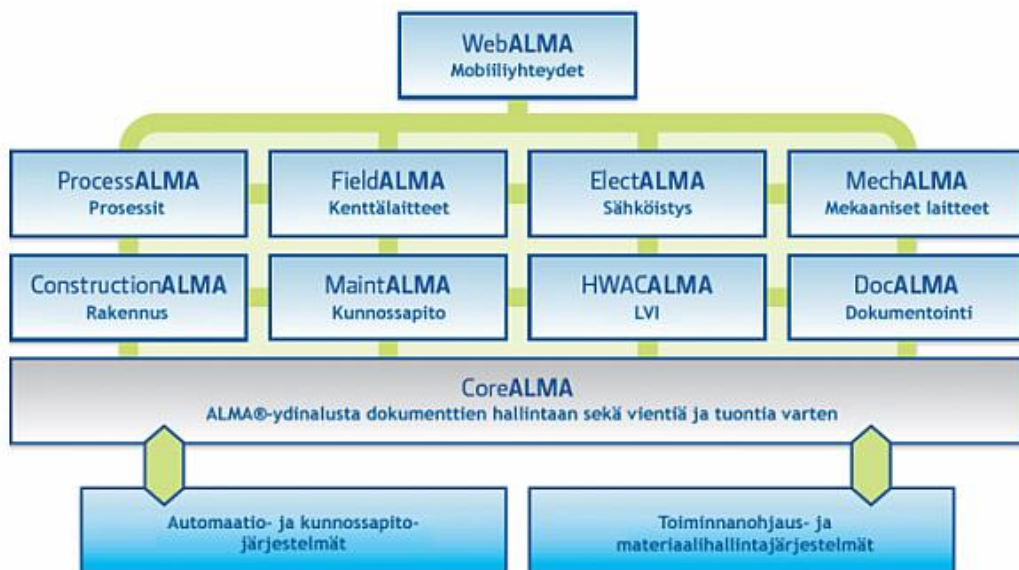
(ALMA ja Neste yhteistyössä...)  
(ALMA Tuoteratkaisut)

## 4.2 Katsaus tuotepiheeseen

ALMA-tuotepihe käsittää kokonaisen rykelmän prosessitekniikan eri osa-alueille tarkoitettuja järjestelmiä. Näiden avulla on tarkoitus luoda ja ylläpitää tehtaan tai laitoksen koko elinkaarenaikaista tiedonhallintaa ja dokumentaatiota. Kukin moduuli voi toimia yksin tai niistä kootaan asiakkaalle sopiva kokonaisuus. Alla on selitetty tämän opin- näytetyön kannalta olennaiset järjestelmät.

**CoreALMA** ALMA -ohjelmistojen ydin. Se toimii viestintärajapintana tietojen vientiin ja tuontiin muiden järjestelmien välillä. Kaikki tuotepiheen muut osat liittyvät tietokantaan CoreALMA:n kautta.

**FieldALMA** Kattaa laitoksen kaikki piirit, laitteet, kytkentätilat ja kentän tiedot. Tämä ohjelmisto on Neste Jacobs Oy:n instrumentoin- tiosaston käytössä.



KUVA 11. ALMA-tuotepihe (<http://www.alma.fi/Suomeksi/Tuoteratkaisut>)

Kuvasta 11 näkyy kaikki tekniikan osa-alueet joihin ALMA:a voidaan käyttää.

**ProcessALMA** Hallitsee prosessituotetehtaan prosessi- ja laitossuunnittelun mitoitustiedot sekä dokumentoinnin. Käsittää virtaus- ja PI-kaaviot, laite- ja instrumenttiluettelot, putkiluokat, virtaus- tekniset mitoitukset ja toimintaprosessikuvaukset.

<b>ElectALMA</b>	Sähköistyksen ja sähkön jakelun suunnittelu. Kattaa laitokset kaikki keskukset, sähkölähdöt, kytkentätilat ja muun sähköistystiedon.
<b>MechALMA</b>	Kokoaa kaikki koneiden ja laitteiden tuotetiedot. Mitoitustiedot, varaosatie-dot, rakennepiirustukset, asennuskuvat, manuaalit, kone- ja laiterekisterit ja viranomaisraportit.
<b>ConstructionALMA</b>	Teollisuuskiinteistön tiedonhallintaohjelmisto. Moderni huoltokirja, joka tukee kiinteistön käyttöä ja kunnossapitoa.
<b>MaintALMA</b>	Sisältää tarvittavat toiminnot teollisuuden kunnossapidon tai huoltoliiketoiminnan johtamiseen ja ohjaukseen. Kustannus-seuranta, kalenterit, seisokkihuoltojen resursointi ja muuta.
<b>HWACALMA</b>	Toimii taloteknisen tiedon kokoavana järjestelmänä, jolla voidaan hallita myös erillisjärjestelmien tieto ja niiden kunnossapito. Keskuslämmitys, vedenjakelu, ilmanvaihto yms.
<b>DocALMA</b>	Dokumenttien hallintaohjelmisto. Työväline suurtenkin tietomäärien hallintaan, tehostaa tiedonhakua. Mahdollistaa dokumenttien muutoshistorian tarkastelemisen.
<b>WebALMA</b>	ALMA:n yhteyteen rakennettu www-palvelinohjelmisto. Mahdollistaa tietokantojen katselun reaaliajassa internetse-laimen kautta, joko tietokoneella tai mobiililaitteella.

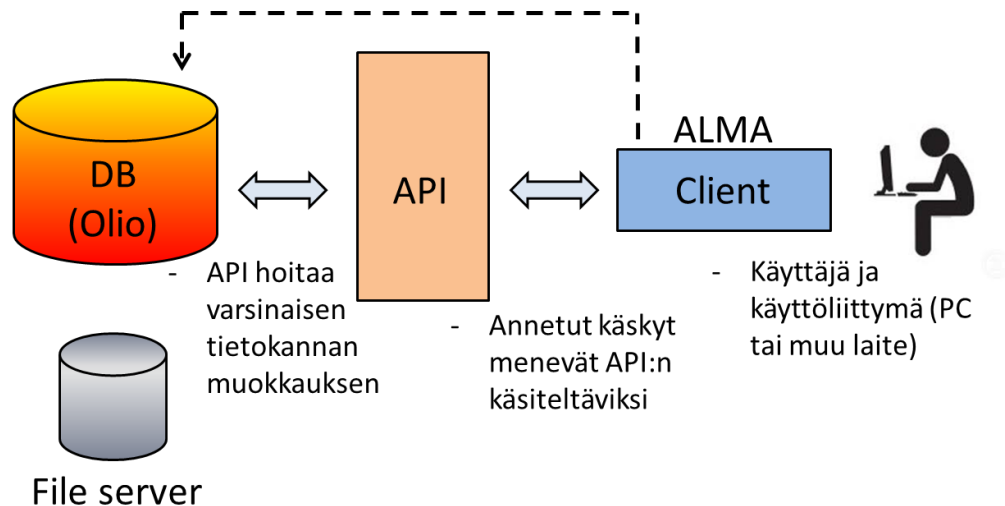
(ALMA Tuoteratkaisut)



### 4.3 Tietokannan malli

ALMA:n tietokantamalli vastaa samaa, kuin kohdassa 3.1 (Tietokanta ja...) kuvassa 1 on havainnollistettu. Koska kuva 1 on pelkistetty esimerkki, niin se ei ALMA:n tapauksessa kerro koko totuutta.

Todellisuudessa tietokanta toimii suunnilleen näin:



KUVA 12. ALMA-tietokantajärjestelmä

Käyttäjä tekee muutokset instrumenttipiireihin PC-käyttöliittymän välityksellä. Muutos voi tarkoittaa uuden laitteen lisäämistä, kytkentöjen muutoksia, laiteposition muuttamista tai vastaavaa.

Käyttäjän koneella oleva ALMA-client välittää muutoskäskyt SQL-koodina API:lle.

Tehokäyttäjä tai asiantuntija voi myös muokata tietokantaa suoraan PC:ltä.

API käsittelee käskyt ja hoitaa varsinaisen tietokannan muokkauksen. Se pitää huolen, että tieto tallentuu oikein ja pysyy tallessa. Tällöin inhimillisiä virheitä ei pääse tapahtumaan.

API välittää muutokset ja tietokannan tiedot takaisin ALMA-clientille, josta käyttäjä voi nähdä ne.

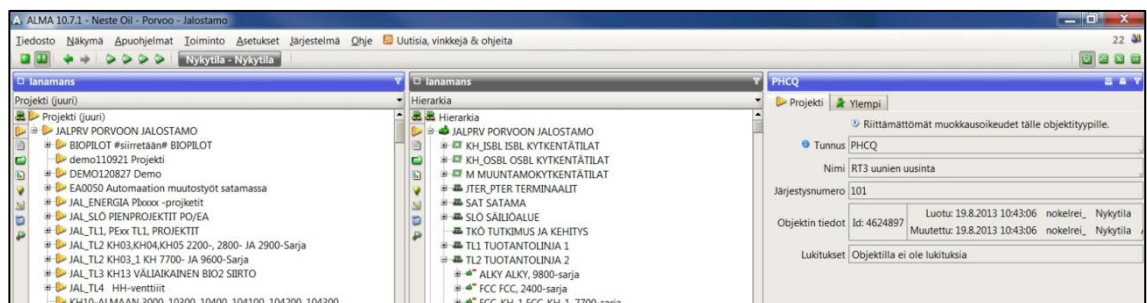
Tiedostopalvelin (File Server) hoitaa tietokantaan liittyvät tiedostokytkennät kuten PDF-dokumentit ja AutoCAD-piirustukset.

## 4.4 Käyttöliittymä

ALMA-järjestelmän tavoitteena on projektin tai laitoksen kaikkien instrumenttien suunnittelun hallinta yhdestä käyttöliittymästä käsin. Kaikki asiat, olivat ne sitten laitteet, projektitunnukset, kaapelit tai kotelot käsitellään objekteina. Jokainen muutos ja muokaus tietokantaan tapahtuvat reaaliajassa. Eli kun käyttäjä tekee muutoksen, se näkyy toisille käyttäjille heti.

Projektille luodaan automaatiopositiot joille annetaan laitteet, liittimet ja kaapelit. Kaapelit kytketään ja johdot vedetään koteloihin, ristikytkentään ja aina logiikkakorteille asti. Kytkentää tai muitakaan muutoksia ei tehdä yksi kerrallaan, vaan tarkoitus on hoitaa useampi samanaikaisesti. Järjestelmästä voidaan sitten generoida AutoCAD-kuvat valmiiden pohjien kanssa.





Kun ALMA:n käynnistää, avautuu ensiksi käyttöliittymä. Näkyviin tulee kolme ikkuna: kaksi tikapuuperiaatteella järjestettyä näkymäikkunaa ja yksi attribuutti-ikkuna. Ohjelman harmaasta ylävalikosta ei löydy paljoa asetuksia. Sieltä tärkeimmät ovat lähinnä kielen valinta ja ohjeet. Kuvassa 13 on käyttöliittymä avattuna, se on tarkemmin nähtävissä liitteessä 1.



KUVA 13. Käyttöliittymä

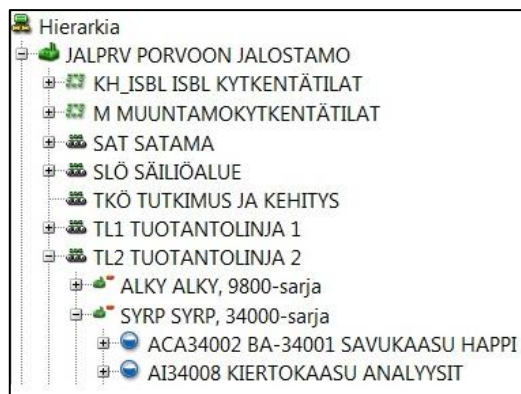
#### 4.4.1 Käyttöliittymän näkymät

Näkymäikkunoissa tehdään objektien muokkaus. Ikkunat ovat vierekkäin ja objekteja voi raahata niiden välillä jos halutaan siirtää tai kopioida niitä. Ikkunat saadaan myös synkronoitua, jolloin ne näyttävät samalta. Välillä voisi olla hyödyllistä, jos näkymäikkunoita saisi useamman kuin kaksi auki samanaikaisesti. Ikkunoihin saa auki erilaisia näkymiä. Liitteessä 1 vasemmanpuoleisessa ikkunassa on Projekti-näkymä ja oikeassa on Hierarkia-näkymä. Käydään seuraavaksi eri näkymät läpi.

	Hierarkia
	Projekti
	Dokumentti
	Mappi
	Raportti
	Ratkaisu
	Suunnitelutyyppe
	Tuotelaji
	Haku

KUVA 14. Näkymäikkunoiden nimitykset

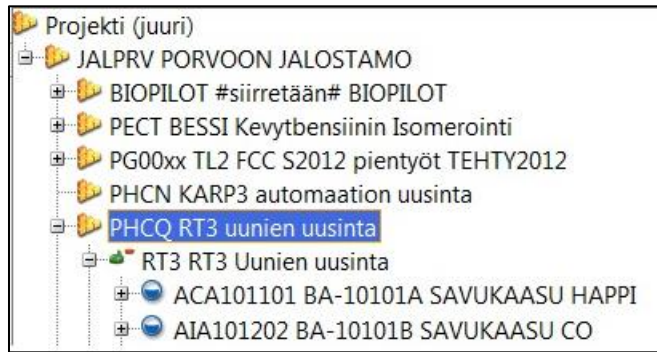
#### Hierarkia



KUVA 15. Hierarkia-näkymä

Hierarkia-näkymässä ovat laitoksen valmiit, jo käytössä olevat osiot. Nämä osat ovat valmistuneet projekteista ja ne on ajettu laitoksen valmiiden järjestelmien joukkoon. Siitä on laajennettu TL2 ja SYRP 34000-sarja, jonka alla näkyy kaksi automatiopositiota. Näkymän osia ei saa muokata, ellei siihen ole pätevää syytä. Mallia saa kuitenkin ottaa.

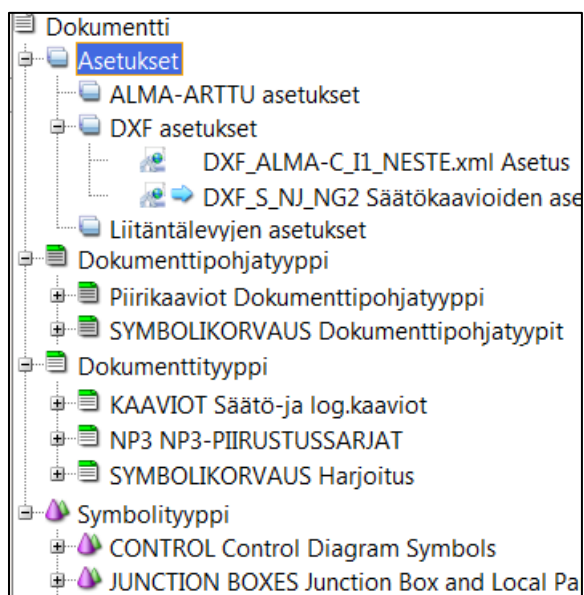
## Projekti



KUVA 16. Projekti-näkymä

Nämä ovat keskeneräiset projektit. Tähän kohtaan voidaan aloittaa uusi projekti ja sille aletaan lisätä positioita. Tuotteistaminen, johdotus ja muut tehdään kaikki tässä.

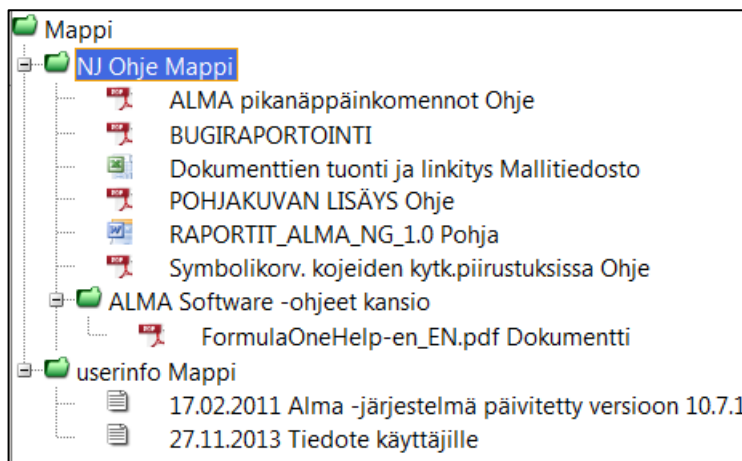
## Dokumentti



KUVA 17. Dokumentti-näkymä

Dokumenteista löytyvät kaikki tietokannan yhteydessä olevan tiedostopalvelimen sisältämät asiakirjat. Siellä on web-palvelun xml-asetukset. Dokumenttipohjatyyppi, jota käytetään symbolikorvauksessa. Dokumenttityypeissä ovat ohjeet, todistukset, AutoCAD-piirustussarjat, laiteluettelot, kojeluettelot, yritysten tuotodokumentit jne. Symbolityypeissä ovat symbolikorvauksen kannalta olennaiset AutoCAD-blokkit.

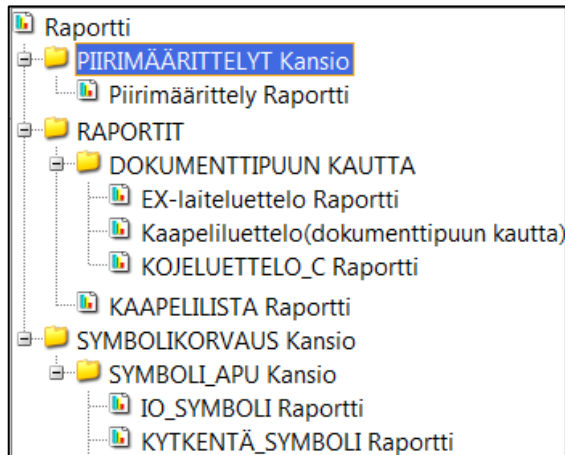
## Mappi



KUVA 18. Mappi-näkymä

Tästä kohdasta löytyvät vielä erikseen ALMA:n käyttäjiä mahdollisesti kiinnostavat dokumentit. Samat tiedostot löytyvät Dokumentti-näkymästä mutta täältä ne löytyvät selkeämmin.

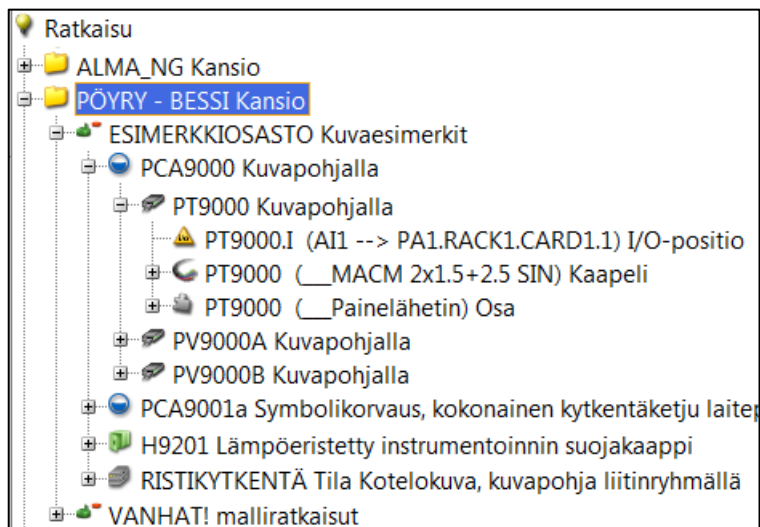
## Raportti



KUVA 19. Raportti-näkymä

Tästä löytyvät erilaisten raporttien luomiseen käytettävät pohjat. Ne ovat jonkin kolmannen osapuolen luomia Java-koodeja. Näillä saadaan luotua laitesarjojen erilaiset luettelot, kuten kojeluettelot.

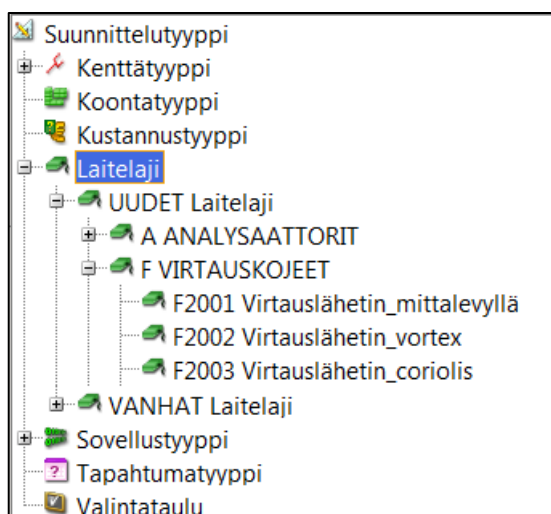
## Ratkaisu



KUVA 20. Ratkaisu-näkymä

Kohdasta löytyvät esimerkit. Pääkäyttaja tai asiantuntija lisää tänne mallit joista peruskäyttaja voi saada vihjeitä tai suoria ratkaisuja. Tästä on hyötyä uusille työntekijöille tai jos projektiin lisätään jokin ennen käyttämätön tyyppi.

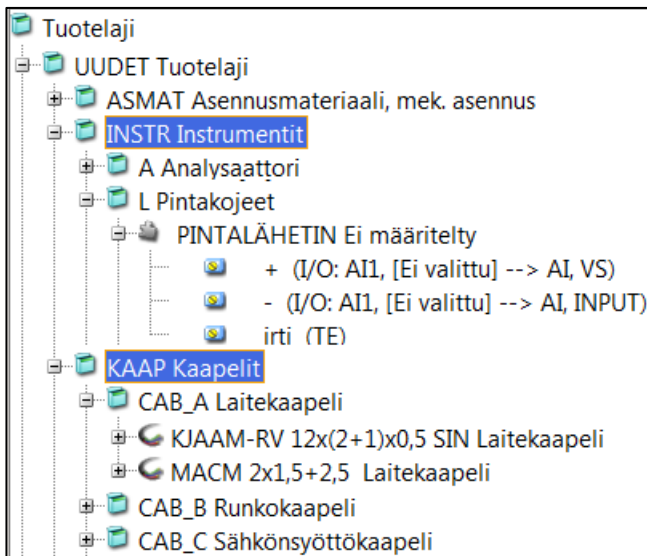
## Suunnittelutyyppi ja laitelajit



KUVA 21. Suunnittelutyyppi-näkymä

Täältä löytyvät kaikki objektityypit. Instrumentoinnin kannalta tärkeimpiä ovat laitelaji ja sovellustyyppi. Tyyppi määrää mitä attribuutteja kyseiseen objektiin liitetään. Koska eri laitteisiin sidotaan eri tietoja, niihin tulee myös eri attribuuttipohja. Laitteilla nämä näkyvät sitten tulostetussa kojeluettelossa.

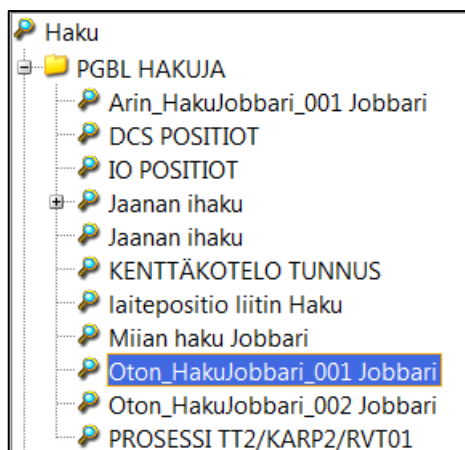
## Tuotelaji



KUVA 22. Tuotelaji-näkymä

Tuotelaji sisältää objektit jotka tulevat laiteobjektien alle. Nämä liittyvät laitteiden kytkemiseen ja kaapeleihin. Täältä asetetaan laitteelle sitä vastaavat liittimet, kunhan ensin tiedetään tarkemmin, mikä laite on käytössä. Siihen liitetään sitten signaalikaapelit ja syöttökaapelit tarpeen mukaan.

## Haku



KUVA 23. Haku-näkymä

Hauulla voidaan etsiä tietokannan eri osista kaikkia sen eri objektityyppejä tai niiden attribuutteja. Tämä on erittäin kätevä keino useiden tietynlaisien objektien yhtäaikaiseen muokkaamiseen, koska haetut objektit ilmestyvät omaan ikkunaan. Jos halutaan vaikka lisätä tietynlainen kaapeli useaan eri laitepositioon, niin ensin haulla suodetaan tulokset, sitten ne valitaan ja raahataan kaapeli yhtä aikaa kaikkiin.

#### 4.4.2 Attribuutti-ikkuna

Tämä ikkuna on käytännössä koko ajan auki ALMA:n oikeassa laidassa. Siihen tulevat näkyviin aina valittuna olevan objektin attribuutit. Tästä esimerkkinä on automaatiopositio BZ101102 kuvassa 24.

BZ101102	
Automaatiopositio	Tyyppi
Ylempi	Dokumenttilinkki
Tunnus	BZ101102
Nimi	BA-10101A PILOT A2
PI-kaaviotunnus	NPO-10396_1
Tunnus kirjainosa	B
Valvonta-aika (open, start) [s]	10
Järj.nro.	101102
Valvonta-aika (close, stop) [s]	2
Toimintaraja	LIEKKI
Tapahtumatyyppi	ALARM
Invertoi tila (o)	
Tapahtumateksti	VALELIEKKI (A11)
Tapahtumatyyppi 2	
Invertoi 2 tila (o)	
Tapahtumateksti 2	EI SYTY (A12)
Tapahtumatyyppi 3	
Invertoi 3 tila (o)	
Tapahtumateksti 3	LIEKKI SAMMUNUT (A13)
Tapahtumatyyppi 4	
Invertoi 4 tila (o)	
Tapahtumateksti 4	EI SAMMU (A14)
TAPAHTUMA- JA NÄYTTÖTIEDOT	
Valvomotunnus	
Operointinäyttö	

KUVA 24. Automaatiopositio BZ101102 attribuutit

Tikapuuikkunassa on valittuna automaatiopositio BZ101102. Attribuutti-ikkunaan ilmestyy silloin sen objektin tiedot.

Attribuutit ovat syötettyjä tietoja jotka halutaan liittää objekteihin. Ne tulevat näkyviin raporteissa ja generoiduissa instrumenttikuvissa, siellä missä tiedot halutaan näkyvän.

Kuvassa 25 on esitetty laiteposition FT101101 attribuuttien välilehti **Tyyppi**. Täällä asetetaan laitteelle laitelaji. Laitelaji on se mikä määrää alas listatut attribuutit, jokaiselle laiteyydelle on annettava omaansa vastaava laitelaji.

The screenshot displays a software interface for configuring device attributes. The top section shows a table for the device position FT101101. The table has three columns: 'Valittu', 'Linkki', and 'Lisätyyppi'. The first row shows 'FT101101' in the 'Valittu' column, a blue arrow in the 'Linkki' column, and 'F2002' in the 'Lisätyyppi' column. Below the table, the 'F2002' configuration form is visible. It includes a 'Laitelaji' tab, a 'Ylempi' button, and a 'Käytössä - lisätyyppi' button. The form contains several fields: 'Tunnus' (F2002), 'Nimi' (Virtauslähetin\_vortex), 'PROSESSITIEDOT', 'Käyttötarkoitus', 'Prosessiaine', and 'Prosessiaineen olomuoto'. A message 'Riittämättömät muokkaus-oikeudet...' is displayed above the fields.

**KUVA 25.** Laiteposition FT101101 laitelaji on F2002

Tietokantaan on syötetty ennalta tieto, että laitelaji F2002 on vortex-lähetin. Laitelajiin F2002 on sitten sidottu vortex-lähetimille halutut attribuutit. Virtauslähetin FT101101 sattuu olemaan vortex-lähetin ja sille asetetaan sitten laitelaji F2002.





## 4.5 Symbolikorvaus

Symbolikorvaus tarkoittaa teoriassa sitä, että tietokannan automaatiopositioista voidaan luoda valmis AutoCAD-instrumenttikuva ”nappia painamalla”.

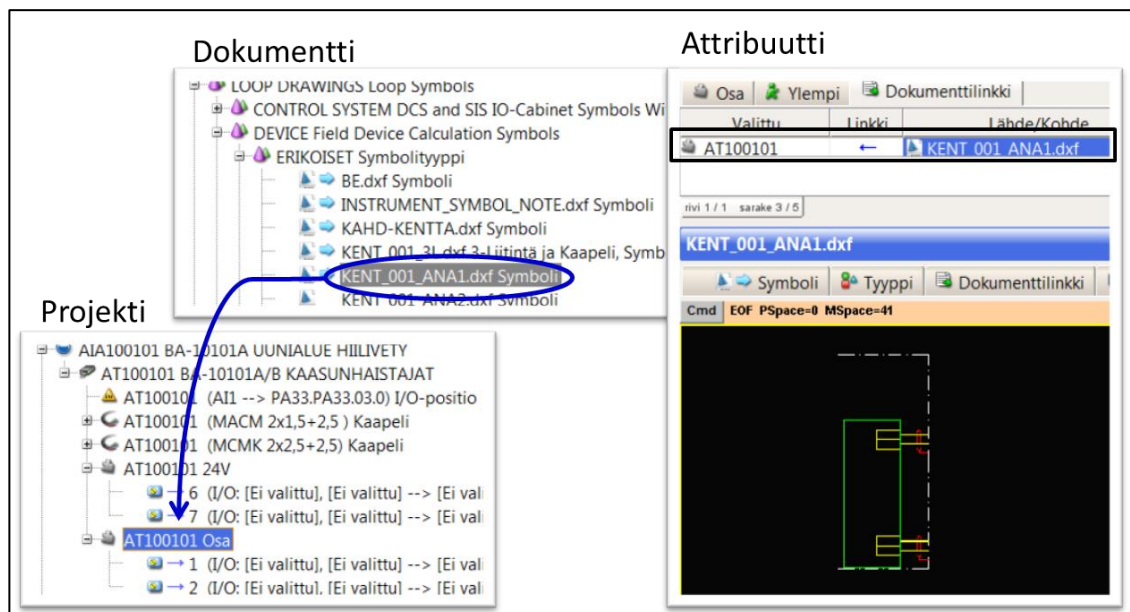
Automaatiopositiolle on ensin asetettava pohjakuva linkkaamalla pohjakuvatiedosto kyseiseen positioon, tämä tehdään raahaamalla (kuva 27).



KUVA 27. Pohjakuvan liittäminen

Pohjakuva PK\_POHJA\_5L\_1.dxf on AutoCAD:illa muokattu kuva, joka koostuu paikalleen asetelluista teksteistä joissa lukee **placeholder** tai **replace**. Nämä tekstit eivät näy generoidussa kuvassa. Kun ALMA:ssa asetetaan tietylle AutoCAD-blokille paikka-attribuutti, se hakeutuu generoitaessa pohjakuvaan sille paikalle.

AutoCAD-blokkit löytyvät dokumentti-näkymän symbolityyppien alta. Blokin määrittämisestä on havainnollistava esitys kuvassa 28.



KUVA 28. Blokin määrittäminen objektille

Kuvan 28 dokumentti-ikkunasta on löydetty haluttu blokki KENT\_001\_ANA1.dxf, sen voi nähdä attribuutti-ikkunassa. Blokki raahataan sille kuuluvalle objektille, tässä tapauksessa osa AT100101.

Osan ja blokin välisen dokumenttilinkin toteutumisen voi nähdä kuvan 28 attribuutti-ikkunan mustissa raameissa. Osa-objektit ovat niitä mistä instrumentti koostuu. Niille linkataan AutoCAD-blokit ja niihin sidotaan myös kytkentäliittimet.

Jotta symbolikorvaus toimii, ovat blokit ja pohjat oltava linkitettynä kaikille tarvittaville objekteille. Lisäksi objekteille on asetettava halutut attribuutit.

Ohjelma ei ole erityisen tarkka siitä, mikä blokki täytyisi liittää mihinkin osaan. Niitä voi periaatteessa linkata mielivaltaisesti ja generoitu kuva voi silti näyttää oikealta.

Kaksi täysin samanlaista instrumenttia voidaan tehdä usealla eri tapaa ja kaikki niistä ovat periaatteessa oikein.

Tämä aiheuttaa sekaannusta suunnittelijalle, koska ohjelma antaa liikaa vapautta. Nopeaa suunnittelutyötä haluttaessa tarvitaan selvät sävelet, miten laite tulee rakentaa. Yrityksen on itse annettava tähän esimerkit tai säännöt, joiden avulla suunnittelijat sitten tekevät työnsä.

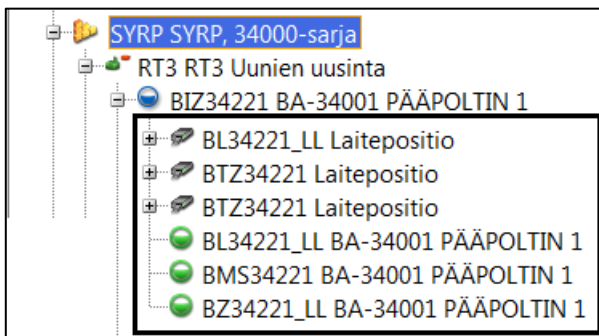
Instrumenttikuva voi vaatia erittäin paljon hienosäätöä ennen kuin se on valmis.

## 5 INSTRUMENTOINTI ALMA:SSA

### Esimerkkisuoritus

Käydään läpi esimerkki instrumentoinnista ALMA:ssa. Projektin on tätä varten täytynyt edetä siihen pisteeseen, että instrumentointi pystytään toteuttamaan. Tämä ei ole ohje, vaan enemmänkin kuvaus siitä miten instrumentti luodaan ALMA:an.

Ensin täytyy lisätä olemassa olevaan projektiin automaatiopositiot. Ne voidaan luoda yksitellen, mutta tehokkain tapa on ajaa ne Import-komennolla jostain valmiiksi luodusta Excel-taulukosta sisään ALMA:an. Taulukko on alun perin luotu niin, että samalla ajetaan tietokantaan myös jokaisen automaatioposition attribuutit. Tarkastellaan liekinvalvontaposition BIZ34221 BA-34001 PÄÄPOLTIN 1.



KUVA 29. Laitepositiot ja sovelluspositiot

Automaatioposition alle lisätään sille tulevat laitepositiot ja sovelluspositiot, kuvassa 29 rajattu mustalla. Tässä siis jo ennakkoon tiedetään, mitä osia instrumentissa on ja mitä se tulee suorittamaan.

BTZ34221 – laitepositioita on 2 koska instrumentti jatkuu generoidussa instrumenttikuvassa toiselle sivulle.

Sen jälkeen tehdään laitelajien määrittäminen. Kuvassa 29 olevat laitelajit:

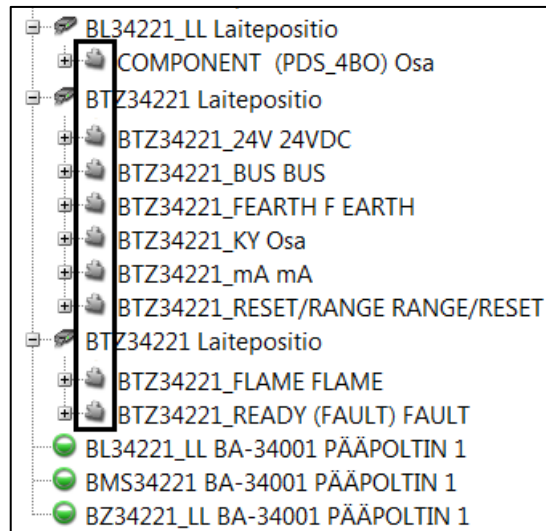
- BL34221                   X2005 Merkkivalo
- BTZ34221                B2001 Liekinvalvoja (molemmat)

Annetaan myös sovellustyypit:

- BL34221\_LL            N\_F\_S3000 Binäärilähtö, ML-ohjaus DCS-BO
- BMS34221             N\_F\_MS1100S Huolto-ohitus, tapahtuma
- BZ34221\_LL           N\_F\_BZ1001 Liekinvalvonta, moottorilohkototeutus

Laitelajit ovat tietokantaan ennalta ajettuja malleja, jotka nyt määräävät kunkin laitelajin ja sovelluksen attribuutit. Ne raahataan laitepositioihin kohdasta **Sovellustyypit**.

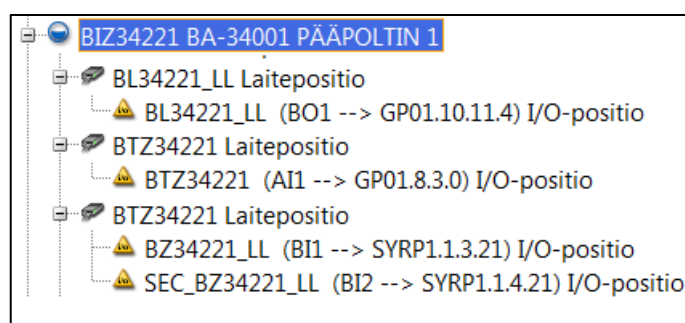
Sovelluspositioiden alle ei tule enempää objekteja. Laiteposioiden alle tulevat osat, liittimet, kaapelit ja I/O-positiot. Ensin katsotaan kentän pääty, eli osa-objektit.



KUVA 30. Osa-objektit

Kuvan 30 ylin osa COMPONENT (PDS\_4BO) otetaan tuotelaji-puusta. Se sopii liittimiltään merkkivalon BL34221\_LL yhteyteen. Sitten osaan linkitetään symbolikirjastosta AutoCAD-blokki joka vastaa merkkivaloa instrumenttikuvan kenttä-osiossa, kuten kuvassa 28 on aiemmin esitetty. Samalla tavalla ositetaan kaikki loputkin laitepositiot.

Tätä ennen suunnittelijan tulee toki tietää, miltä valmiin instrumenttikuvan tulee näyttää edes suurin piirtein, jotta osataan varata oikeannäköiset blokit oikeaan paikkaan.



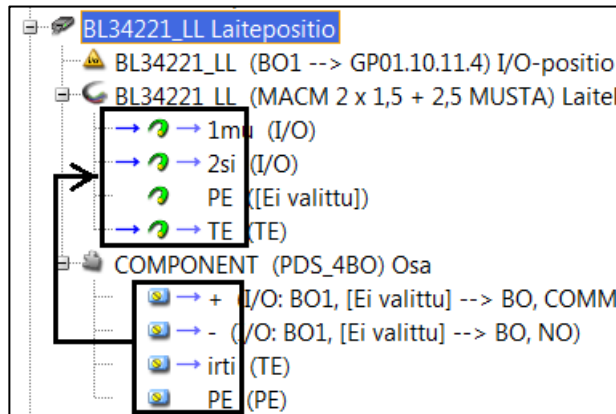
KUVA 31. I/O-objektit

Tiedot ovat luettavissa attribuutti-ikkunasta. Kuvan I/O-objektille BTZ34221 on varattu analogiasignaali AI1. Sen Prosessiasema on GP01, korttikehikko 8, korttipaikka 3 ja korttikanava 0.

Instrumenttikuvassa osat sijoittuvat kentälle ja kytkentähuoneisiin. AutoCAD-blokkeina ne käsittävät varsinaiset prosessi-instrumentit, liittimet, koekaapit, johdot, ristikytkennän ja syötön. Osa kuuluu sille laitepositiolle minkä alla se on projektipuussa.

I/O-positiot ovat kuvassa 31 keltaiset objektit. Niille on projektin aikana suunniteltu paikat I/O-korteissa ja blokit sijoittuvat instrumenttikuvien oikeaan laitaan.

Kaapelointi tapahtuu liittämällä kytkentäliitos johtimeen ja johdin liitetään sitten seuraavaan kytkentäliitokseen.

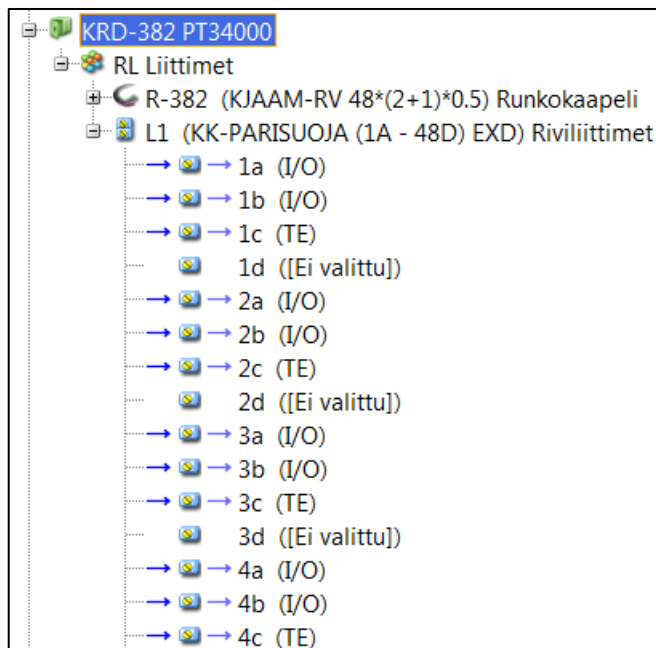


KUVA 32. Johtimien kytkentä

Kuvassa 32 on merkkivalon

BL34221\_LL johdotus, mikä on yksi johdinpari + häiriösuoja TE. Osa liittimiltä tehdään linkki (siniset nuolet) johtimeen. Sen jälkeen johdin voidaan linkittää eteenpäin. Ikään kuin kaapelia kytkettäisiin. Kaapeliksi on valittu MACM 2x1,5+2,5 MUSTA laitekaapeli.

Merkkivalo on esimerkkinä yksinkertainen mutta esimerkiksi liekinvalvojan kaapelointi on toteutettu 7-parisella erikoiskaapelilla. Molempien laitepositioiden osista tehdään linkki kaapelille.



KUVA 33. Kenttäkotelo

Sitten kaapeli linkitetään eteenpäin kenttäkotelolle jos sellainen on, kuten tässä tapauksessa. Kenttäkotelo sitten listaa tulevat linkit sen riviliittimille, kuten kuvassa 33. Projektissa on tätä ennen toki suunniteltu, mikä instrumenttikaapeli liittyy mihinkin riviliittimeen.

Kuvassa on ylhäällä vielä mainittu KJAAM-RV 48\*(2+1)\*0,5 -runkokaapeli. Tämä sitten jatkaa kotelolta eteenpäin kytkentähuoneeseen ja ristikytkentään, kuten kuvan 26 grafiikkaeditorissa on esillä. Kaapelien linkittämistä jatketaan sitten niin kauan, kunnes tullaan I/O-kortteille.

Jos edelliset osiot on tehty oikein, voidaan generoida valmis instrumentointikuva.

Generointia varten automaatiopositio tarvitsee pohjakuva-dokumentin. Eli se pitää lisätä viimeistään tässä vaiheessa. Jos instrumentti tarvitsee kaksi sivua, niin sitten lisätään toinen pohjakuva.

Annetaan viimeistään tässä vaiheessa blokeille paikat, joka tapahtuu antamalla paikka-tieto attribuuteissa. Sitten dxf-generaattorilla lisätään uusi dokumentti joko palvelimelle tai oman koneen kovalevyille. Tällä tavoin ohjelma tekee ns. paikkavarauksen generoitavalle dokumentille.

Liitteessä 2 on BIZ34221 kaikkine osineen. Siihen on lisätty pohjakuvat ja valittu ALMA:n yhteydessä olevalta tiedostopalvelimelta SYRP 34200-sarja, johon dokumentti generoidaan.

Valmis, generoitu instrumenttikuva on liitteessä 3. Instrumentti jatkuu toiselle sivulle jossa on lisäksi merkkivalo.

BTZ34221 osat löytyvät nimettyinä instrumenttia kuvaavan laatikon sisältä, omilla kytkennöillään varustettuna. Vertaamalla liitettä 2 ja 3 nähdään mistä kukin osa löytyy.

Näin tietokantaan luodaan yksittäinen laite ja sille generoidaan instrumenttikuva. Kuvaa voidaan tarvittaessa vielä muokata AutoCAD:lla.

## 6 TIETOKANTOJEN KÄYTTÖ

### 6.1 ALMA:n käyttäminen

Instrumentin generointi esimerkissä on huono näytös ALMA:n ominaisuuksista. Siinä tehtiin ja generoitiin yksi instrumentti. Ollakseen täydellinen, piiriin tarvitsee asettaa kaikki oikeat attribuutit kaikille objekteille. Myös käyttöliittymän käyttäminen täytyy osata. Pitää tietää, mitä kannattaa muokata ja mihin vielä ei kannata koskea. Sitä ei opi kuin tekemällä. Olisi erittäin tehotonta tehdä suuren laitoksen sadat piirit tällä tapaa.

Yksi tapa tehostaa työskentelyä on haku-toiminnon käyttö. Sillä saadaan suodatettua useita objekteja käsiteltäväksi samaan aikaan. Attribuutit, kaapeloinnit, laitepositiot ja liitokset voidaan tehdä samaan aikaan kaikkiin. Lisäksi, kun tiedetään etukäteen miten yksi instrumentti koostuu, voidaan kaikki samanlaiset instrumentit tehdä kopioimalla. Tämä keino on hyödyllisin jo valmiina tietokannassa olevien tietueiden muokkaukseen.

Vielä yksi, ehdottomasti tehokkain keino, on Import-toiminnon käyttäminen. ALMA:n objekteilla ja attribuuteilla on kaikilla oma tunnuksensa, oma pääavain. Tätä avainta käytetään tietueen paikantamiseen. Excel-taulukossa oleva avain määrää, mihin siinä lukevat kenttätiedot kohdennetaan ALMA-tietokannassa. Näin voidaan ALMA:an ajaa automaatiopositiot, laitteet, kytkennät, attribuutit, kaikki. Ongelmana on se, että tällainen Excel-taulukko pitää ensin osata tehdä.



## 6.2 Excel-taulukot ja import

Neste Jacobsissa käytetään import-tapaa paikoittain. Käytännössä kaiken tiedon ajaminen Excel-taulukoiden avulla onnistuu, mutta muokkaaminen tehdään ALMA:sta käsin. Ongelma tulee eteen lähinnä kytkentöjen tekemisessä. Ristikytkentöjä ja muita on niin paljon, että semmoisen taulukon tekeminen ja tarkistaminen on työlästä.

Alla esimerkkikuva 34 I/O-positioiden import-taulukosta. Siinä on kymmenen saraketta. Ylimmällä rivillä on ALMA:n avaintunniste ja niiden alla sisään ajettavat tiedot. Vasemmalta oikealle, ensin tulevat objektit: automaatiopositio, laitepositio ja I/O-positio. Sen jälkeen tulee objekteihin sidotut attribuutit, eli I/O-kortin kytkentätiedot.

INSTRUMENT	HOB	HOB	IOTAG	IOTAG	IOTAG	IOTAG	IOTAG	IOTAG	IOTAG	IOTAG
PARENT LOOP	PARENT INSTRUMENT	ANSWER IOTAG	ANSWER IOTAG	ANSWER IOTAG	ANSWER IOTAG	ANSWER IOTAG	ANSWER IOTAG	ANSWER IOTAG	ANSWER IOTAG	ANSWER IOTAG
ALMA_CODE	ALMA_CODE	ALMA_CODE	ALMA_F	ALMA_IO_CABINET	ALMA_PROCI	ALMA_CARD	ALMA_CARD	ALMA_CARD	ALMA	ALMA_CARD_TYPE
HS101121	HS101121_C	HS101121_C	BI1	KH10_LK307-1E	RT3_SIS1	RACK09	RACK09	01 JA 02	2	SDI-1624_Exd
BI2101121	BZ101121_LLL	BZ101121_LLL	BI1	KH10_LK307-1E	RT3_SIS1	RACK09	RACK09	01 JA 02	3	SDI-1624_Exd
HS101122	HS101122_O	HS101122_O	BI1	KH10_LK307-1E	RT3_SIS1	RACK09	RACK09	01 JA 02	4	SDI-1624_Exd
HS101122	HS101122_C	HS101122_C	BI1	KH10_LK307-1E	RT3_SIS1	RACK09	RACK09	01 JA 02	5	SDI-1624_Exd
BI2101122	BZ101122_LLL	BZ101122_LLL	BI1	KH10_LK307-1E	RT3_SIS1	RACK09	RACK09	01 JA 02	6	SDI-1624_Exd
HS101123	HS101123_O	HS101123_O	BI1	KH10_LK307-1E	RT3_SIS1	RACK09	RACK09	01 JA 02	7	SDI-1624_Exd
HS101123	HS101123_C	HS101123_C	BI1	KH10_LK307-1E	RT3_SIS1	RACK09	RACK09	01 JA 02	8	SDI-1624_Exd
BI2101123	BZ101123_LLL	BZ101123_LLL	BI1	KH10_LK307-1E	RT3_SIS1	RACK09	RACK09	01 JA 02	9	SDI-1624_Exd
HS101124	HS101124_O	HS101124_O	BI1	KH10_LK307-1E	RT3_SIS1	RACK09	RACK09	01 JA 02	10	SDI-1624_Exd
HS101124	HS101124_C	HS101124_C	BI1	KH10_LK307-1E	RT3_SIS1	RACK09	RACK09	01 JA 02	11	SDI-1624_Exd
BI2101124	BZ101124_LLL	BZ101124_LLL	BI1	KH10_LK307-1E	RT3_SIS1	RACK09	RACK09	01 JA 02	12	SDI-1624_Exd
HS101125	HS101125_O	HS101125_O	BI1	KH10_LK307-1E	RT3_SIS1	RACK09	RACK09	01 JA 02	13	SDI-1624_Exd
HS101125	HS101125_C	HS101125_C	BI1	KH10_LK307-1E	RT3_SIS1	RACK09	RACK09	01 JA 02	14	SDI-1624_Exd
BI2101125	BZ101125_LLL	BZ101125_LLL	BI1	KH10_LK307-1E	RT3_SIS1	RACK09	RACK09	01 JA 02	15	SDI-1624_Exd
HS101126	HS101126_O	HS101126_O	BI1	KH10_LK307-1E	RT3_SIS1	RACK09	RACK09	03 JA 04	1	SDI-1624_Exd
HS101126	HS101126_C	HS101126_C	BI1	KH10_LK307-1E	RT3_SIS1	RACK09	RACK09	03 JA 04	2	SDI-1624_Exd
BI2101126	BZ101126_LLL	BZ101126_LLL	BI1	KH10_LK307-1E	RT3_SIS1	RACK09	RACK09	03 JA 04	3	SDI-1624_Exd
HS101127	HS101127_O	HS101127_O	BI1	KH10_LK307-1E	RT3_SIS1	RACK09	RACK09	03 JA 04	4	SDI-1624_Exd
HS101127	HS101127_C	HS101127_C	BI1	KH10_LK307-1E	RT3_SIS1	RACK09	RACK09	03 JA 04	5	SDI-1624_Exd
BI2101127	BZ101127_LLL	BZ101127_LLL	BI1	KH10_LK307-1E	RT3_SIS1	RACK09	RACK09	03 JA 04	6	SDI-1624_Exd
HS101128	HS101128_O	HS101128_O	BI1	KH10_LK307-1E	RT3_SIS1	RACK09	RACK09	03 JA 04	7	SDI-1624_Exd

KUVA 34. I/O-import -taulukko

Tämän kaltaisilla taulukoilla voitaisiin kaikki tiedot ajaa tietokantaan. Ainakin I/O-tietojen tapauksessa näin tehdään NJ:llä.

Ongelmille jää kuitenkin tilaa vielä jälkeensä. Tietojen muokkaaminen on hidasta. Kun sattuu inhimillinen virhe, kokematon työntekijä muokkaa laitepositiota väärin, niin viite-eheys särkyä. Kaikki positioon liitetyt linkit ja attribuutit viittaavatkin nyt paikkaan mikä ei enää ole sillä nimellä. Tämä ongelma tulee esiin jossain vaiheessa ja se pitää korjata manuaalisesti, katsomalla taulukosta ja syöttämällä tiedot käsin.

Syksyllä annetuista tehtävistä jotkut jäivät ajatuksen tasolle tai kesken. Tästä on seuraavaksi yksi hyvä esimerkki, joka liittyy taulukoihin.

Tehtävänä oli luoda helposti tulkittava, yksinkertainen ja toimiva Excel-taulukkopohja ulkomaisen alihankkijan työstettäväksi. Taulukoon täytettäisiin kaikki projektin laitepositioiden attribuutit. Lisäksi se olisi pitänyt saada ladattua suoraan ALMA:an import-komennolla. Alihankkijan tulisi vain katsoa oikeat attribuutit laitetiedoista, täyttää ne taulukkoon ja lähettää taulukko takaisin.

Kuten olettaa voi, ongelmia on paljon. Yhteisen kielen puute, sähköpostiviestinnän kankeus, välimatkat jne. Alihankkijalle ei anneta ALMA:n muokkausoikeuksia koska heitä ei ole opetettu käyttämään järjestelmää.

Tehtävä oli helppo ymmärtää mutta käytännössä vaikea toteuttaa. ALMA:sta saatiin export-komennolla jonkinlainen taulukkopohja, mitä lähdettiin muokkaamaan. Mutta ALMA:n laitepositioita on erilaisia ja jokaisella oma attribuuttipohjansa. Attribuutteihin liittyvät pääavaimet voivat olla samoja eri laitteilla ja aiheuttaa ristiriitoja import-komennossa. Attribuuttien järjestys vaihteli eri laitteilla, oli hidasta löytää yhteneväiset osat. Jos Excelin rivit käännetään pystyasentoon, ALMA ei enää tunnista avaimia. Näitä ongelmia riittää loputtomiin. Tuli selväksi, että tähän tehtävään tarvittiin kokeneempaa taulukontekijää.

Kuitenkin varmasti useita vastaavia taulukoita tehdään ja muokataan projektin kuluessa. Joihinkin on valmiit pohjat ja toisiin ei. Uusien pohjien luominen on hidasta ja vaikeaa.

### 6.3 Projektinhallinnan ongelmat

Projektissa työskennellään ns. hallitsemattomassa ympäristössä. Koko organisaation suurta tietomäärää työstää joukko työntekijöitä, jotka käyttävät siihen eri systeemejä ja dokumentteja. Systeemit kantavat mukanaan turhaa dataa, mikä niihin on jäänyt päivitysten ja väärän käytön takia.

Uuden projektin alkaessa, suuri määrä tietoa kerätään yhteen eri dokumenteista ja systeemeistä. Tieto tallennetaan erilaisiin muotoihin, kuten Excel, Word ja AutoCAD-dokumentteihin. Nämä ovat enimmäkseen projektikohtaista tietoa eikä niitä pidetä elämäntaari-dokumentteina, eli niitä ei päivitetä sen jälkeen, kun ne on luovutettu asiakkaalle.

Tieto on tallennettu johonkin valmiiseen pohjaan tai käyttäjän omien mieltymysten mukaan. Tiedostot nimetään ilman minkäänlaista yhteistä käytäntöä ja ne tallennetaan käyttäjien työkoneiden kovalevyille, vieläpä useisiin eri kansioihin. Kun projekti etenee, useampia työntekijöitä tulee mukaan ja he tarvitsevat informaatiota. Työntekijät jakavat aiemmin muokattuja tiedostoja keskenään ja niitä muokataan taas lisää uusien mieltymysten mukaan. Tämä toistuu useasti projektin aikana.

Samaa tietoa toistetaan useasti ja muokataan projektin edetessä. Tieto ei välttämättä muutu niin että se olisi väärin, vaan niin että siitä tulee hajanaista ja vaihtelevaa niin sisällöltään kuin muodoltaan.

Hajanainen tieto vaikeuttaa projektinhallintaa ja organisaation kykyä suoriutua tehtävistä. Tiedon laatu heikkenee ja virheiden mahdollisuus on suuri. Tietoa on vaikea jäljittää ja hallita, eikä se palvele ketään niin kuin pitäisi. Pieniä muutoksia ei dokumentoida ja ne jäävät hetkeksi vain muokkaajan omaan päähän, kunnes katoavat kokonaan.

Ongelmaan ei ole selvää ratkaisua. Tämä johtuu lähinnä kiireellisistä aikatauluista ja siitä, ettei selkeälle tiedonhallinnalle ole esitetty vaatimuksia. Ainoa vaatimus on, että tieto on virallisten standardien mukaan oikein ja se on toimitettu ajallaan virallisten dokumenttien muodossa.

Nykyään suurin osa tiedosta päätyy jonkinlaiseen suunnittelujärjestelmään, missä virallisesti toimitettavat dokumentit luodaan ja julkaistaan. Dokumentit ja tiedot tallennetaan oikein ja niitä voidaan käyttää uusiksi. Tällä hetkellä eletään kuitenkin siirtymäaikaa, missä paperidokumentit, tiedostot ja tietovarastot elävät samanaikaisesti. Tämä tekee tiedon hallinnasta vähintäänkin vaikeaa, vaikka käytössä ovat tietokantapohjaiset suunnittelujärjestelmät.

Suurimpana ongelmana kaupallisissa suunnittelujärjestelmissä, on niiden tarkoitus olla mahdollisimman laajakäyttöisiä. Ne palvelevat tiettyä käyttötarkoitusta johonkin pisteeseen asti. Sen jälkeen kaikki pitää muokata käyttäen jonkinlaisia raporttityökaluja, ohjelmointikieltä tai kolmannen osapuolen ohjelmaa. Tämä vaatii taas erikoisosaamista, eikä pelkkä automaattisuunnittelijan koulutus välttämättä riitä. Suunnittelijan taidot, keskittyminen ja työmoraali uppoavat loputtomaan tiedonhallintaan.

(Neste Jacobs Oy, asiantuntija)

## 7 TYÖN TARKASTELU JA LOPPUSANAT

Tiedonhallintaan liittyvät ongelmat ovat periaatteessa näkymätön ongelma mutta ne tuntuvat tuloksessa sitäkin kovemmin. En olisi itse tiennyt näiden ongelmien olemassaolosta ilman työnohjaajan kanssa käytyjä keskusteluja.

Tämän työn suurin haaste oli minulle lähes täysin uuden asian oppiminen ja ymmärtäminen. Tietokantoja ei oltu juuri käsitelty opiskelun aikana, koska ne eivät ensisijaisesti kuulu automaatioalan pääkohtiin. Tietokannat ovat enemmän tietotekniikan ja järjestelmäasiantuntijoiden heiniä. Nyt oli kuitenkin pakko astua rajojen yli ja opetella uutta.

Alussa tietomäärä tuntui raskaalta ja sitä oli vaikea käsitellä. Kuitenkin ajan kanssa se alkoi selvitä ja arvokkaan työkokemuksen myötä pystyin aloittamaan työn kokoamisen.

Toisena haasteena oli työn rajaaminen. Kuinka pystyn kirjoittamaan aiheesta tarpeeksi, kuitenkaan menemättä liikaa yksityiskohtiin. Tarkoituksena oli opastaa lukija ymmärtämään kahta eri maailmaa, instrumentointia ja tietokantoja. Kuinka nämä ovat yhteistyössä ja miksi on hyvä tietää molemmista. Työn kokoamiseen ei annettu rajoitteita tai ohjeita. Minun tuli kirjoittaa siitä niin, että itseni kaltaiset noviisit ymmärtäisivät sen.

Mielestäni työssä on hyvin esillä ajatusten herättäminen. Insinöörin tulee olla kiinnostunut asioista hieman pintaa syvemmältä ja pitää katse kehityksessä. Ongelmat on tuotava esiin, jotta ratkaisu niihin lähtee hautumaan.

Heikompana puolena työssä on mielestäni ymmärrettävyys. Työtä kannattaa tarkastella samalla kun itse työskentelee ALMA:n parissa. Tietenkään kaikilla ei tähän ole mahdollisuutta. Järjestelmässä on niin paljon omia käsitteitä, että ilman käytännön kokemusta niistä on vaikea saada koppiä.

Mielestäni työ saavuttaa tavoitteensa opastajana. Nyt ymmärrän ainakin osittain tietokantojen merkityksen ja niiden käytön instrumentointityössä. Koen, että pystyisin opettamaan tätä tietoa muillekin. Alojen välinen tehokas yhteistyö kuitenkin vaatii molempien asioiden ainakin osittaista ymmärtämistä.

## LÄHTEET

AB+ Trainer – Tietokannat Access, Keskiikonen, M. 2001. (Luettu 25.10.2013)

Access 2003 Perusteet – Yleistä, Tietokannan peruskäsitteitä (Viitattu 28.10.2013)  
<http://www.ratol.fi/opensource/ac/yleista/cnt.html>

ALMA ja Neste yhteistyössä kohti kestävästä kehitystä. (Viitattu 14.12.2013)  
[http://www.alma.fi/Suomeksi/Kohtaamispaikka/ALMA-asiakaskirje\\_kesakuu\\_2009](http://www.alma.fi/Suomeksi/Kohtaamispaikka/ALMA-asiakaskirje_kesakuu_2009)

ALMA Tuoteratkaisut (Viitattu 15.12.2013)  
<http://www.alma.fi/Suomeksi/Tuoteratkaisut>

Relaatiotietokantojen peruskäsitteet, Jyväskylän yliopisto (Luettu 30.10.2013)  
<http://appro.mit.jyu.fi/doc/tiedonhallinta/tietokannat/index2.html>

Lähiverkkopohjaisten relaatiotietokantajärjestelmien tehokkuustekijöiden arviointi, Fa-  
 gerholm, J. Diplomityö. 1990. (Viitattu 5.11.2013)

Neste Jacobs in brief  
<http://www.nestejacobs.com/index.php?page=2/6&lang=en>

Neste Jacobs Oy, syksy 2013  
 Asiantuntijan lausunnot ja opetukset

Mt. San Antonio College, Lecture Tutorial ACC1 (Luettu 10.11.2013)  
[http://www.mtsac.edu/~rpatters/CISB15/Tutorials/TutorialACC\\_01/TutACC01/LectureFrame.htm](http://www.mtsac.edu/~rpatters/CISB15/Tutorials/TutorialACC_01/TutACC01/LectureFrame.htm)

SQL, Structured Query Language, 23.4.2013 (Viitattu 15.11.2013)  
<http://fi.wikipedia.org/wiki/SQL>

Tietokantojen historia ennen SQL:ää, Eila Helena Salmela, 2007 (Viitattu 15.10.2013)  
<http://www.cs.helsinki.fi/u/kerola/tkhist/k2007/alustukset/tietokannat/TiKaHistoria.pdf>

Tietokantojen suunnittelu ja indeksointi, Hovi, Huotari, Lahdenmäki, 2003 (Luettu 22.10.2013)

Älykkään opastusjärjestelmän toteuttaminen: verkostoon perustuva toteutus, 4.1. Relaa-  
 tiotietokannoista ja tietokannoista yleensä (Luettu 28.10.2013)  
<http://www.mit.jyu.fi/opiskelu/seminarit/ohjelmistotekniikka/aopast/#Heading12>

## LIITTEET

## Liite 1. ALMA käyttöliittymä

The screenshot displays the ALMA system interface for project configuration. The main window is titled 'BZ101102' and contains several panes:

- Top Bar:** Includes navigation options like 'Irdosto', 'Näkymä', 'Apuohjelmat', 'Toiminto', 'Asetukset', 'Järjestelmä', 'Ohje', 'Uutisia, vinkkejä & ohjeita', and 'Näyttilä'.
- Left Pane (Ilanamans):** Lists projects with columns for 'Projektin nimi', 'Projektin tyyppi', and 'Projektin tila'. Projects include 'JAL\_TL1\_PExx TL1\_PROJEKTIIT', 'JAL\_TL2 KH03\_KH04\_KH05 2200- 280', 'JAL\_TL2 KH03\_1\_KH 7700- JA 9600-Sarja', 'JAL\_TL3 KH13 VALLAKAINEN BIO2 SIIRTO', 'JAL\_TL4 HH-venttiilit', 'KH10-ALMAAN 3000, 10300, 10400, 104100, 104200, 104300', 'PE9145 TL1 kaasuilmaisinten uusinta', 'PECT BESSI Kevytbensiniin Isomerointi', 'PG00xx TL2 FCC S2012 pienytöt TEHTY2012', 'PHCN KARP3 automaation uusinta', 'PHCO RT3 uunien uusinta', and 'RT3 RT3 Uunien uusinta'.
- Middle Pane (Hierarkia):** Shows a hierarchical tree structure of automation tasks, including 'JALPRV PORVOON JALOSTAMO', 'KH\_ISBL ISBL KYTKENTÄTILAT', 'KH\_OSBL OSBL KYTKENTÄTILAT', 'M\_MUUNTAMOKYTKENTÄTILAT', 'JTER\_PTER\_TERMINAALIT', 'SAT SATAMA', 'SLO SAILIOALUE', 'TKO TUTKIMUS JA KEHITYS', 'TL1 TUOTANTOLINJA 1', 'TL2 TUOTANTOLINJA 2', 'ALKY ALKY, 9800-sarja', 'FCC FCC, 2400-sarja', 'FCC\_KH\_1 FCC\_KH\_1, 7700-sarja', 'FCC\_KH\_2 FCC\_KH\_2, 9600-sarja', 'FCC\_KH\_3 FCC\_KH\_3, 6100-sarja', 'HW4 HW4, 33000-sarja', 'HW5 HW5, 63000-sarja', 'KARP2 KARP2, 2800-sarja', 'KTO3 KTO3, 2500-sarja', 'KTO3\_KH KTO3\_KH, 9400-sarja', 'LK2 LK2, 2300-sarja', 'MH MH, 2600-sarja', 'MH\_KH MH\_KH, 8900-sarja', 'NKIT NKIT, 37000-sarja', 'RTO3 RTO3, 33300-sarja', 'RTO4 RTO4, 33500-sarja', 'RTO5 RTO5, 33700-sarja', 'RTO6 RTO6, 33800-sarja', 'RVTO1 RVTO1, 2900-sarja', 'RVTO3 RVTO3, 33100-sarja', 'SYRP SYRP, 34000-sarja', 'TT2 TT2, 2200-sarja', 'TL3 TUOTANTOLINJA 3', 'TL4 TUOTANTOLINJA 4', 'TLY TUOTANTOLINJA YMPÄRISTÖ', and 'ZZZ\_HARJOITUS TEHDAS NI ALMA KOULUTUS'.
- Right Pane (Automaatiohistorio):** Lists automation tasks with columns for 'Tunnus', 'Nimi', 'Pi-kaaviotunnus', 'Tunnus kirjainosa', 'Valvonta-aika (open, start) [s]', 'Järi.no.', 'Valvonta-aika (close, stop) [s]', 'Toimintarajä', 'Tapahtumatyyppi', 'Invertori tila (o)', 'Tapahtumateksti', 'Tapahtumatyyppi 2', 'Invertori 2 tila (o)', 'Tapahtumateksti 2', 'Tapahtumatyyppi 3', 'Invertori 3 tila (o)', 'Tapahtumateksti 3', 'Tapahtumatyyppi 4', 'Invertori 4 tila (o)', 'Tapahtumateksti 4', and 'TAPAHTUMA- JA NÄYTTÖTIEDOT'.

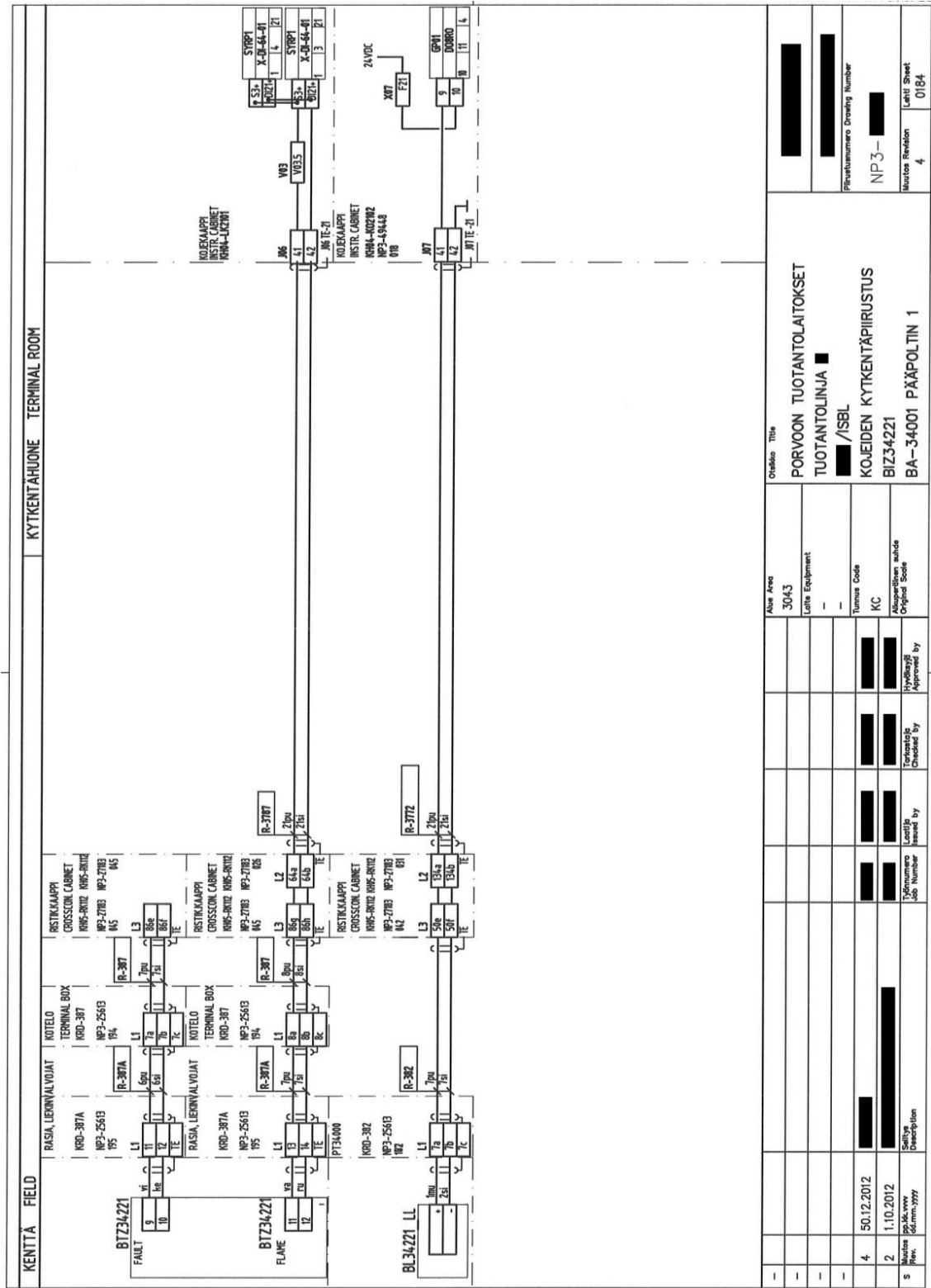
Liite 2. Generointi

The screenshot displays a software application window with the following components:

- Top Panel:** A table with columns: Lähde, Dokumenttipöytä, Dokumentti, Toiminta, Dokumenttityyppi. It lists files like 'BIZ34221 N PKAA' and 'BIZ34221 PK\_POHJA\_5L\_1.dxf'.
- Left Panel:** A tree view under 'lanamans Hierarkia' showing a project structure with nodes like 'BIZ34221 BA-34001 PÄÄPOLTIN 1' and various sub-nodes with status icons.
- Center Dialog:** A 'Lisää dokumentti' dialog box with a list of files. The file 'NP3-49914 SYRP\_34200-Sarja' is highlighted in blue.
- Right Panel:** A vertical toolbar with buttons: 'Näytä kaikki', 'Generoi/...', and 'Gener'.







KENTTÄ FIELD		KYTKENTÄHUONE TERMINAL ROOM	
Area	304.3	Order No.	304.3
Equipment	Terminal Equipment	Title	PORVOON TUOTANTOLAITOKSET TUOTANTOLINJA ■
Turnout Code	KC	Drawing Number	NP3- NP3-
Approved by	[Signature]	Revision	4
Checked by	[Signature]	Sheet	0104
Issued by	[Signature]	KONEIDEN KYTKENTÄPIIRUSTUS BIZ34221 BA-34001 PÄÄPOLTIN 1	
Job Number	[Blank]	Pöytäkirja	
Revision	4	50.12.2012	
Revision	2	1.10.2012	
Revision	1	10.06.2012	