

MOOTTORIPRIIRIKAAVIOIDEN SUUNNITTELUN AUTOMATISOINTI

Tuomas Ikonen

Huhtikuu 2013

Automaatiotekniikka
Tekniikan ja liikenteen ala





Tekijä(t) IKONEN, Tuomas	Julkaisun laji Opinnäytetyö	Päivämäärä 6.4.2013
	Sivumäärä 67	Julkaisun kieli Suomi
		Verkojulkaisulupa myönnetty (X)
Työn nimi MOOTTORIPRIIRIKAAVIOIDEN SUUNNITTELUN AUTOMATISOINTI		
Koulutusohjelma Automaatiotekniikan koulutusohjelma		
Työn ohjaaja(t) HÄKKINEN, Veli-Matti, Lehtori SELOSMAA, Seppo, Koulutusvastaava, automaatiotekniikka		
Toimeksiantaja(t) Pronor Engineering Oy BURMAN, Kari, Toimitusjohtaja		
Tiivistelmä <p>Pronor Engineering tekee pääasiallisesti sähkösuunnittelua teollisuussahojen automatisointiprojekteihin. Yritys on siirtynyt käyttämään E3-sähkösuunnitteluohjelmaa. E3:n kaikki toiminnot pystytään automatisoimaan Visual Basic Script -komentosarjoilla. Opinnäytetyön tarkoituksena oli selvittää, kuinka Excel-tiedostona olevan projektinmäärittelytiedoston pohjalta voidaan luoda uusi tietokanta E3:seen käyttäen VBS-komentosarjoja.</p> <p>Sähkösuunnittelun automatisointi nopeuttaa työskentelyä ja vähentää virheitä suunnitteludokumenteissa. Teollisuussahan sähködokumentit ovat paksu kansiolinen asiakirjoja, joissa on piirikaavioita, kaapeliluetteloita, kytkentälistoja, väyläkuvia ja muita dokumentteja. Monet näistä papereista sisältävät samoja tietoja, mutta niiden esitystavat ovat erilaisia, koska niitä tarvitaan eri työvaiheissa. E3 on tietokantapohjainen suunnitteluohjelma. Se muodostaa uuden työn aloittaessa tietokannan, johon on koottu kaikki projektin tiedot. Tietokannan pohjalta on helppo tuottaa kaikki tarvittavat suunnitteludokumentit. Muutokset tietokantaan päivittyvät kaikkiin dokumentteihin, mikä parantaa ylläpidettävyyttä.</p> <p>Työn tuloksena syntyi kaksi komentosarjaa. Näistä ensimmäinen pystyy siirtämään tietoa Excelistä E3-ohjelmaan. Ohjelma piirtää projektinmäärittelytiedostoon merkityn moottoripiirikaavion ja vaihtaa sen komponentit haluttuihin. Työssä piirrettiin kolme valmista moottoripiirikaaviota, mutta niitä voidaan piirtää myöhemmin lisää ja komentosarja osaa piirtää myös nämä. Toinen komentosarja luo Exceliin kaapeliluettelon tietokannan pohjalta. Komentosarjojen käyttämisestä laadittiin ohjeet toimeksiantajalle.</p> <p>Komentosarjat ovat liitteissä hyvin kommentoituna ja niitä voi käyttää esimerkkinä. Opinnäytetyössä on kerrottu, mistä komentosarjojen ohjelmoimisesta saa tietoa. Komentosarjojen avulla voi räätälöidä E3:n luomaan omiin tarpeisiin sopivia suunnitteludokumenteja.</p>		
Avainsanat (asiasanat) Automaatio, suunnittelu, E3, Excel, piirikaavio, komentosarja, skripti, COM, Visual Basic Script Edition, sähkösuunnitteluohjelma		
Muut tiedot		



Author(s) IKONEN, Tuomas	Type of publication Bachelor's Thesis	Date 06042013
	Pages 67	Language Finnish
		Permission for web publication (X)
Title AUTOMATING MOTOR STARTER WIRING DIAGRAM DESIGNING		
Degree Programme Degree programme in Automation Engineering		
Tutor(s) HÄKKINEN, Veli-Matti, Senior lecturer SELOSMÄÄ, Seppo, Education coordinator, automation technology		
Assigned by Pronor Engineering Oy BURMAN, Kari, General manager		
Abstract <p>Pronor Engineering makes mainly electrical designing for automation projects in sawmills. The company has started to use the E3 electrical designing program in its projects. In E3 it is possible to automate all of the functions with Visual Basic Script. The purpose of this bachelor's thesis was to find out how you can generate a E3 project from the data based on the project definition file in Excel using VBScripts.</p> <p>Automating electrical designing speeds up working and reduces errors in designing documents. Electrical documents of a sawmill are thick folder of forms including circuit diagrams, cable lists, connection tables, bus diagrams and other documents. Many of these files include the same data but the way how they are presented varies because they are used in different phases of work. E3 creates the database when new work is started where all the data in the project is gathered. Using the database it is easy to create all the documents needed. Any changes in the database are updated into every document of the project, which increases maintainability.</p> <p>As the result of the bachelor's thesis two scripts were created. The first one of these can transfer data from Excel to E3. The program generates a desired circuit diagram into the sheet and changes its components to those which are marked in the project definition file. While carrying out the bachelor's thesis three circuit diagrams were drawn but it is possible to draw more of them and include them also in the sheet. The other script generates a cable list into Excel from the project. Instructions how to use the scripts were also made for the company.</p> <p>Scripts can be found as an attachments of the bachelor's thesis and they are well commented. They can be used as an example. Information from where to learn program scripts is described in this bachelor's thesis. By using the scripts you can tailor E3 to create electrical designing documents for your individual needs.</p>		
Keywords Automation, designing, E3, Excel, circuit diagram, script, COM, Visual Basic Scripting Edition, electrical designing program		
Miscellaneous		

SISÄLTÖ

KÄYTETYT LYHENTEET	4
1 JOHDANTO	5
2 OPINNÄYTETYÖN TOIMEKSIANTAJA.....	7
2.1 Pronor Group	7
2.2 Sahateollisuus	9
3 SÄHKÖSUUNNITTELUN AUTOMATISOINTI.....	13
3.1 Automaation käsite	13
3.2 Suunnittelun automatisoinnin edut ja ongelmat	14
3.3 Komentosarjat	16
3.4 E3 Cable -sähkösuunnitteluohjelmisto	16
3.4.1 Esittely	16
3.4.2 Ohjelman käyttäminen.....	17
3.4.3 E3:n toimintojen automatisoiminen	20
3.5 Ohjelmointirajapinta	20
3.5.1 Visual Basic Scripting Edition -komentosarjakieli	20
3.5.2 WSH-komentosarjojen ajoympäristö.....	21
3.5.3 COM-komponenttimalli	21
3.6 Sähkösuunnittelu	23
3.6.1 Dokumenttityypit	23

3.6.2	Standardit.....	25
3.6.3	Sähkösuunnitteluun liittyvät ohjeet.....	28
3.7	Moottoripiirikaaviot.....	29
3.7.1	Moottoripiirikaavion osat	29
3.7.2	Moottoripiirien komponentit.....	30
4	KOMENTOSARJOJEN OHJELMOINTI.....	38
4.1	Tavoitteiden rajaus.....	38
4.2	Moottoripiirikaavioihin ja suunnitteluohjelmistoon tutustuminen	40
4.3	Moottoripiirikaavioiden piirtäminen.....	41
4.4	Projektinmäärittelytiedoston muokkaaminen.....	42
4.5	Ohjelmointiympäristöön perehtyminen.....	43
4.6	Komentosarjojen kirjoittaminen	44
5	KEHITYSKOHTEET.....	47
5.1	Lisätoiminnot	47
5.2	Käyttöliittymän kehittäminen.....	47
5.3	Virheiden hallinta.....	48
6	OPINNÄYTETYÖN SOVELTUVUUS KÄYTÄNTÖÖN	48
	LÄHTEET	51
	LIITTEET	53
	Liite 1. Taajuusmuuttajakäynnisteisen sähkömoottorin piirikaavio	53
	Liite 2. Valitut sähkösuunnittelun symbolit	54
	Liite 3. Projektinmäärittelytiedosto	56
	Liite 4. Piirikaavioiden luontikomentosarja	59

Liite 5. Kaapeliluettelon luontikomentosarja.....	66
---	----

KUVIOT

Kuvio 1. Pronor Groupin yritykset	7
Kuvio 2. Sahatavaran tuotanto Suomessa.....	10
Kuvio 3. Teollisuussahalinjasto	13
Kuvio 4. Muutokset dokumentteihin tapahtuvat kaksisuuntaisesti	15
Kuvio 5. E3:n piirtoalue	18
Kuvio 6. Komponentti-ikkuna	19
Kuvio 7. Tietokannassa olevat dokumentit näkyvät omassa ikkunassa.....	20
Kuvio 8. Excelin COM-komponenttimalli.....	23
Kuvio 9. Standardisointiorganisaatiot.....	27
Kuvio 10. Moottorisuojakatkaisimen symboli on muodostettu yhdistämällä muita standardista löytyviä symboleja.	29
Kuvio 11. Moottorinsuojakatkaisin ja sen piirrosmerkki.....	31
Kuvio 12. Kontaktori ja sen piirrosmerkki	32
Kuvio 13. Taajuusmuuttaja ja sen piirrosmerkki.....	33
Kuvio 14. Jarruvastus ja sen piirrosmerkki	34
Kuvio 15. Oikosulkumoottori ja sen piirrosmerkki	35
Kuvio 16. Ohjaussulake ja sen piirrosmerkki	36
Kuvio 17. Digitaalinen sisääntulokortti ja sen piirrosmerkki	36
Kuvio 18. Turvakytkin ja sen piirrosmerkki	37

KÄYTETYT LYHENTEET

CAD = Computer Aided Design, tietokoneavusteinen suunnittelu

COM = Component Object Model, komponenttimalli

I/O = Input/Output, tulo ja lähtö

NC = Normally Closed, kytkin joka on lepotilassa kiinni

NO = Normally Open, kytkin joka on lepotilassa auki

PLC = Programmable Logic Controller, ohjelmoitava logiikka

VB = Visual Basic, ohjelmointikieli

VBS = Visual Basic Scripting Edition, komentosarjakieli

WSH = Windows Script Host, komentosarjojen ajoympäristö

1 JOHDANTO

Tietokoneohjelmistojen kehittyessä nykyisiin teollisuuden suunnitteluohjelmiin on pyritty lisäämään ominaisuuksia, jotka helpottavat ja nopeuttavat suunnittelua. Sähkösuunnittelun asiakirjat sisältävät paljon eri dokumentteja, joissa on samoja tietoja eri tavoilla esitettyinä. Suunnitteluautomaatio tarkoittaa sitä, että suunnitteludokumentit luodaan automaattisesti yhden dokumentin pohjalta. Niitä ei siis tarvitse tehdä yksi kerrallaan. Myös viivojen ja piirrosmerkkien piirtäminen pystytään automatisoimaan. Tämä nopeuttaa suunnittelutyötä. Se myös vähentää virheitä ja parantaa ylläpidettävyyttä. Jos dokumentteihin tehdään muutoksia, automaatio pitää huolen, että kaikki muutokset tulevat täsmällisesti myös kaikkiin muihin dokumentteihin. Suunnittelun automatisointia on useimmissa johtavissa suunnitteluohjelmissä. Autodeskin tuotteet, Solidworks sekä kotimaiset Vertex- ja CADS-ohjelmistot sisältävät jo suunnitteluautomaatiota.

Suunnittelun automatisoinnista ei ilmiönä ole kirjoitettu paljon. Tietotekniikan nopean kehittymisen ansiosta suunnittelun automatisointi lisääntyy tulevaisuudessa. 1800-luvulla teollistumisen aikakaudella koneet lisäsivät tuotantoa. Myöhemmin tuli automaatiotekniikka, joka etenkin 1970-luvulla vähensi yksinkertaisia ja toistuvia työvaiheita teollisuudessa, jotka aikaisemmin tehtiin käsin. Teollistuneista länsimaista tämän kaltainen yksinkertainen tuotanto siirtyi Aasian. 2000-luvulla tietotekniikan suosio on kasvanut ja se on helpottanut monia arkisia toimintoja. Pankkiasiat, junalippujen ostaminen, kirjojen lainaaminen kirjastosta, pullojen palautus, postipakettien lähettäminen ja ostosten maksaminen ruokakaupassa on jo automatisoitu eikä niiden hoitamiseen tarvita enää virkailijaa tai myyjää. Tietokoneiden käyttämisestä on tullut koko ajan arkipäiväisempää. Useimmilla länsimaalaisilla kotitalouksilla on jo useampi tietokone, joilla hoidetaan työ- ja vapaa-ajan asioita. Tekniikan alan suunnittelutöissä on paljon toistuvia ja yksinkertaisia

työvaiheita, kuten viivojen piirtämistä ja tietojen täyttämistä taulukkolaskentaohjelmaan, jotka voidaan automatisoida. Siksi on selvää, että myös suunnittelun automatisointi lisääntyy tulevaisuudessa.

Pronor Engineering Oy:lle suunnittelun automatisointi on ajankohtainen aihe, koska yritys on siirtymässä käyttämään uutta E3 Cable -nimistä sähkösuunnitteluohjelmistoa. Ennen yrityksellä oli käytössään Vertex ED. Uusista projekteista luodaan aina projektinmäärittelytiedosto. Se on Excel-tiedosto, jossa on lueteltu tietoja projektissa käytettävistä laitteista. Kaikki suunnitteludokumentit luodaan tämän tiedoston pohjalta. Ongelmana vanhassa suunnitteluohjelmassa on, että se pystyy luomaan automaattisesti n. 90 % tarvittavista suunnitteludokumenteista projektin määrittelytiedoston pohjalta. Loppu 10 % täytyy tehdä käsin. Tämä on työlästä ja aiheuttaa helposti virheitä, mikä johtaa usein ongelmiin kentällä. E3 pystytään ohjelmoimaan siten, että tarvittaessa kaikki dokumentit pystytään luomaan projektin määrittelytiedoston pohjalta. Vertex ED on aikoinaan luotu ohjelmistoarkkitehtuurin päälle, joka on nykyään vanhentunut ja siihen ei enää löydy osaamista.

Opinnäytetyön tarkoituksena oli selvittää, kuinka Excel-tiedostona olevan projektinmäärittelytiedoston pohjalta voidaan luoda uusi tietokanta E3:seen käyttäen VBS-komentosarjoja. Tarkoitus oli sitten luoda ohjeita suunnittelijoille kuinka automatisointi käytännössä tehdään. Tämä helpottaa suunnittelutyötä jatkossa.

2 OPINNÄYTETYÖN TOIMEKSIANTAJA

2.1 Pronor Group

Pronor Group toimittaa sähköautomaatiojärjestelmiä saha-, ajoneuvo- ja elintarviketeollisuudelle. 80 % asiakkaista on sahateollisuudessa. Pronor Groupin jäseniä ovat Pronor Engineering Oy, Pronor Automation Oy, Pronor Control Oy ja Pronor Service Tmi (ks. kuvio 1). Yritykset ovat rinnakkaisia, itsenäisiä yrityksiä, jotka kukin tekevät oman osa-alueensa sahojen automatisointiprojekteissa. (Burman 2012.)



Kuvio 1. Pronor Groupin yritykset

Pronor Engineering Oy

Pronor Engineeringin toimitusjohtaja on Kari Burman. Yritys työllistää toimitusjohtajan lisäksi yhden insinöörin. Pronor Engineering tekee sahojen automatisointiprojektien sähkösuunnittelun ja automaatiojärjestelmien dokumentoinnin. Yrityksen tuotanto valmistaa myös ajoneuvoihin toimitettaviin järjestelmiin johtosarjoja ja kenttäkoteloita. Burman perusti yrityksen vuonna 1995 ja aloitti toiminnan tuolloin sahojen sähköautomaatioasennusurakoitsijana. Myöhemmin yritys on laajentanut työkuvaansa automaatiojärjestelmien suunnitteluun. Se on myös vaihtanut nimeään ja kokenut organisaatiomuutoksia.

Muut Pronor Groupin jäsenet ovat myöhemmin tulleet mukaan. Sahateollisuuden järjestelmiä toimittava Jartek omistaa 20 % Pronor Engineering Oy:n osakkeista. (Burman 2012.)

Pronor Automation Oy

Pronor Automation tekee sahojen automaatioasennukset. Yritys työllistää kolme henkilöä. Kari Burman toimii päällikkönä myös tässä yrityksessä. Ennen automaatioasennukset ja suunnittelu tehtiin samassa yrityksessä. Pronor Automation ja Pronor Engineering on hajautettu erilleen, että voidaan nähdä kuinka suuri osa kustannuksista tulee suunnittelusta ja kuinka paljon asennuksista. (Burman 2012.)

Pronor Control Oy

Pronor Control toimii Lahdessa ja vastaa ohjelmoinnista. Se työllistää kuusi henkilöä ja toimitusjohtajana toimii Raimo Seppänen. Yrityksessä tehdään sahojen automaatiojärjestelmien logiikka-, valvomo- ja käyttöliittymäohjelmointi. Pronor Control tekee myös ohjelmistoja, jotka tuottavat tietoa sahojen hallintoa ja markkinointia varten. Yritys käyttää ohjelmointiin Siemensin S5 ja S7, Visual Basic, sekä C-ohjelmointikieliä. (Burman 2012.)

Pronor Service Tmi

Pronor Service Tmi Juha Kokko tekee toimitettuihin järjestelmiin elinkaarihuoltoa. Palveluihin kuuluu laitteistojen huolto ja päivitykset. (Burman 2012.)

Markkinointi

Pronor Groupin asiakkaista puolet on ulkomaisia ja puolet kotimaisia. Venäjä on yksi merkittävimmistä ulkomaisista asiakkaista. Pronor Group markkinoi järjestelmiään ottamalla suoraan yhteyttä asiakkaisiin puhelimitse ja sähköpostilla. Asiakkaalle saatetaan luoda yksilöllisesti mainos, jossa markkinoidaan ratkaisua heidän tarpeisiinsa. Joskus yhteistyökumppaneiden kautta saadaan jotakin tietoa mahdollisten asiakkaiden tarpeista, joihin pystytään sitten tarjoamaan ratkaisua.

Tapaamiset yhteistyökumppaneiden, kuten sahalaitosten komponenttien toimittajien kanssa, ovat erittäin tärkeitä myynnin ja markkinoinnin kannalta. Toisinaan myös asiakkaat voivat ottaa yritykseen päin yhteyttä ja pyytää ratkaisua ongelmaansa. (Burman 2012.)

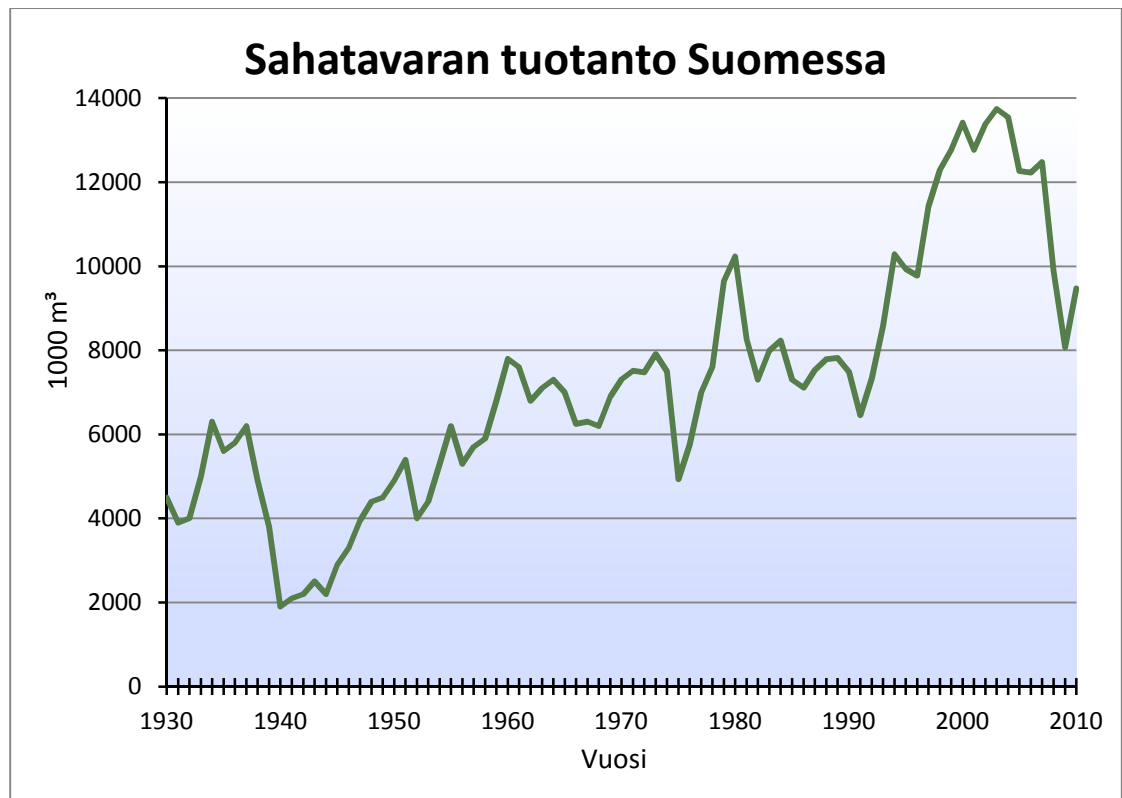
2.2 Sahateollisuus

Sahateollisuudessa sahataan ja höylätään tukkipuita. Metsästä kaadetut puut viedään sahalle, jossa jalostetaan halutuiksi tuotteiksi. Sahateollisuuden perustuotteita ovat hirret, laudat ja listat. Sahatavaratuotteita käytetään rakennusmateriaalina ja niistä voidaan jatkojalostaa huonekalujen osia, liimalevyjä, sisustus- ja ulkoverhouspaneeleja, ikkunoiden ja ovien karmeja, runkotolppia, kattotuoleja ja muita rakenteita.

Raakapuusta sahatavaraa tulee 45–50 %. Sivutuotteena sahoilta tulee haketta (28–32 %), purua (10–15 %) ja kuorta (10–12 %) (Sipi 1998, 23). Sivutuotteita käytetään massa- ja paperiteollisuudessa, kuitu- ja lastulevyn valmistamiseen ja energiantuotannossa (Paloheimo 2000, 83–89). Sivutuotteista voidaan tehdä pellettejä, joita käytetään kotitalouksien ja suurempien energiantuotantolaitosten polttoaineena.

Sahat jaetaan tuotantokapasiteetin mukaan piensahoihin ja teollisuussahoihin. Piensahalla tarkoitetaan laitosta, joka tuottaa alle 10 000 m³ sahatavaraa vuodessa. Piensahat ovat usein maatiloilla toimivia pieniä kiinteitä laitoksia tai ajoneuvon perässä hinattavia siirreltäviä rahtisahureita. Piensahoilla toiminta on useimmiten sivutoimista. Toiminnassa olevia piensahoja oli vuonna 2010 1206 kappaletta. Toimintakuntoisia piensahalaitoksia oli kaikkiaan olemassa n. 1600, mutta osa niistä ei ole enää käytössä. Piensahojen lukumäärä on vähentynyt murto-osaan 50 vuoden aikana, mutta sahatavaran tuotanto on kuitenkin lisääntynyt (ks. kuvio 2). Sahojen

tuotanto on siis selvästi siirtynyt enemmän isoihin teollisuussahoihin. (Torvelainen 2012, 1–4.)



Kuvio 2. Sahatavaran tuotanto Suomessa (Sipi 1998, 14; Metsätilastollinen vuosikirja 2011, 328, tieto yhdistetty molemmista lähteistä.)

Teollisuussahalla tarkoitetaan laitoista, joka tuottaa yli 10 000 m³ sahatavaraa vuodessa. Teollisuussahoja on Suomessa noin 70 kappaletta (Metsäteollisuus ry rekisteröityneet jäsenet 2012; Suomensahat ry jäsenet 2012; The Sawmill Database 2012.) Vuonna 2010 Suomen sahatavaran tuotanto oli yhteensä 9 473 000 m³. Teollisuussahojen osuus tuotannosta oli silloin 97 % ja piensahojen 3 %. (Metsätilastollinen vuosikirja 2011, 327.)

Puun matka metsästä valmiiksi sahatavaraksi alkaa puun hankinnalla.

Sahateollisuuden raakapuusta 60–70 % tulee yksityismetsistä. Puukauppa tehdään

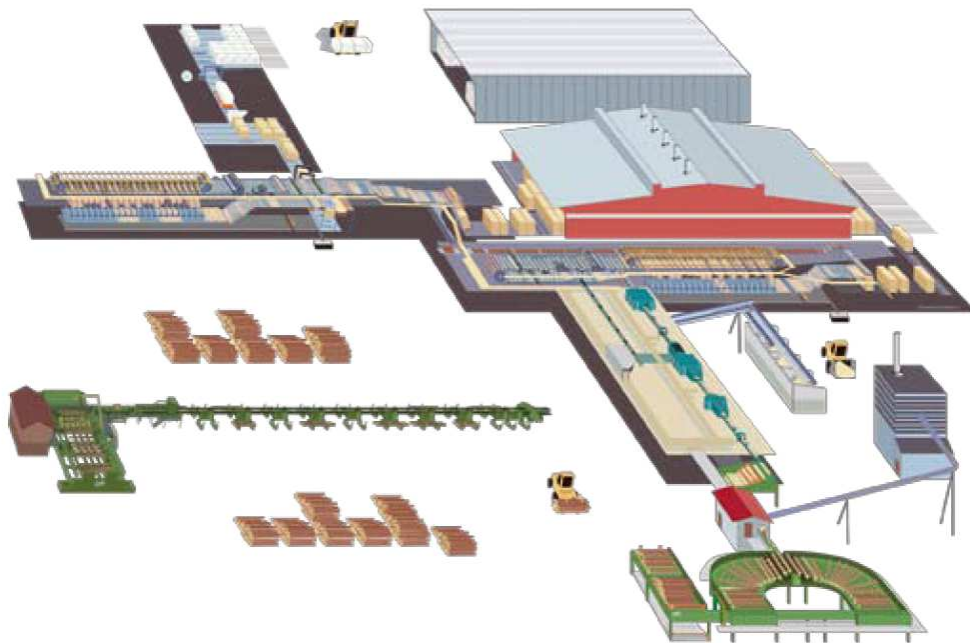
joko pysty- tai hankintakauppana. Pystykaupassa puun ostaja vastaa puiden hakkuista, lajittelusta ja kuljetuksesta. Hankintakaupassa puun myyjä korjaa itse puun ja toimittaa sen kuljetusreitinvarteen. Pystykauppa on nykyään yleisempää, koska puun korjuu on koneistunut ja yritykset haluavat vaikuttaa itse vaikuttaa puun korjuun laatuun ja toimitusaikoihin. Korjattu puu mitataan, ja sen perusteella muodostuu maksu metsän omistajalle. Puut voidaan mitata, kun ne ovat vielä pystyssä, hakkuun aikana hakkuukoneella tai kuljetuksen jälkeen kaukokuljetusvarastossa. Nykyään mittaukseen käytetään enemmän automatisoitua tehdasmittausta, jolla se voidaan tehdä tarkasti ja tehokkaasti. Tehdasmittauksessa voidaan käyttää esimerkiksi konenäköön tai infrapuna- ja lasersäteillä mittaamiseen perustuvaa järjestelmää. Tukin liikkuessa kuljettimella mittausta voidaan tehdä muutamien senttimetrin välein, joten tietokone saa hyvin tarkan tiedon puunrungon profiilista. Tätä tietoa voidaan käyttää apuna puun laatulajittelussa. (Sipi 1998, 25–60.)

Sahalla tukit ensin kuoritaan. Kuori täytyy ottaa pois, ettei sitä joudu sahausken yhteydessä syntyvän hakkeen joukkoon. Massa- ja paperiteollisuus eivät voi käyttää haketta jossa on kuorta seassa. Kuorinta pyritään tekemään siten, että mahdollisimman vähän puuta tukin rungosta menee hukkaan. Sahalinjalla tukeista tehdään halutun muotoisia lopputuotteita. Sahaus voidaan tehdä kehäsahoilla, pyörösahoilla, vannesahoilla, pelkkahakkureilla tai näiden yhdistelmillä. Kehäsahat rupeavat olemaan jo vanhentunutta tekniikkaa, eikä niitä juuri enää käytetä. (Sipi 1998, 61–97.)

Sahauksen jälkeen sahatavara viedään kuivaamoon, jossa siitä poistetaan ylimääräinen kosteus. Suomessa kuivaamo on yleensä suljettu halli, jossa on lämmitys ja koneellinen ilmankierto. Polttoaineena kuivaamossa käytetään sahalla sivutuotteena syntyvää puunkuorta. Kuivaus parantaa puun säilyvyyttä, koska esim. sinistäjä ja homesienet eivät kasva kuivassa puussa. Kosteus vaikuttaa myös puun

lujuus- ja jäykkyyssominaisuuksiin, työstettävyyteen, liimattavuuteen, kyllästettävyyteen, pintakäsittelyominaisuuksiin ja sähkönjohtokykyyn. Kuivaamossa sahatavara pinotaan nippuihin, joissa kerrosten välissä on poikkittaisia rimoja. Tämä takaa tasaisen ilmankierron sahatavaran välissä. Kuivatessa sahatavaraan saattaa tulla vikoja, kuten halkeamia ja vääntymiä. Pinoamalla niput tasaisesti ja laittamalla poikkirimoja tarpeeksi tiheästi voidaan välttyä muodonmuutoksilta. Oikea ilman kiertonopeus, ilman suhteellinen kosteus ja lämpötila vähentävät myös kuivausvikojen syntymistä. (Sipi 1998, 106–130.)

Kuivauksen jälkeen sahatavara katkaistaan lopulliseen pituuteensa ja lajitellaan laaturyhmiiin. Laaturyhmät määräytyvät lujuuden ja ulkonäön perusteella. Laatuun vaikuttavat oksat, halkeamat, sienivauriot, väri, muoto ja monet muut viat. Jokaisella maalla on oma sahatavaran lajittelusäännöstönsä. Ulkonäköön perustuva lajittelu on nykyisin toteutettu monesti konenäköjärjestelmällä. Lujuutta voidaan testata taivuttamalla kappaletta koneellisesti. Myös ulkonäöstä voidaan päätellä sahatavaran lujuutta, kuten katsomalla oksanreikien, halkeamien ja muiden vaurioiden määrää. Sahatavara leimataan sahan yksilöllisellä leimalla päätyyn, josta ilmenee laatu ja sahan nimi. Seuraavaksi sahatavara voidaan pakata ja laittaa varastoon odottamaan kuljetusta. Teollisuussahalinjastosta on kuva kuviossa 3. (Sipi 1998, 131–156.)



Kuvio 3. Teollisuussahalinjasto (Heinola Sawmill Solutions yleisesittely n.d.)

3 SÄHKÖSUUNNITTELUN AUTOMATISOINTI

3.1 Automaation käsite

Automaatio tarkoittaa itsestään, ilman ohjausta, tapahtuvaa tai toimivaa. Automaatio voidaan jakaa useampiin käsitteisiin, joihin kuuluvat automaatiotekniikka ja tuotantoautomaatio. Automaatiotekniikalla ymmärretään laitteen sisäisiä, elektronisia toimintoja. Tuotantoautomaatio voidaan jakaa koneautomaatioon, josta käytetään myös nimitystä mekatroniikka tai kappaletavara-automaatio, sekä prosessiautomaatioon. Koneautomaatiolla käsitetään fyysisiä laitteita, kuten kuljettimia, robotteja ja muita teollisuuden toimilaitteita, joita ohjataan

elektronisella järjestelmällä. Prosessiautomaatiossa käsitellään nestemäisiä ja kaasumaisia aineita, jotka kulkevat putkissa ja säiliöissä pumpun voimalla. Prosessiautomaatiossa säädetään usein aineiden lämpötilaa ja pinnan korkeuksia säiliöissä. Tämä työ kuuluu automaatiotekniikan käsitteen piiriin, koska siinä ei käsitellä fyysisiä kappaleita tai aineita, vaan ohjelmoidaan komentosarjoja. (Keinänen, Kärkkäinen, Lähetkangas & Sumujärvi 2010, 7–8.)

3.2 Suunnittelun automatisoinnin edut ja ongelmat

Sähkösuunnittelun automatisointi helpottaa toistuvien dokumenttien tuottamista ja vähentää virheitä niissä. Ennen jokainen suunnittelun dokumentti kuten piirikaavio, kaapeliluettelo, kytkentätaulukko ja tasopiirustus, piti tehdä yksitellen. Tietokantapohjaisessa suunnittelussa suurin osa dokumenteista luodaan automaattisesti, ja se nopeuttaa työtä. Eri tehtävissä toimivat henkilöt tarvitsevat eri dokumentteja. Pääkaavio on sähkösuunnittelun perusdokumentti, mutta vikaa etsivä huoltomies tarvitsee piirikaaviota. Kenttäkotelon kokoamiseen tarvitaan tasopiirustusta ja automaatioväylien asentamiseen tarvitaan väyläkuva. Laitteita hankittaessa tarvitaan laite- ja kaapeliluettelot, jotta niitä voidaan tilata oikea määrä. Kaikissa näissä dokumenteissa on paljon samoja asioita, mutta hieman eri muodossa. Tietokantapohjaisen sähkösuunnitteluohjelmiston avulla pystyy tuottamaan helposti kaikki tarvittavat dokumentit.

Tietokantapohjainen sähkösuunnittelu helpottaa järjestelmän ylläpitoa. Tehdyt muutokset yhteen dokumenttiin päivittyvät myös kaikkiin muihin. Ohjelmiston automaatio pitää huolta, että kaikki muutokset tapahtuvat luotettavasti. Tämä vähentää virheitä ja aikaa verrattuna käsin tehtyihin muutoksiin. Muutokset tapahtuvat kaksisuuntaisesti (ks. kuvio 4). Tämä tarkoittaa sitä, että esimerkiksi pääkaavion komponenttien sijaintitiedot on alun perin luotu

projektinmäärittelytiedoston pohjalta. Jos komponenttien sijaintitietoja vaihdetaan pääkaavioon, ne muuttuvat myös alkuperäiseen projektinmäärittelytiedostoon.



Kuvio 4. Muutokset dokumentteihin tapahtuvat kaksisuuntaisesti

Suunnittelun automatisoinnin huonona puolena on joustamattomuus. Ohjelmisto tuottaa dokumentit aina samalla tavalla eikä tee koskaan poikkeuksia. Toisaalta ohjelma on aivan yhtä hyvä kuin sen ohjelmointi. Mahdollisia poikkeuksia pystytään ohjelmoimaan vaikka loputtomiin. Kaikki voidaan automatisoida. Automatisoinnissa kannattaa kuitenkin ottaa huomioon se, että sen pitää ylittää kriittinen sarjaluku, jotta sitä kannattaa lähteä toteuttamaan. Tämä tarkoittaa sitä, että automatisointi on aikaa vievää, mutta jos toiminto, joka toistuu tarpeeksi monta kertaa, automatisoidaan, säästetään aikaa. Toisin sanoen on järkevintä automatisoida vain

yleisimmät toiminnot ja jos dokumentteihin tarvitsee tehdä poikkeuksia, ne kannattaa tehdä käsin.

Toinen huono puoli suunnittelun automatisoinnissa on systemaattisen virheen mahdollisuus. Tämä tarkoittaa sitä, että jos automatisoinnin ohjelmoinnissa on virhe, se toistuu kaikissa toiminnoissa, johon kyseinen osa ohjelmasta vaikuttaa. Esimerkiksi jos jokin tieto on tallennettu väärään muuttujaan, niin kaikki kohtiin, joihin muuttuja tulostaa tekstiä, tulee silloin väärää tietoa. Suunnitteluautomaatiolla tuotetut dokumentit kannattaa tarkistaa aina, kun ohjelmoinnin logiikkaan on tehty muutoksia, jotta kaikki tieto on siellä oikealla paikallaan.

3.3 Komentosarjat

Komentosarjat ovat lista peräkkäisiä komentoja, jotka toteutetaan yhdellä kertaa.

Niiden käyttäminen nopeuttaa usein toistuvia peräkkäisiä toimintoja.

Komentosarjoihin voidaan tehdä myös valintarakenteita, jolloin niiden kirjoittaminen muistuttaa ohjelmointia. Tässä opinnäytetyössä tietojen tuominen projektinmäärittelytiedostosta E3-suunnitteluohjelmaan on toteutettu komentosarjana. Kun komentosarja ajetaan, tietokone tulkitsee komentoja ja alkaa toteuttaa niitä.

3.4 E3 Cable -sähkösuunnitteluohjelmisto

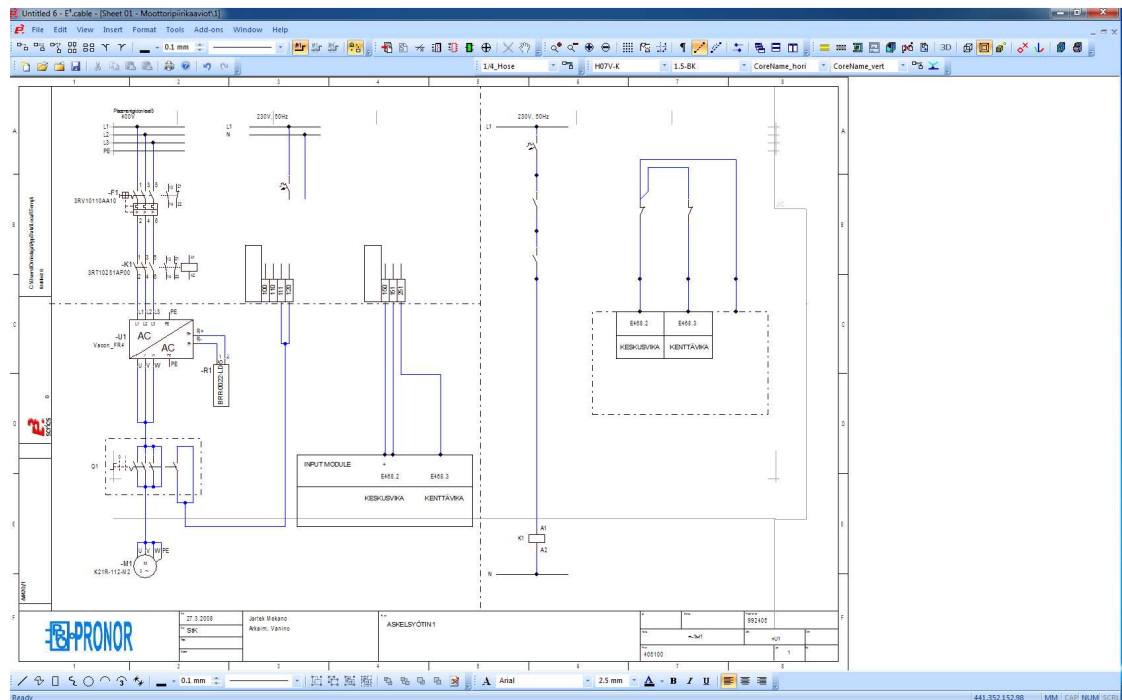
3.4.1 Esittely

E3 Cable on elektroniikka- ja sähkösuunnitteluohjelmisto, joka on tarkoitettu nimenomaan piirikaavioiden ja johtosarjojen suunnitteluun. E3 Cable kuuluu ohjelmistosarjaan nimeltä E3 series, johon kuuluu monta erilaista ohjelmistoa, jotka soveltuvat moniin sähkösuunnittelua sisältävien toimialojen tarpeisiin. Sarjan

ohjelmistoja käytetään monilla toimialoilla, kuten energiatekniikassa ja ajoneuvoteollisuudessa. Pronor Engineeringillä on käytössä E3 Cable. Ohjelman on kehittänyt japanilainen yritys nimeltään Zuken. Ohjelma sisältää valmiita piirrosmerkkejä, joiden avulla piirikaavioiden tekeminen on helppoa. Ohjelma on tietokantapohjainen, ja se muodostaa uuden työn alkaessa tiedoston, jota kutsutaan projektiksi. Projektista on helppo tuottaa kaapeliluettelot, kytkentätaulukot, piirikaaviot ja muut dokumentit. Projektiin tehtävät muutokset päivittyvät kaikkiin dokumentteihin.

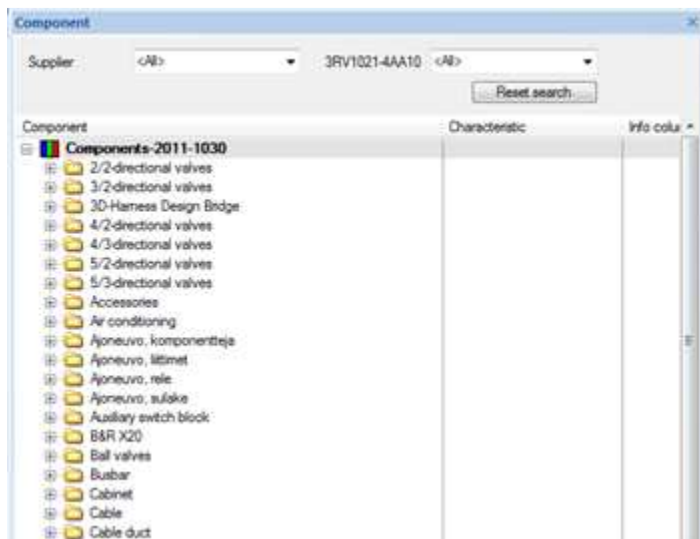
3.4.2 Ohjelman käyttäminen

Tärkein ikkuna E3:ssa on piirtoalue (ks. kuvio 5). Piirtoalueella on ruudukko, joka helpottaa piirrosmerkkien asettamista ja johtimien vetämistä. Ruudukon kokoa voi säätää asetuksista tai sen voi asettaa näkymättömäksi. Näkymän suurentaminen ja pienentäminen tapahtuu pitämällä pohjassa Ctrl-näppäintä ja pyörittämällä hiiren rullapainiketta. Panorointi tapahtuu näppäinyhdistelmällä Shift ja hiiren keskipainike ja liikuttamalla hiirtä.



Kuvio 5. E3:n piirtoalue

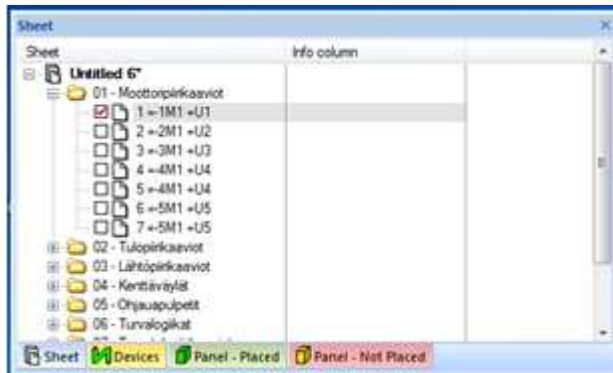
Piirrosmerkit saa raahattua piirtoalueelle komponentti-ikkunasta. Komponentti-ikkunassa on symboleita ja laitteita. Symbolit ovat pelkkiä piirrosmerkkejä. Laitteet ovat laitevalmistajien todellisia olemassa olevia tuotteita. Laitteet sisältävät tietoja komponentista, sen piirrosmerkin ja tasopiirustuksen fyysisen laitteen etupuolelta. Komponentti-ikkunasta löytyy hyvä valikoima piirrosmerkkejä ja laitteita, jotka on siististi laitettu kategorioihin (ks. kuvio 6). Kun komponentteja selataan, ohjelman esikatseluikkunassa näkyy jokaisen komponentin piirrosmerkki. Tämä on hyvä toiminto, koska sen avulla saa nopeasti yleiskuvan saatavilla olevista piirrosmerkeistä. Jos valikoimasta ei löydy tarvittavaa laitetta, sen voi tehdä muokkaamalla olemassa olevia laitteita tai tekemällä sen itse alusta alkaen.



Kuvio 6. Komponentti-ikkuna

Kun laitteet on asetettu piirtoalueelle, ohjelmassa on toiminto, joka yhdistää automaattisesti kaikki vaaka- tai pystysuunnassa olevat johtimet piirustuksessa. Tämä nopeuttaa mukavasti työskentelyä. Johtimia voi tietenkin piirtää myös käsin, jos on tarvetta tehdä niihin mutkia. Painamalla hiiren oikeaa painiketta ja valitsemalla *properties*, pääsee katsomaan piirrosmerkin asetuksia. Sieltä voi vaihtaa ylemmän järjestelmän tunnusta, lähdön numeroa tai laitteen tunnusta ja voi katsoa fyysisen laitteen tasopiirustusta.

Kaikki tietokannassa olevat dokumentit on listattu omassa ikkunassaan (ks. kuvio 7). Tähän voi tehdä hakemistoja, jolloin eri asiakirjat saa järjestykseen. Samassa ikkunassa on välilehti, jossa on listattu kaikki yksittäiset laitteet projektissa. Toinen hyödyllinen ikkuna on viesti-ikkuna, jossa ohjelma tiedottaa omista sisäisistä toiminnoistaan.



Kuvio 7. Tietokannassa olevat dokumentit näkyvät omassa ikkunassa.

Kaikkien ikkunoiden kokoja voi säätää ja ne voi siirtää toiselle ruudulle, jos käytössä on kaksi monitoria. Tämä vapauttaa tilaa isommalle piirtoalueelle. Ohjelmassa on myös kielitietokanta, joka mahdollistaa dokumentin tekstien kääntämisen eri kielelle. Työn voi suunnitella omalla äidinkielellään ja sen voi vaihtaa valikosta toiselle kielelle, kun työ pitää lähettää ulkomaiselle asiakkaalle.

3.4.3 E3:n toimintojen automatisoiminen

Kaikki ohjelman toiminnot voidaan automatisoida Visual Basic Scripting Edition -komentosarjoina. Ohjeita automatisoinnin toteuttamiseen löytyy ohjelman omasta help-tiedostosta. Ohjeet ja luokkakirjaston saa näkyviin painamalla ohjelmassa *help*, *contents* ja avaamalla *scripting interface* -välilehden.

3.5 Ohjelmointirajapinta

3.5.1 Visual Basic Scripting Edition -komentosarjakieli

Opinnäytetyön käytännön toteutuksessa käytettiin Microsoftin kehittämää Visual Basic Scripting Edition -komentosarjakieltä, eli lyhyemmin VBScriptiä. Tätä ei tule sekoittaa Visual Basic -ohjelmointikielen, jonka ohjelmien tekemiseen käytetään tyypillisesti Visual Studio -ohjelmointiympäristöä. Visual Basicin lähdetiedosto on

*.vb-päätteinen eikä sitä voi suoraan ajaa. Ohjelmointiympäristö pystyy kuitenkin kääntämään sen tietokoneen prosessorin ymmärtämäksi konekieliseksi koodiksi, jolloin siitä tulee *.exe-päätteinen tiedosto, joka voidaan ajaa. VBScript-komentosarjoja taas kirjoitetaan yksinkertaisimmillaan muistiolla. Komennot kirjoitetaan peräkkäin ja tallennetaan *.vbs-päätteisenä. Sen jälkeen tiedosto voidaan ajaa sellaisenaan, ja Windows Script Host tulkitsee komennot ja toteuttaa ne. Molempien ohjelmointikielien syntaksi on lähellä toisiaan. VBScript-komentosarjat kannattaa tehdä kuitenkin tehdä esimerkiksi Notepad++ tai VBSEdit-nimisillä ohjelmilla, koska ne sisältävät paljon kirjoittamista helpottavia toimintoja.

3.5.2 WSH-komentosarjojen ajoympäristö

WSH on lyhennys sanoista Windows Script Host. Se on oletuksena asennettu Windows 98 käyttöjärjestelmään ja sitä myöhempisiin versioihin, ja se mahdollistaa komentosarjojen toimimisen. Parhaimmillaan se on, kun sitä käytetään automaattisen sisäänkirjautumisen toteuttamiseen, pääkäyttäjän toimintojen tekemiseen tai ohjelmien automatisointiin. (What is WSH? 2013.)

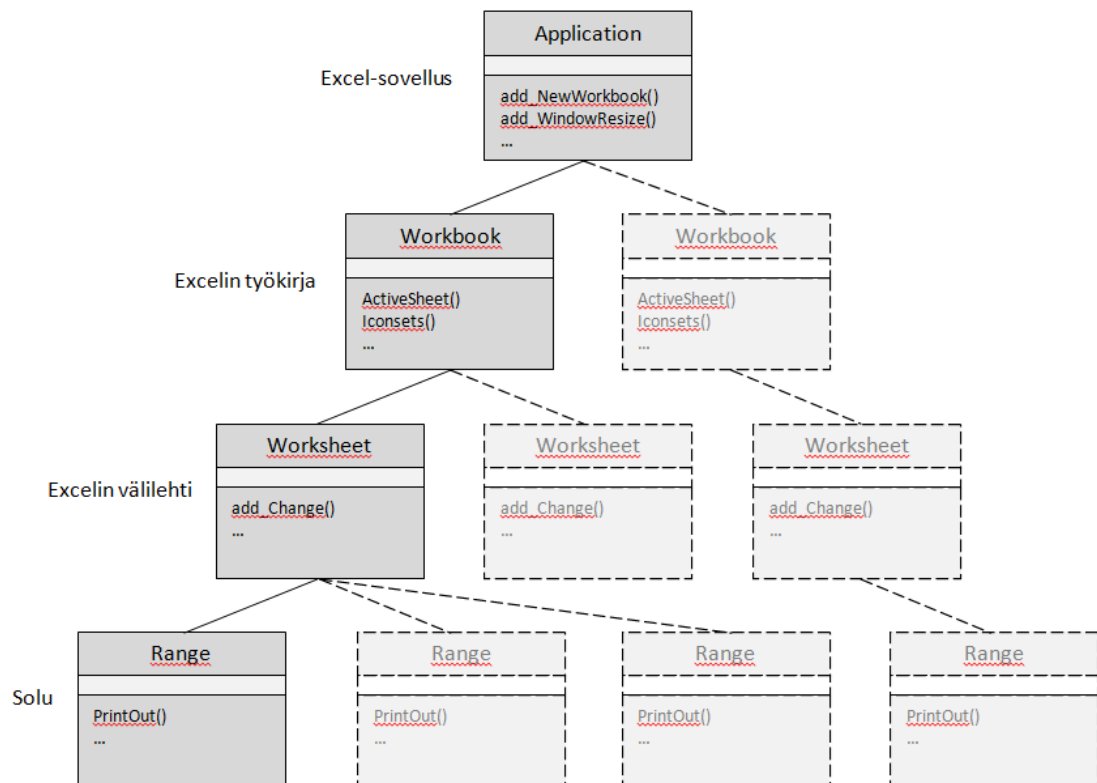
3.5.3 COM-komponenttimalli

COM-mallit on olio-ohjelmointipohjainen järjestelmä, jota käytetään ohjelmointikomponenttien luomiseen. COM-lyhennys tulee sanoista Component Object Model. COM-malli on binaarinen standardi ja laitteistoriippumaton. Etuna siinä on laajennettava rakenne, joka mahdollistaa hyvin aikaisemmin kirjoitetun koodin uudelleenkäyttämisen. (Component Object Model 2013.)

Opinnäytetyössä käytettyjen ohjelmien, Excelin ja E3:n rakenne noudattaa hierarkista luokkarakennetta. Rakenteen käsittämiseksi on syytä ymmärtää olio-ohjelmoinnin peruskäsitteitä, kuten luokka, olio, periytyminen, attribuutti ja metodi. Luokkaa voisi ajatella piparkakkumuottina, kun taas olio olisi pipari, joka on tehty

piparkakkumuotilla. Kun Excel käynnistetään, luodaan Application-luokkaan kuuluva Excel-sovellusolio. Näitä Excel-ohjelmia voidaan käynnistää vaikka useampi rinnakkain, jolloin ne kukin muodostavat itsenäisen olion, mutta ovat täysin samanlaisia, koska kuuluvat samaan luokkaan. Luokassa on määritelty attribuutteja, jotka ovat muuttujia, joihin voidaan tallentaa tietoa. Niillä on myös metodeja, eli toimintoja, joita oliot voivat tehdä. Application-luokassa on esimerkiksi metodi, jolla luodaan uusi työkirjaolio. Työkirjaolio voi taas luoda välilehtiolion, jossa on yksittäisiä solu oliota, joihin Excelissä tallennetaan tietoa. Nämä luokat periytyvät toisistaan, kuten kuviossa 8 on kuvattu. Välilehteä ei voi olla olemassa, jos ensin ei ole luotu työkirjaa.

Metodit on kuvattu luokkakirjastossa. Siellä on kerrottu, mitä ne tekevät, ja usein niistä on myös esimerkkikoodinpätkiä. Kun ohjelmoija haluaa toteuttaa jonkun toiminnon, hänen pitää etsiä sitä luokkakirjastosta. Myös E3 sisältää samantapaisen COM-komponenttimallin.



Kuvio 8. Excelin COM-komponenttimalli

3.6 Sähkösuunnittelu

3.6.1 Dokumenttityypit

Sähkö- ja automaatio-suunnittelun dokumentteja tarvitaan laitoksen tai laitteiston suunnittelussa, rakentamisessa ja asentamisessa, käyttöönotossa, käytössä, huoltamisessa ja laitteiston poistossa. Eri tehtävissä työskentelevät ihmiset tarvitsevat eri dokumentteja työnsä tekemiseen. (Mäkinen, Kallio & Tantarimäki 2009, 240.)

Tasopiirustus on mittakaavaan tehty piirustus laitoksen tai laitteiston osista. Layout-kuva on toinen vakiintunut nimitys tasopiirustukselle. Esimerkiksi asentaja tarvitsee kenttäkotelon tasopiirustusta kootessaan sitä. Tasopiirustukseen on syytä tehdä

törmäystarkastelu, jossa varmistetaan, että laitteet, tasot ja putket eivät törmää toisiinsa. Törmäystarkastelua tehdessä tulee huomioida myös, että luukut ja ovet pystytään todellisuudessa sulkemaan ja liittimien kiinnittämiseksi jää tarpeeksi tilaa. 3D-CAD-suunnittelusta on etua törmäystarkastelun tekemiseen 2D-suunnitteluun nähden. (Mäkinen ym. 2009, 241.)

Laiteluettelot ovat listoja järjestelmissä käytettävistä laitteista. Niissä on kerrottu laitteiden positio, tyyppi ja valmistaja. Lisäksi niissä on kuvattu tapa kuinka laitteet on liitetty automaatiojärjestelmään ja niissä on merkintöjä mittausalueista sekä hälytysrajoista. Automaatiosuunnittelijat tarvitsevat laiteluetteloa piirikaavion suunnitteluun ja automaatiojärjestelmän ohjelmointiin. (Mäkinen ym. 2009, 241–242.)

Kaapeliluettelossa on kerrottu kaapeleiden tunnuksot, tyyppi, poikkipinta-ala ja kaapelin loppu- ja alkupää. Asentaja tarvitsee kaapeliluetteloita vetäessään kaapeleita. (Mäkinen ym. 2009, 251.)

Väyläkuvassa on esitetty, kuinka eri laitteet on kytketty automaatiojärjestelmään. Väyläkuvaa tarvitaan järjestelmää asennettaessa.

Instrumentoinnin asennustyyppiirustuksessa esitetään laitteiden mekaaninen kytkeminen. Asennustyyppiirustuksia voidaan laatia sekä sähköisistä että pneumaattisista ja hydraulisista laitteistoista. Asentajat tarvitsevat asennustyyppiirustuksia kokoonpanossa. (Mäkinen ym. 2009, 251.)

Sekvenssikaavio on automaatio suunnittelun dokumentti, jossa on kuvattu suoritusjärjestys kuinka peräkkäiset toiminnot suoritetaan. Sitä voidaan kutsua myös toimintadiagrammiksi. Automaatiosuunnittelija käyttää sekvenssikaaviota ohjelmoimiseen. Standardissa SFS-848 on annettu ohjeita sekvenssikaavion laatimiseen. (Mäkinen ym. 2009, 248.)

Logiikkaavio on toiminnallinen kuvaus automaatiojärjestelmän toiminnasta. Toiminta on esitetty siinä Boolean logiikalla eli symboleilla, jotka kuvaavat ja, tai, ei ja rs-kiikun toimintoja. Logiikkakaaviota käytetään myös automaation ohjelmointiin.

Lukituskaaviossa käytetään samanlaista esitysmuotoa, ja se kuvaa prosessilaitteiston lukitustoimintoja. (Mäkinen ym. 2009, 248.)

Säätökaavio on kuvaus säätöpiirin toiminnasta. Säätöpiireillä säädetään esimerkiksi säiliöön sisään ja ulos virtaavan nesteen nopeutta siten, että pinnankorkeus pysyy halutulla tasolla, tai sen lämpötilaa. Säätökaavioissa on esitetty säätötekniikan symbolien avulla säätöön liittyvät rajoitukset, lukitukset ja asetusarvojen muodostuminen. Automaatiosuunnittelija tarvitsee säätökaavioita automaation ohjelmoinnissa. (Mäkinen ym. 2009, 249.)

Sähköteholaskemassa on laskettu järjestelmän laitteiden tehot yhteen ja selvitetty induktiivisen ja kapasitiivisen kuorman suhde resistiiviseen kuormaan.

Sähköteholaskelmaa tarvitaan muuntajan, loistehon kompensoinnin ja kaapelien mitoittamiseen. Sähkösuunnittelija ja keskusvalmistaja tarvitsevat näitä tietoja. (Mäkinen ym. 2009, 243.)

3.6.2 Standardit

Sähkösuunnittelun dokumenttien piirtämiseen liittyvät säännöt perustuvat standardeihin. Standardisointiorganisaatioita on olemassa kolmea tasoa: maailmanlaajuinen, alueellinen ja kansallinen taso. Standardien tarkoitus on yhtenäistää toimintatapoja ja vähentää yhteensopivuusongelmia tekniikassa. Paperien, kirjekuorien, vaatteiden, kenkien, ruuvien ja muttereiden koot sekä monet muut arkipäiväisten asioiden mitat perustuvat standardeihin. (Pere & Kivimäki 1998, 5; Suomen standardoimisliiton SFS ry:n kotisivut n.d.)

Standardeja saa käyttää ja hyödyntää kuka tahansa vapaasti. Niiden hankkiminen on kuitenkin maksullista. Suomen standardoimislautakunta rahoittaa toimintaansa tällä tavalla. Standardien noudattaminen on vapaaehtoista. Jos kuitenkin jonkun tuotteen valmistaja päättää noudattaa standardia, hän voi hakea sille sertifikaattia.

Sertifikaatti on osoitus siitä, että tuote täyttää standardien vaatimukset. Sertifikaattia voi hakea akkreditoituilta sertifiointielimiltä. (Suomen standardoimisliiton SFS ry:n kotisivut n.d.)

ISO (International Organisation for Standardization) on kansainvälinen standardisointijärjestö. CEN (Comité Européen de Normalisation) on eurooppalainen standardisoimisjärjestö. SFS ry (Suomen standardisoimislautakunta Finlands Standardiserungskommission) on Suomen standardisoimisliitto ja standardisoimisen keskusjärjestö Suomessa. (Suomen standardoimisliiton SFS ry:n kotisivut n.d.)

ISOon kuuluu noin 200 eri alojen teknistä komiteaa. Esimerkiksi teletekniikalla ja lentokoneiteollisuudella on omat standardisointiorganisaationsa. Tärkein ISO:n yhteistyökumppani on IEC (International Electrotechnical Commission), joka on kansainvälinen elektroniikka- ja sähköalan standardisoimisjärjestö. CENELEC (Comité Européen de Normaliation Electrotechnique) on eurooppalainen elektroniikka- ja sähköalan standardisoimisjärjestö. SESKO ry on suomalainen elektroniikka- ja sähköalan standardisoimisjärjestö. SESKO on SFS ry:n toimialayhteisö. Kuviossa 9 on kuvattuna yleisiä ja sähköalan standardisointiorganisaatioita. (Suomen standardoimisliiton SFS ry:n kotisivut n.d.)

	Sähkö- ja Elektroniikka	Yleinen
Kansainvälinen	IEC	ISO
Eurooppalainen	CENELEC	CEN
Suomalainen	SESKO	SFS

Kuvio 9. Standardisointiorganisaatiot

ISO-standardit merkitään tunnuksella ISO, CENin standardit tunnuksella EN ja Suomessa hyväksytyt standardit SFS-tunnuksella. Esimerkiksi standardi, jossa kerrotaan graafisten symbolien käytöstä teknisillä aloilla, on SFS-EN ISO 81714-1:1999. Tunnukset on yhdistetty standardin nimeen, koska se on hyväksytty Suomessa, Euroopassa ja kansainvälisesti. Suomalaiset standardit on tehty eurooppalaisten standardien pohjalta, joista taas n. 30 % on tehty kansainvälisten standardien pohjalta. Myös muissa maissa on kansallisia standardisointiorganisaatioita. Saksassa vastaava merkintä on DIN-EN ja Ruotsissa SS-EN. (Suomen standardoimisliiton SFS ry:n kotisivut n.d.)

3.6.3 Sähkösuunnitteluun liittyvät ohjeet

Standardissa SFS 501 on ohjeita piirikaavioissa käytettävistä piirrosmerkeistä.

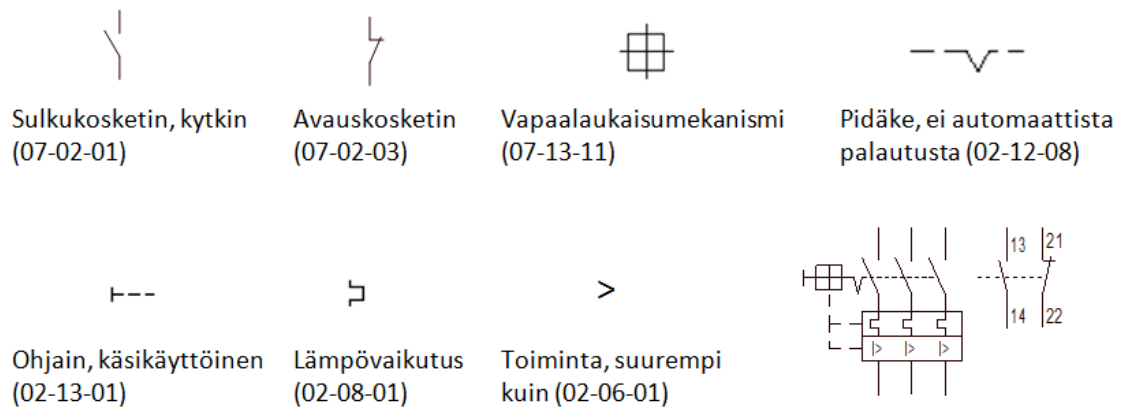
Standardi sisältää suomenkielisen käännöksen IEC 60617-standardissa hyväksytyistä sähköpiirrosmerkeistä. Lisäksi se sisältää SFS-EN ISO 81714-1:1999-standardin, jossa on kerrottu yleisiä ohjeita, mitä tekniseen sähköpiirustukseen liittyy.

Sähkösuunnittelun dokumentteja koskevat samat yleisohjeet jotka on hyväksytty muissa teknisen dokumentoinnin standardeissa. Yleisohjeissa on määritelty piirustusohjeiden kokoja, viivanleveyksiä, viivatyyppejä, fonttikokoja ja piirrosmerkkien mittasuhteita. Näistä on kerrottu standardeissa SFS-ISO 3098/1 (1989/01), SFS-EN ISO 3098-3 (2000/12), SFS-EN ISO 3098-4 (2000/12), SFS-ISO 128-23 (2009/09), SFS-ISO 129-1 (2009/09). Myös hyväksytyjä matemaattisia merkkejä ja tietotekniikan standardeissa hyväksytyjä merkintätapoja voidaan käyttää. (SFS 501 2004; Valtanen 2012, 519.)

Käytettävien piirrosmerkkien tulee olla yksinkertaisia, jotta niiden hahmottaminen ja piirtäminen olisi mahdollisimman helppoa. Piirrosmerkin tulee olla yhdistettävissä hyvin kuvaamaansa asiaan. Silloin piirrosmerkin merkitys on itsestään selvä ja helppo muistaa.

Automaattisesti palautuvat liikkuvat osat, kuten releet ja jousipalautteiset napit, pitää esittää aina lepotilassa eli releet siinä tilassa, jossa niihin ei vaikuta ohjausjännite, ja jousipalautteinen nappi yläasennossa. Osat, joissa ei ole automaattista palautusta, kuten venttiilit ja kaksiasentoiset sähkökytkimet, esitetään ei-aktiivisessa tilassa. Sähkökytkimet esitetään auki ja venttiilit kiinni. Kummassakin tapauksessa siis sähkön tai veden kulkeminen on estetty.

Piirrosmerkkejä voidaan luoda yhdistämällä muita standardista löytyviä symboleja yhdeksi kokonaisuudeksi (ks. kuvio 10). Jokainen yksittäinen symboli tuo lisäinformaatiota laitteesta ja kuvailee paremmin kuinka se toimii.



Kuvio 10. Moottorisuojakatkaisimen symboli on muodostettu yhdistämällä muita standardista löytyviä symboleja.

3.7 Moottoripiirikaaviot

3.7.1 Moottoripiirikaavion osat

Pääkaavio

Pääkaavio on sähkösuunnittelun perusdokumentti. Se on yleiskuva, josta näkee piirin pääkomponentit, niiden lukumäärän, pääjännitteen jakelun sekä käynnistys- ja suojalaitteiden sijainnin. Pääkaaviota tarvitaan sulakkeita vaihdettaessa ja keskuksen varatiloja arvioitaessa. Siihen merkitään pääjännite, taajuus, kaapeleiden tyypit, koskettimien numerointi, kulutuskojeiden sähkötehot, virrat ja esimerkiksi sähkömoottorien pyörimisnopeudet. Tässä työssä piirikaavioihin on merkitty myös ohjelmoitavien logiikoiden osoitteita, ryhmäkilpien tiedot, turvapiirin numero ja riviliittimien numerointi. Pääkaavioissa on erotettu katkoviivalla komponentit, jotka

kuuluvat sähkökeskukseen, ja kentällä olevat laitteet. Monesti myös samalla paperilla on esitetty pääkaavio ja ohjauspiirikaavio, jotka on erotettu toisistaan katkoviivalla. (Mäkinen ym. 2009, 245.)

Teollisuussaha on tyypillisesti jaettu useaan erilliseen turvapiiriin. Turvapiirit on merkitty laitoksen tasopiirustukseen. Turvapiirit mahdollistavat linjaston osan sulkemisen turvallisesti lyhyeksi aikaa, esimerkiksi huoltotöiden tai kuljettimelta pois lähteneen tukin oikaisua varten, ilman että koko laitosta täytyy sulkea. Samaan aikaan kun yksi osa linjastosta seisoo, sitä edellisessä turvapiirissä sahatavaraa alkaa varastoitua reserviin. Jos reservit täyttyvät, automaatiojärjestelmän anturit havaitsevat sen ja pysäyttävät linjaston. Näin tuotanto jatkuu sahan muissa osissa, vaikka yksi turvapiiri on otettu pois päältä. (Burman 2012.)

Ohjauspiirikaavio

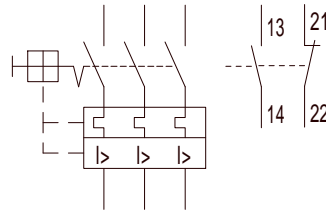
Ohjauspiirikaaviossa on esitetty samoja asioita kuin pääkaaviossa, mutta eri tavalla. Se on yhden piirin toimintaselostus, jossa esitetään moottorin tai muun laitteen käynnistin ja ohjaustapa. Ohjauspiirikaavio on pääkaaviota parempi esitystapa, kun etsitään vikoja. Usein ohjauspiirikaavio on esitetty omana sivunaan, mutta tätä työtä tehdessä sijoitin pääkaavion ja ohjauspiirikaavion vierekkäin samalle sivulle katkoviivalla erotettuna, jos ne mahtuivat. (Mäkinen ym. 2009, 246.)

3.7.2 Moottoripiirien komponentit

Moottorisuojakatkaisin

Moottorisuojakatkaisimia käytetään laitteiston suojaamiseen liialta virralta ja virtapiirin avaamiseen ja sulkemiseen. Niissä on kaksi mekanismia, jotka pystyvät katkaisemaan virran. Ensimmäinen on terminen laukaisija, joka suojaa laitteistoa ylikuormitustilanteessa. Siinä on bi-metalli-liuskan päälle kierretty johdin, jossa kulkeva virta lämmittää liuskaa. Tämä saa liuskan taipumaan ja katkaisee virtapiirin, jos virta kasvaa liian suureksi. Termisen laukaisijan toiminta-aluetta voidaan säätää

asetusruuvista. Toinen mekanismi on magneettinen pikalaukaisija, joka katkaisee virtapiirin oikosulkuilanteessa. Siinä on rele ja jos virtapiirissä on yhtäkkiä suuri virta, rele toimii ja sen kosketin katkaisee virran. Moottorisuojakatkaisimet pystyvät sulkemaan virtapiirin nimellisvirtaa moninkertaisesti suuremmalla virralla ylikuormituksessa sekä oiko- tai maasulkutapauksessa. Siinä on myös tavallinen käsikäyttöinen nappi, josta piirin voi avata tai sulkea. Moottorisuojakatkaisinta saatetaan joskus kutsua etukojeeksi, mutta se on kansanomaisen nimitys. Kuviossa 11 on moottorisuojakatkaisimen kuva ja sen piirrosmerkki. (Mäkinen ym. 2009, 91–93.)



Kuvio 11. Moottorisuojakatkaisin ja sen piirrosmerkki

Kontaktori

Kontaktori on nimitys releelle, jolla ohjataan suuria sähkövirtoja, joita on esimerkiksi kolmivaihesähkömoottoreissa. Kontaktoreja käytetään päävirtapiireissä ohjaamaan pääjännitteitä, suuria virtoja ja suuria sähkötehoja. Kun kelalle tuodaan ohjausjännite, kaikki kolme vaihetta sulkeutuvat. Kuviossa 12 on kontaktorin kuva ja piirrosmerkki. (Mäkinen ym. 2009, 98.)



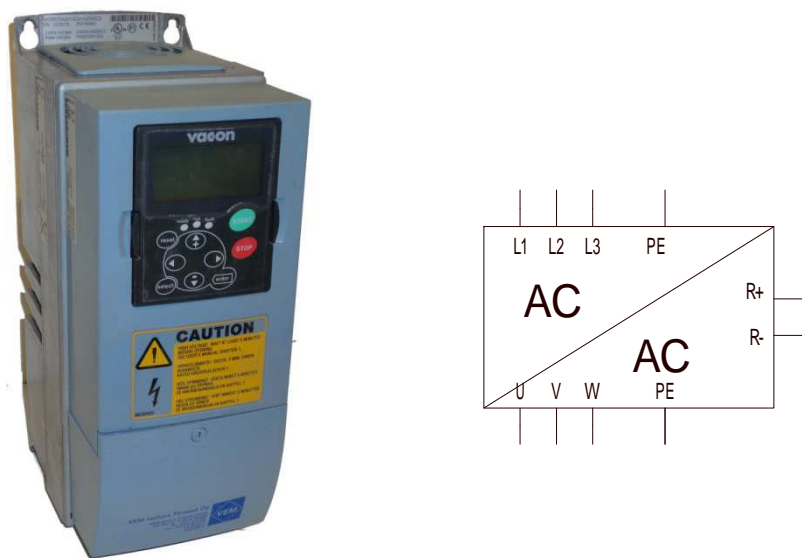
Kuvio 12. Kontaktori ja sen piirrosmerkki

Taajuusmuuttaja

Taajuusmuuttaja muuttaa nimensä mukaisesti syöttöjännitteen taajuutta. Niillä voidaan säätää sähkömoottorin nopeutta portaattomasti. Tyypillisesti taajuusmuuttajalla ohjattuja moottoreita on kuljettimissa, pumpuissa ja puhaltimissa. Taajuutta alentamalla saadaan hitaampi pyörimisnopeus ja vastaavasti taajuutta nostamalla saadaan nopeampi pyörimisnopeus. Taajuusmuuttajaa käyttämällä saadaan useita etuja. Ensinnäkin sillä voidaan alentaa sähkömoottorien käynnistyessä tapahtuvaa virtapiikkiä. Toiseksi taajuusmuuttajalla saadaan moottori käynnistymään tasaisesti kiihtyen. Niinpä esimerkiksi kuljettimien päällä olevat esineet eivät kaadu helposti, koska moottori ei aiheuta nykäisevää liikettä käynnistyessään. Hammasrattaisiin ja mekaanisiin voimasiirtolaitteistoihin ei tule myöskään vaurioita ja kulumista äkkinäisen käynnistymisen takia.

Prosessiteollisuudessa vältetään myös nesteiden aiheuttamilta paineiskuilta putkistoissa pumpun käynnistyttyä äkkinäisesti. Kolmanneksi saadaan säästettyä sähköenergiaa, koska taajuusmuuttajalla saadaan säädettyä moottorin nopeus aina kuormituksen vaatimaksi. Neljäs etu taajuusmuuttajassa on se, että se voidaan liittää osaksi automaatiojärjestelmää. Tällöin se voidaan ohjelmoida säätämään nopeutta automaattisesti tai sitä voidaan kauko-ohjata valvomosta. (Mäkinen ym. 2009, 139–145.)

Taajuusmuuttajan käytössä huonoja puolia on että moottorin melu, lämpeneminen, värinä ja häviöt voivat kasvaa. Tämä johtuu siitä, että taajuusmuuttajan syöttämä jännite ei ole sinimuotoista. Nämä ongelmat voidaan välttää käyttämällä moottoria, jonka käämeissä on vahvennettu eristys tai kytkemällä taajuusmuuttajan ja moottorin väliin dU/dt -kuristin tai siniaaltosuodin. Taajuusmuuttaja tarvitsee myös huoltoa aika ajoin. Sen puhallin ja ilmansuodatin tulee vaihtaa tarvittaessa. Kuviossa 13 on taajuusmuuttajan kuva ja sen piirrosmerkki. (Mäkinen ym. 2009, 139–145.)



Kuvio 13. Taajuusmuuttaja ja sen piirrosmerkki

Taajuusmuuttajan jarruvastus

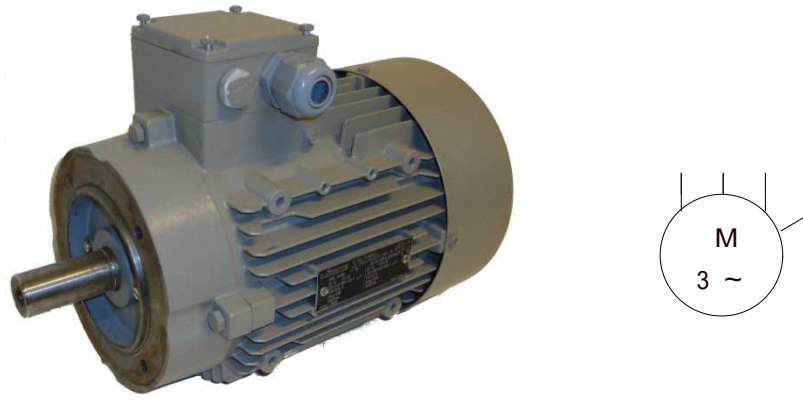
Sähkömoottorin vauhdin hidastamiseen tarvitaan jarruvastusta. Oikosulkumoottorin liike-energia synnyttää vaihtovirran taajuusmuuttajaan, joka muutetaan siellä tasavirraksi ja syötetään välipiiriin. Taajuusmuuttajan häviöt ja oma sähkökulutus alkavat tällöin jarruttaa moottoria. Kun siihen kytketään vielä erillinen jarruvastus, ylimääräinen teho voidaan syöttää siihen. Jarrutus syntyy siis siitä, että sähköenergia muutetaan lämpöenergiaksi jarruvastuksessa. Kuviossa 14 on jarruvastuksen kuva ja sen piirrosmerkki. (Keinänen ym. 2010, 160.)



Kuvio 14. Jarruvastus ja sen piirrosmerkki

Oikosulkumoottori

Oikosulkumoottori on yleisin teollisuudessa käytetty sähkömoottorityyppi. Sitä voidaan käyttää pumppujen, tuulettimien, kuljettimien ja sahojen liikuttamiseen sahateollisuudessa. Lisäksi oikosulkumoottoria voidaan käyttää monien muiden laitteiden käyttövoimana. Oikosulkumoottoreita voidaan kutsua myös epätahti- eli asynkronimoottoreiksi. Oikosulkumoottoreita valmistetaan sekä yksi- että kolmivaiheisina. Moottorin pyörivän keskiosan, eli roottorin, ympärillä on paikallaan pysyvä staattori. Staattorissa on käämejä, joihin johdettu vaihtovirta aiheuttaa pyörivän magneettikentän staattorin ja roottorin väliin. Roottorissa on oikosulkuun kytkettyjä käämejä, joihin staattorin magneettikentästä indusoituva jännite saa aikaan toisen magneettikentän. Roottorin käämien magneettikenttä lähtee seuramaan staattorista syntyvää magneettikenttää ja näin moottorin akseli alkaa pyöriä. Kuviossa 15 on oikosulkumoottorin kuva ja sen piirrosmerkki. (Keinänen ym. 2010, 147–149.)



Kuvio 15. Oikosulkumoottori ja sen piirrosmerkki

Ohjaussulake

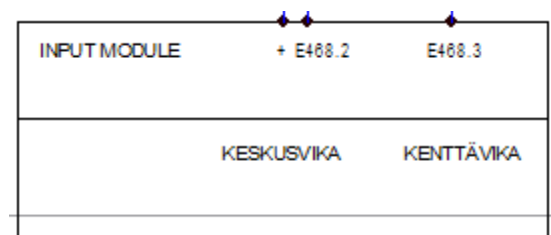
Ohjaussulake suojaa moottorilähdöissä ohjausjännitepiiriä liian suurelta virralta. Ohjaussulakkeena voidaan käyttää vanhemman mallista tulppasulaketta tai johdonsuojakatkaisijaa. Tulppasulakkeet ovat kertakäyttöisiä, ja ne joudutaan vaihtamaan aina, jos virta on päässyt kasvamaan liian suureksi. Kuviossa 16 on esitetty johdonsuojakatkaisija, joka pystytään sulakkeen laukeamisen jälkeen laittamaan takaisin päälle vivusta. Johdonsuojakatkaisijoissa on moottorisuojakatkaisimien tavoin kaksi mekanismia jotka pystyvät katkaisemaan virran. Jos kuormitus piirissä kasvaa liian suureksi, suurentunut virta saa termisen laukaisijan avaamaan ohjaussulakkeen. Oikosulkutilanteessa magneettinen pikalaukaisija avaa ohjaussulakkeen.



Kuvio 16. Ohjaussulake ja sen piirrosmerkki

Ohjelmoitavien logiikoiden korttimoduulit

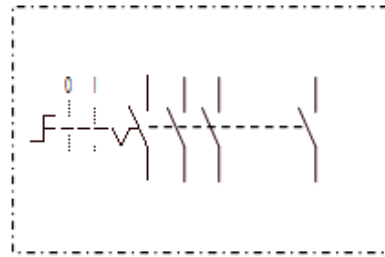
Ohjelmoitavat logiikat ohjaavat automaatiojärjestelmää. Ne ovat modulaarisia ja niihin pystytään lisäämään haluttu määrä digitaalisia tai analogisia sisään- ja ulostulokortteja. Tässä opinnäytetyössä esiintyvissä piirikaavioissa moottoreiden käyntitietoja on viety digitaaliseen sisääntulokorttiin. E3:n komponenttiluettelosta ei löytynyt valmiina sopivaa piirrosmerkkiä, joten piirsin itse korttimoduulia kuvaavan piirrosmerkin (ks. kuvio 17).



Kuvio 17. Digitaalinen sisääntulokortti ja sen piirrosmerkki

Turvakytkin

Turvakytkin on käsikäyttöinen vipu tai väännin, joka kytkee virran moottoripiiriin. Vipun asento ilmaisee selkeästi, onko virtapiiri kiinni vai auki. Turvakytkimiä valmistetaan kolme tai nelinapaisina rakenteina. Ne kestävät piirin kytkennän ja katkaisun siinä esiintyvillä virroilla. Turvakytkin tulisi olla valmistettu siten, että se täyttää auki-asennossa erottimen vaatimukset. Kytkimessä on paikka riippulukolle, jotta sen voi lukita kiinni asentoon huoltotöiden ajaksi. Tämä lisää työturvallisuutta, koska kukaan ei voi vahingossa kytkeä sähköpiiriin jännitettä työskentelyn aikana. Kuviossa 18 on turvakytkimen kuva ja sen piirrosmerkki. (Mäkinen ym. 2009, 106–107.)



Kuvio 18. Turvakytkin ja sen piirrosmerkki

4 KOMENTOSARJOJEN OHJELMOINTI

4.1 Tavoitteiden rajaus

Pronor Engineering on siis siirtynyt käyttämään uutta sähkösuunnitteluohjelmistoa nimeltään E3 Cable. Aikaisemmin yrityksellä oli ollut käytössä Vertex ED. Ongelmana edellisessä suunnitteluohjelmassa oli se, että sen suunnitteluautomaatiikka ei toiminut halutulla tavalla. Automaattisesti tuotetut dokumentit eivät olleet halutun näköisiä, eikä ohjelma ei pystynyt esittämään esimerkiksi ylemmän järjestelmän tunnusta ja lähdön numeroa peräkkäin. Asian korjaaminen on hankalaa, koska Vertex perustuu vanhalle ohjelmistoarkkitehtuurille, johon ei löydy enää osaamista. E3:ssa on parempi rajapinta omien komentosarjojen toteuttamiseen, jolla päästään haluttuun tulokseen. E3:n kaikki toiminnot pystytään tekemään VBS-komentosarjoina, ja niiden käyttämiseen löytyy ohjeet ohjelman omasta help-tiedostosta.

Pronor Engineeringin tehtävä on tuottaa sähkösuunnittelun dokumentit, joita tarvitaan teollisuussahalla. Dokumentit sisältävät piirikaavioita, kaapeliluetteloita, väyläkuvia ja muita asiakirjoja, joita tarvitaan automaatiojärjestelmää asentaessa ja myös myöhemmin korjaus-, huolto- tai muutostöitä tehtäessä. Asiakirjoja on paksu kansiollinen, mutta jokaista dokumenttia ei tarvitse tehdä yksitellen käsin. Sähkösuunnitteluohjelmistoilla on mahdollista luoda ne automaattisesti projektinmäärittelytiedoston pohjalta. Projektinmäärittelytiedosto on Excel-taulukko, jossa on listattu kaikki sahalla olevat moottorilähdöt ja tietoja niistä, esimerkiksi moottorilähdön nimi, ylemmän järjestelmän tunnus, PLC-korttimoduulien I/O-osoitteita ja muuta olennaista tietoa. Jos moottorilähtöjä on sahalla 50, projektinmäärittelytiedostossa on silloin 50 riviä. Näiden tietojen pohjalta saadaan tehtyä kansiollinen dokumentteja. Monet näistä asiakirjoista, esimerkiksi kaikki

taajuumuuttajakäynnisteisten moottoreiden piirikaaviot, ovat muuten samanlaisia, mutta tiedot, kuten lähdön nimi ja PLC-osoitteet, vaihtuvat. Kaikki dokumentit on kuitenkin tarpeen tehdä. Dokumenteilla on myös erilaisia esitystapoja. Vaikka niissä on samoja tietoja, ne pitää tuottaa, koska niitä tarvitaan erilaisia töitä varten.

E3 Cable -sähkösuunnitteluohjelman mukana oli tullut komentosarjatiedosto, joka pystyy siirtämään tietoa Excelistä siihen. Siinä ei kuitenkaan ollut kaikkia haluttuja toimintoja. Toimeksiantaja antoi minulle tehtäväksi selvittää, kuinka komentosarja toimii. Komentosarjan mukana tuli malliprojektinmäärittelytiedosto. Se ei ollut kuitenkaan esitystavaltaan sellainen, joka miellytti toimeksiantajaa, joten projektinmäärittelytiedosto piti muokata halutunlaiseksi. E3:ssa on myös toiminto, joka luo kaapeliluettelon projektista automaattisesti. Näin luotu kaapeliluettelo ei ollut kuitenkaan oikeanlainen, vaan pikemminkin kytkentätaulukko. Tähän piti keksiä ratkaisu, jonka tuloksena ohjelma loisi halutunlaisen kaapeliluettelon.

Sähkösuunnittelun komponenttien esittämiseen on olemassa monenlaisia eri piirrosmerkkejä. Toimeksiantoon kuului tehdä lista käytettävistä piirrosmerkeistä ja laatia kirjalliset ohjeet kuinka komentosarjoja käytetään. Opinnäytetyön tavoitteet olivat siis tiivistetysti seuraavat.

Projektin määrittelytiedosto muokataan halutunlaiseksi:

- Sarakkeiden otsikot käännetään suomenkielisiksi
- Valitaan sopiva määrä sarakkeita moottorilähdön komponenteille
- Valitaan sopivat nimet sarakkeille, esim. moottorinsuojakatkaisin, kontaktori, ohjaussulake, taajuusmuuttaja, taajuusmuuttajan jarruvastus, turvakytkin ja moottori
- Tehdään ohje suunnittelijoille kuinka määrittelytiedostoa luetaan.

Piirretään moottoripiirikaavioita:

- 1-suuntainen suorakäynnisteinen
- 2-suuntainen suorakäynnisteinen
- Taajuusmuuttaja käynnisteinen.

Lisäksi tuli toteuttaa seuraavat:

- E3n pitää luoda halutunlainen kaapeliluettelo
- Tehdään lista käytettävistä symboleista.

4.2 Moottoripiirikaavioihin ja suunnitteluohjelmistoon tutustuminen

Aloitin työn tutustumalla moottoripiirikaavioiden piirrosmerkkeihin. Käytin apuna teoksia Automaatiojärjestelmien logiikat ja ohjausjärjestelmät sekä Prosessiteollisuuden sähkö- ja automaatioasennukset. Niistä löytyi kaikki symbolit, joita Pronorin käyttämissä moottoripiirikaavioissa oli. Tutustuin myös siihen minkälaisia komponentit, kuten moottorinsuojakatkaisija, kontaktori ja taajuusmuuttajan jarruvastus ovat fyysisesti. Pronorilla oli näitä kyseisiä komponentteja hyllyssä, josta pääsin niitä tutkimaan. Kävin myös valmistajan kotisivuilla katsomassa niistä tietoja. Prosessiteollisuuden sähkö- ja automaatioasennukset -kirjasta löytyi tietoa, kuinka komponentit toimivat.

Jatkoin työtä tutustumalla E3-suunnitteluohjelmaan. Ohjelma ei ollut minulle entuudestaan tuttu, joten tuntui järkevältä ensin opetella käyttämään ohjelmaa. Kirjallisuuden perusteella oivalsin, että moottoripiirikaaviot on syytä piirtää, ennen kuin aloittaa ohjelmoinnin, koska komentosarjaa ei voi kirjoittaa, ellei ole olemassa valmista piirikaaviota, johon tietoa tuodaan.

4.3 Moottoripiirikaavioiden piirtäminen

Moottoripiirikaavioiden piirtämisen päätin aloittaa taajuusmuuttajakäynnisteisestä. Pyysin toimeksiantajalta valmiita esimerkkimalleja piirikaavioista ja tutustuin niihin. Esimerkkimallit olivat paperiversioina, joten piirsin ne uudestaan E3:lla, jotta piirikaaviot saataisiin sähköiseen muotoon. Esimerkkikaavioon oli jäänyt huolimattomuusvirhe, jossa ohjaussulakkeen merkitsemiseen oli käytetty kahta erilaista symbolia. Korvasin sen standardin SFS 501 mukaisella piirrosmerkillä. Muut piirrosmerkit ja esittämistavat piirikaaviossa olivat standardin mukaisia, joten en tehnyt niihin muutoksia. Taajuusmuuttajakäynnisteisen moottorin piirikaavio on esitetty liitteessä 1.

Jotta komentosarja pystyy luomaan projektiin piirikaavion, se täytyi muuttaa tiedostomuotoon (*.e3p). Kun pohjakuva luodaan, täytyy kaikki piirikaaviossa olevat komponentit valita hiirellä raahaamalla. Sen jälkeen valitaan file-valikosta *export* ja edelleen *drawing*. Aukeavasta ikkunasta varmistutaan, että *selection*-ruutu on valittuna, kirjoitetaan pohjakuvalle tiedoston nimi ja painetaan tallenna. Pohjakuva kannattaa nimetä hyvällä piirikaaviota kuvaavalla nimellä, koska komentosarja etsii tarkalleen samannimistä tiedostonimeä, jonka käyttäjä on kirjoittanut projektinmäärittelytiedoston pohjakuva-sarakkeeseen. Esimerkiksi isot ja pienet kirjaimet tulkitaan eri merkeiksi. Kaksi muuta piirikaaviota piirsin työn loppupuolella, kun olin jo pääosin tehnyt ohjelmointityön.

Laadin listan piirikaaviossa käytettävistä symboleista toimeksiantajan toivomuksesta. Tämän oli tarkoitus yhdenmukaistaa suunnittelutyötä. Perehdyttyäni sähkösuunnittelun dokumentoinnin standardeihin totesin, että yrityksen aikaisemmin tekemät dokumentit olivat asianmukaisesti tehtyjä. Mielestäni listasta ei tule olemaan paljonkaan hyötyä. Lisäksi sillä, mitä piirrosmerkkejä pohjakuvaan

laitetaan, ei ole mitään väliä, koska komentosarja muuttaa piirrosmerkit sellaisiksi kuin halutaan. Lista valituista piirrosmerkeistä on liitteessä 2.

4.4 Projektinmäärittelytiedoston muokkaaminen

Projektinmäärittelytiedoston pohjaksi toimeksiantaja valitsi vanhasta projektista yhden tiedoston, joka miellytti häntä eniten. Tällä pohjalla oli tehty myös monia aikaisempia projekteja. Tätä pohjaa käyttämällä uusien projektien tekeminen olisi helppoa, koska tiedot voi helpommin kopioida ja liittää vanhoista projekteista uusiin projekteihin. Lisäsin mallipohjaan sarakkeet pohjakuvalle, pohjakuvan sijainnille, turvapiirin numerolle, moottorin tyypille, viidelle PLC-osoitteelle, liitälaitteelle, moottorinsuojakatkaisimelle, kontaktorille, ohjaussulakkeelle ja turvakytkimelle. Laitoin sarakkeisiin vielä kommentit, kuinka laitteet ja PLC-osoitteet tulee nimetä piirikaavioihin, jotta komentosarja osaa hakea tiedot oikeasta paikasta.

Opinnäytetyön tekemisen aikana projektinmäärittelytiedostosta oli olemassa useampi versio. Ensimmäiset versiot oli tehty E3:n oman esimerkkitiedoston pohjalta, jonka käänsin suomen kielelle. Siinä oli kuitenkin paljon tilaa turhalle tiedolle, joten tein alusta alkaen oman version, jossa oli varattu tilaa vain sille tiedolle, jota itse käsittelin. Esittelin toimeksiantajalle eri versioita projektinmäärittelytiedostoista ja kehitin sitä saadun palautteen mukaan. Lopulta päädyttiin kuitenkin muunneltuun versioon tiedostosta, jota oli käytetty useammassa vanhassa projektissa. Esimerkkimalli lopullisesta projektinmäärittelytiedostosta on liitteessä 3. Tiedosto on Excelissä usean ruudun levyinen, joten se on jaettu usealle sivulle.

4.5 Ohjelmointiympäristöön perehtyminen

Koska Visual Basic ei ollut ohjelmointikielenä minulle tuttu, aloin tutustua siihen tekemällä neljä harjoitusta, jotka perehdyttivät ohjelman toimintoihin. Seuraavaksi tutustuin Excelin COM-komponenttimalleihin ja tein harjoitukseksi muutaman kokeilun, jossa hyödynsin niitä. Tässä vaiheessa tajusin, että VBS-komentosarjoja ei tehdä Visual Studiolla, vaan muistioon kirjoittamalla, ja ohjelmointi olisi hieman erilaista, kuin mitä olin harjoitellut harjoituksia tehdessä. Komentosarjojen tekemiseen tarvitaan komentosarjakieli VBScript. Hukkaan harjoitukset eivät menneet, sillä Visual Basicin ja VBScriptin syntaksissa on paljon yhtäläisyyksiä ja opin, kuinka esimerkiksi kommentit merkitään. Komentosarjojen kirjoittamiseen latsin Notepad++ ohjelman, joka tunnistaa VBS:n syntaksin ja värittää esimerkiksi kommentit ja tietotyypit eri väreillä, joten sitä on helpompi lukea. Ohjelma sisältää myös hakutoimintoja, joilla koodista voidaan etsiä muuttujia ja metodeja.

Tutustuin esimerkkinä olleeseen komentosarjaan ja selvitin rivi kerrallaan siinä esiintyvien funktioiden ja luokkien sisällön. Käytin apuna E3:n help-tiedostosta löytyvää luokkakirjastoa.

VBS syntaksia ja funktioita opiskelin internetistä löytyvistä ohjeista. Microsoftin omilta kotisivuilta löytyy Sesame Script -niminen blogi, joka opastaa komentosarjojen tekemiseen. Sivun lähtee liikkeelle aivan perusteista, ja ensimmäisen luvun harjoituksessa tehdään komentosarja, joka on yhden rivin pituinen. Myöhemmissä kappaleissa vaikeustasoa lisätään asteittain. Blogissa kerrotaan perusteellisesti, mitä komennot tekevät. Lisäksi ohjeet on kevyeen ja humoristiseen tyyliin kirjoitettu, joten niitä on mukava lukea. Sivun löytyy osoitteesta:

<http://technet.microsoft.com/en-us/scriptcenter/dd772284>. Toinen hyvä VBS-sivusto on osoitteessa <http://www.w3schools.com/>. Sivulla on kerrottu tiiviisti, kuinka VBS-muuttujat, taulukot, funktiot, aliohjelmat ja muut olennaiset asiat

toimivat. Kaikesta on olemassa hyvät esimerkit. Näiden lisäksi käytin myös Michael Halvorsonin Microsoft Visual Basic 2010 -kirjaa opiskelussa.

4.6 Komentosarjojen kirjoittaminen

Komentosarjan, joka luo moottoripiirikaaviot ja täyttää tiedot niihin, jaoin neljään osaan, jotta sen hahmottaminen ja muokkaaminen myöhemmin olisi helpompaa.

Ensimmäisessä osassa otetaan yhteys E3:en ja luodaan tarvittavat oliot, jotka sisältävät tarpeelliset metodit, joita tarvitaan projektin käsittelyssä. Ohjelma avaa ikkunan, jossa käyttäjää pyydetään valitsemaan haluttu projektinmäärittelytiedosto. Esimerkkinä käyttämäni komentosarjaan verrattuna tein sellaisen parannuksen, että nyt voidaan valita myös *.xlsx-muodossa tallennettu projektinmäärittelytiedosto eikä pelkästään *.xls.

Toisessa osassa luetaan tiedot projektinmäärittelytiedoston ensimmäisestä lähdestä. Tietojen perusteella luodaan ensimmäinen dokumentti projektiin. Loput asiakirjat luodaan myöhemmässä vaiheessa. Syy, miksi niitä ei luoda kaikkia samalla, on se, että tyhjässä projektissa on ensimmäinen sivu jo valmiiksi luotuna. Loput sivut luodaan toistosilmukalla, samalla kun tietoja luetaan projektinmäärittelytiedostosta, joten ensimmäisen sivun käsittely on erilainen. Jos käyttäjä on epähuomiossa valinnut tyhjän projektinmäärittelytiedoston, ohjelma näyttää ruudulla virheilmoituksen, joka ilmoittaa, että projektinmäärittelytiedosto on tyhjä, ja lopettaa komentosarjan suorittamisen.

Toisen osan alussa ohjelma tallentaa muuttujaan hakemiston polun, josta komentosarja on ajettu. Komentosarja ajetaan hakemistosta, jonka alla pitää olla */drawings_e* -niminen kansio, johon on tallennettu kaikki tarvittavat pohjakuvat. Ohjelma katsoo projektinmäärittelytiedostosta, mikä piirikaavio halutaan piirtää ja etsii samannimistä pohjakuvaa */drawings_e* -kansioista. Komentosarja voidaan

käynnistää mistä tahansa polusta, kunhan sen alla on kansio, johon pohjakuvat on tallennettu.

Suunnitteludokumentti saattaa sisältää useita sivuja. Esimerkiksi moottorilähdön pääkaavio ja ohjauspiirikaavio voi olla sijoitettu eri sivuille. Jos dokumenttiin liittyy useampi sivu, ne pitää kirjoittaa allekkain samaan soluun projektinmäärittelytiedostossa. Excelissä riviä solun sisällä vaihdetaan Alt+Enter -näppäinyhdistelmällä. Kun komentosarja lukee pohjakuvan projektinmäärittelytiedostosta, se luo taulukon, jossa on yhtä monta alkioita, kuin kyseisessä solussa on pohjakuvia merkittynä. Jokainen erillinen pohjakuva tallennetaan taulukon eri alkioon. Käytännössä useimmiten taulukossa on vain yksi alkio ja joskus kaksi. Jos alkioita on enemmän kuin yksi, komentosarja luo vastaavan määrän sivuja lisää projektiin ja tuo niihin kyseisen pohjakuvan. Uudet sivut nimetään järjestyksessä juoksevalla numeroinnilla.

Aina kun ohjelma tuo pohjakuvan sivulle, se kutsuu aliohjelmaa, jossa komponenttien tyypit vaihdetaan projektinmäärittelytiedostossa määriteltyihin komponentteihin. Tämä aliohjelman toteutus on komentosarjan neljännessä osassa. Aliohjelmaa käytetään ohjelmassa usein toistuvien asioiden tekemiseen. Aliohjelma etsii laitteita, jotka on nimetty tunnuksilla "LAITE01", "LAITE02" ja näin edespäin, ja vaihtaa näiden tilalle halutun laitteen. Myös PLC-logiikoiden osoitteet vaihtuvat samalla periaatteella. Ne vaihdetaan piirikaavioon tekstityökalulla tehdyille teksteille, joissa lukee "PLC001", "PLC002" ja näin edespäin. Laitteita ja PLC-osoitteita pystytään vaihtamaan viisi kappaletta molempia jokaiseen piirikaavioon. Tämä tekee komentosarjasta yleiskäyttöisemmän. Nyt käyttäjä voi piirtää itse piirikaavioita lisää, jos kolme piirtämääni piirikaaviota eivät riitä.

Tietojen vaihdon yhteydessä sivulle tuodaan myös ylemmän järjestelmän tunnus ja lähdön numero. Lisäksi ohjelma katsoo projektinmäärittelytiedostosta suunnittelijan

nimen, päivämäärän, projektin numeron ja muita tietoja, jotka se täyttää piirikaavion tietolaatikkoon.

Komentosarjan kolmannessa osassa luetaan tiedot projektinmäärittelytiedostosta toisesta lähdöstä eteenpäin. Kolmas osa on hyvin samanlainen kuin toinen osa, joka kuvailtiin edellisissä luvuissa, sillä erolla että toisessa osassa ei luotu ensimmäistä sivua, koska se on jo olemassa tyhjässä projektissa. Samalla kun tietoja luetaan, komentosarja luo uusia sivuja projektiin, laittaa niihin oikean pohjakuvan ja vaihtaa sen tiedot. Tämän toteutin silmukkarakenteena, jota ohjelma toistaa niin kauan, kuin projektinmäärittelytiedostossa on lähtöjä. Kolmannen osan lopussa kutsutaan vielä aliohjelmaa, joka poistaa tarpeettomat oliot, jotta ne eivät turhaan jää kuormittamaan keskusmuistia. Komentosarja, joka luo moottoripiirikaaviot projektinmäärittelytiedoston pohjalta, löytyy liitteestä 4.

E3:ssa on myös toiminto, jolla pystytään luomaan kaapeliluetteloita. Kaapeliluettelo, jonka toiminto luo, ei ollut kuitenkaan halutunlainen, vaan pikemminkin kytkentälista. Kaapeliluettelon luominen perustuu siihen, että se käynnistää komentosarjan eräästä kansioista, johon ohjelma on asennettu. Tein tämän komentosarjan tilalle oman versioni siitä. Se tulostaa suomenkielisen ja rakenteeltaan halutunlaisen kaapeliluettelon. Tätä kehitystä en vienyt kuitenkaan ihan loppuun asti, vaan kaapeliluetteloon tuodut tiedot jäivät puutteellisiksi. Sain kuitenkin selvitettyä, kuinka logiikka, jolla tietoa tuodaan myös ulospäin suunnitteluohjelmasta, toimii. Myös tästä komentosarjasta toimeksiantajalle jäi kommentoitu versio, jota voidaan kehittää eteenpäin. Komentosarja löytyy liitteestä 5.

Komentoin kattavasti komentosarjojani, jotta lähdettyäni muut pystyvät paremmin ymmärtämään ja kehittämään niitä jatkossa. Olen kommentoinut komentosarjat suomeksi toimeksiantajan toivomuksesta. Kävimme keskustelua aiheesta ja ehdotin

kommentointikieleksi englantia, koska englanti on yleismaailmallisempi kieli kuin suomi ja näin mahdollisesti useampi ihminen ymmärtää komentosarjaa, jos sattuu sitä lukemaan. Englanti tuntui luontevammalta myös siksi, että kaikki materiaali, jolla opettelin ohjelmoimaan, oli englanniksi. Opin näin ollen kaikki käsitteet myös englanniksi ja jouduin keksimään joillekin käsitteille nimiä suomeksi. Päädyin käyttämään toimeksiantajan toivomuksesta kommentointiin kuitenkin suomea. Lopuksi pidin esitelmän toimeksiantajalle ja toiselle yrityksessä työskentelevälle insinöörille komentosarjojen tekemisestä ja suunnittelun automatisoinnista.

5 KEHITYSKOhteet

5.1 Lisätoiminnot

Suunnittelun automatisointiprojekti ei missään nimessä tullut valmiiksi opinnäytetyön tekemisen aikana. Opinnäytetyön tekemisen loppupuolella yritys sai tilauksen sahalaituksen sähkösuunnittelun tekemisestä, missä oli yli 50 moottorilähtöä. Olisi ollut mielenkiintoista nähdä, kuinka hyvin tekemäni komentosarja olisi toiminut tositilanteessa. Samalla olisi huomattu, mitä lisätoimintoja komentosarjan toteutuksessa olisi tarvittu. Uskon, että opinnäytetyöni helpottaa jonkin verran dokumenttien tekemistä, mutta jättää vielä paljon asioita käsin tehtäväksi.

5.2 Käyttöliittymän kehittäminen

Komentosarja käynnistetään resurssien hallinnasta. E3:n tulee olla käynnistetty, ja siinä pitää olla avattuna uusi tyhjä projekti. Jos uutta projektia ei ole avattuna,

komentosarja ei toimi. Tämä on vaivalloista ja käyttäjän pitää olla hyvin perillä, missä järjestyksessä nämä asiat pitää tehdä. Käyttöliittymää voisi parantaa siten, että Exceliin avattuun projektinmäärittelytiedostoon tehtäisiin nappi, joka luo suunnitteludokumentit. Napissa voisi lukea selkeästi ”Luo suunnitteludokumentit E3:een”. Visual Basicilla on mahdollista tehdä sellaisia painikkeita. Napin painaminen käynnistäisi komentosarjan, joka tutkisi, mitä ohjelmia on jo käynnissä, ja käynnistäisi niitä tarpeen mukaan ilman, että käyttäjän tarvitsee huolehtia niistä. Tällä tavoin käyttöliittymää kehittämällä voitaisiin päästä tilanteeseen, jossa valmiin projektinmäärittelytiedoston muuttaminen kansiolliseksi suunnitteludokumentteja olisi vain kahden painalluksen päässä: luo suunnitteludokumentit ja tulosta.

5.3 Virheiden hallinta

Komentosarjojen virheiden hallintaa pitäisi kehittää. Ohjelman pitäisi esimerkiksi tarkistaa tietojen löytymistä, ja jos niitä ei löydy, ohjelman pitäisi antaa virheilmoitus sen sijaan että se kaatuisi hallitsemattomasti. Ohjelman pitäisi myös toimia, vaikka projektinmäärittelytiedostossa olisi pieniä kirjoitusvirheitä. Tällä hetkellä esimerkiksi isot ja pienet kirjaimet tulkitaan eri merkeiksi. Tämän asian korjaamiseen on olemassa funktio, mutta en ehtinyt toteuttaa sitä.

6 OPINNÄYTETYÖN SOVELTUVUUS KÄYTÄNTÖÖN

Olen tyytyväinen saavuttamiini tuloksiin, koska sain toteutettua kaikki tavoitteet, jotka työtä aloittaessa asetettiin. Sen lisäksi toteutin komentosarjan tavalla, joka mahdollisti minkä tahansa piirikaavion luomisen. Tämä lisää komentosarjan yleiskäyttöisyyttä huomattavasti. Aluksi komentosarjani pystyi luomaan vain ne

kolme piirikaaviota, jotka olin itse piirtänyt. Ohjelma etsi vain niitä komponentteja, jotka tiesin piirtäneeni piirikaavioihin ja vaihtoi ne haluttuihin. Työtä tehdessäni ymmärrykseni kasvoi koko ajan siitä, mihin suunnitteluautomaatiolla pyrittiin, ja siitä, mihin suunnitteluohjelman ohjelmointirajapinta pystyi. Kommentosarja pystyy tuomaan mitä tahansa piirikaavioita ja muuttamaan niiden komponentteja. Piirikaavioita tehdessä pitää muistaa vain nimetä siinä esiintyvät laitteet tietyllä tavalla.

Heti opinnäytetyön valmistumisen jälkeen toimeksiantaja vaikutti melko tyytyväiseltä työn tuloksiin. Erityisen tyytyväinen hän oli pitämäni esitelmään, joka auttoi paljon eteenpäin suunnitteluautomaation kehityksessä. Se lisäsi paljon ymmärrystä siitä, kuinka tietoa siirretään Excelistä suunnitteluohjelmaan ja mitä kaikkia toimintoja on mahdollista tehdä. Kehityskohteena toimeksiantaja olisi toivonut valmiimpaa tuotetta.

Kaksi kuukautta komentosarjojen valmistumisen jälkeen, kävin toimeksiantajan luona katsomassa kuinka työtä oli hyödynnetty. Toimeksiantaja oli kokeillut komentosarjojen toimivuutta pienen projektin sähkösuunnittelun toteuttamiseen, joka sisälsi yli kaksikymmentä moottorilähtöä. Kommentosarjat toimivat halutulla tavalla ja suuri osa suunnitteludokumenteista muodostuivat automaattisesti, mutta osa jouduttiin tekemään käsin. Isompien projektien suunnittelussa tullaan kevään aikana käyttämään Vertex ED:iä, koska sen suunnitteluautomaatiikka toimii vielä tällä hetkellä paremmin. Toimeksiantaja oli kuitenkin luottavainen E3:n mahdollisuuksiin, koska näkee siinä enemmän potentiaalia. Vertex ED:in lisenssi oli sanottu jo irti ja se on käytössä enään vuoden loppuun asti.

E3:n toimintojen automatisointi tarvitsee vielä tuotekehitystä ja karkeasti arvioiden tuotekehitysprosessi on 50 prosenttisesti valmis. Työn tekeminen vei tuotekehitystä eteenpäin ja säästi toimeksiantajan aikaa jota olisi kulunut komentosarjojen

toteuttamisen tutkimiseen ja selvittämiseen. Suurin hyöty työstä oli se, että toimeksiantajan ymmärrys kasvoi siitä mitä komentosarjoissa tapahtuu ja kuinka niitä tehdään. Työ loi pohjaa suunnitteluohjelmiston kehittämiseksi.

Ensimmäisen projektin aikana ilmeni useita jatkokehityskohteita, kuinka suunnittelun automatisointia tulee kehittää. Dokumenttien kehyksiin tulevat tiedot pitäisi saada pohjalehtien attribuutteihin, sen sijaan, että ne ovat vain pohjalehden päälle luotuja tekstikenttiä. PLC-osoitteet tulisi lisätä tulopiirikaavioon. Kaapelien tunnuksia pitäisi myös saada näkymään halutulla tavalla.

Opinnäytetyön aihe liittyi niin laajaan tuotekehityskohteeseen, että sitä ei pystytty viemään loppuun asti opinnäytetyön tekemiseen varatun ajan puitteissa.

Opinnäytetyön tekemiseen käytettiin 470 tuntia aikaa. Tarvitaan arviolta noin 200 tunnin työpanos lisää, jotta suunnittelun automatisointi on kehitetty pisteeseen, jossa se selkeästi nopeuttaa työskentelyä. Toimeksiantaja oli halukas palkkaamaan minut töihin, jotta kehitys saadaan vietyä loppuun asti.

Työ hyödyttää kokonaisuudessaan automaatioalaa, koska mikä tahansa yritys, joka käyttää tietokantapohjaista suunnitteluohjelmaa ja jolle suunnittelun automatisointi ei ole tuttu käsitteenä, voi oppia paljon tästä työstä. Komentosarjojen avulla voi räätälöidä E3:n luomaan omiin tarpeisiin sopivia suunnitteludokumentteja.

LÄHTEET

Burman, Kari. 2012. Pronor Engineering Oy toimitusjohtaja. Haastattelu 7.11.2012.

Component Object Model. MSDN Library. Artikkelin Microsoftin verkkosivuilla. Viitattu 6.1.2013. [http://msdn.microsoft.com/en-us/library/windows/desktop/ms680573\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/windows/desktop/ms680573(v=vs.85).aspx)

Heinola Sawmill Solutions yleisesittely. n.d. Esite. Viitattu 8.1.2013.

<http://www.heinolasm.fi/fi/esitteet/>

Keinänen, T., Kärkkäinen, P., Lähetkangas, M. & Sumujärvi, M. 2010.

Automaatiojärjestelmien logiikat ja ohjaustekniikat. 1.–2. p. Helsinki: WSOY.

Metsä ja Puu II –Tukista tuotteeksi. 2000. Toim. E. Paloheimo. Helsinki: Rakennustieto.

Metsäteollisuus ry:n rekisteröidyt jäsenyritykset. Viitattu 6.11.2012.

<http://www.metsateollisuus.fi/esittely/Sivut/jasenyryshaku.aspx>

Metsätilastollinen vuosikirja 2011. Sastamala: Metsäntutkimuslaitos. Viitattu 5.11.2012.

http://www.metla.fi/metinfo/tilasto/julkaisut/vsk/2011/vsk11_kokonaan_11.pdf

Mäkinen, M., Kallio, R. & Tantarimäki, R. 2009. Prosessiteollisuuden sähkö- ja automaatioasennukset. Helsinki: Otava.

Pere, A. & Kivimäki, J. 1998. Koneen- ja sähköpiirustuksen perusteet. Espoo: Kirpe.

SFS 501. 2004. Sähkökaavioissa käytettävät piirrosmerkit. Helsinki: Suomen Standardoimisliitto SFS, Viitattu 20.1.2013. <http://www.jamk.fi/kirjasto>, Nelli-portaali, SFS Online.

Sipi, M. 1998. Sahatavaratuotanto. Helsinki: Opetushallitus.

Suomen sahat ry jäsenyrytykset. Viitattu 6.11.2012. <http://www.suomensahat.fi/>

Suomen standardoimisliiton SFS ry:n kotisivut. Viitattu 8.1.2013. <http://www.sfs.fi/>

Suomen sähkö- ja elektroniikka-alan kansallinen standardoimisjärjestö SESKO ry:n verkkosivut. Viitattu 13.11.2012. <http://www.sesko.fi/portal/fi/>

The Sawmill Database. Viitattu 6.11.2012. <http://www.sawmilldatabase.com/>

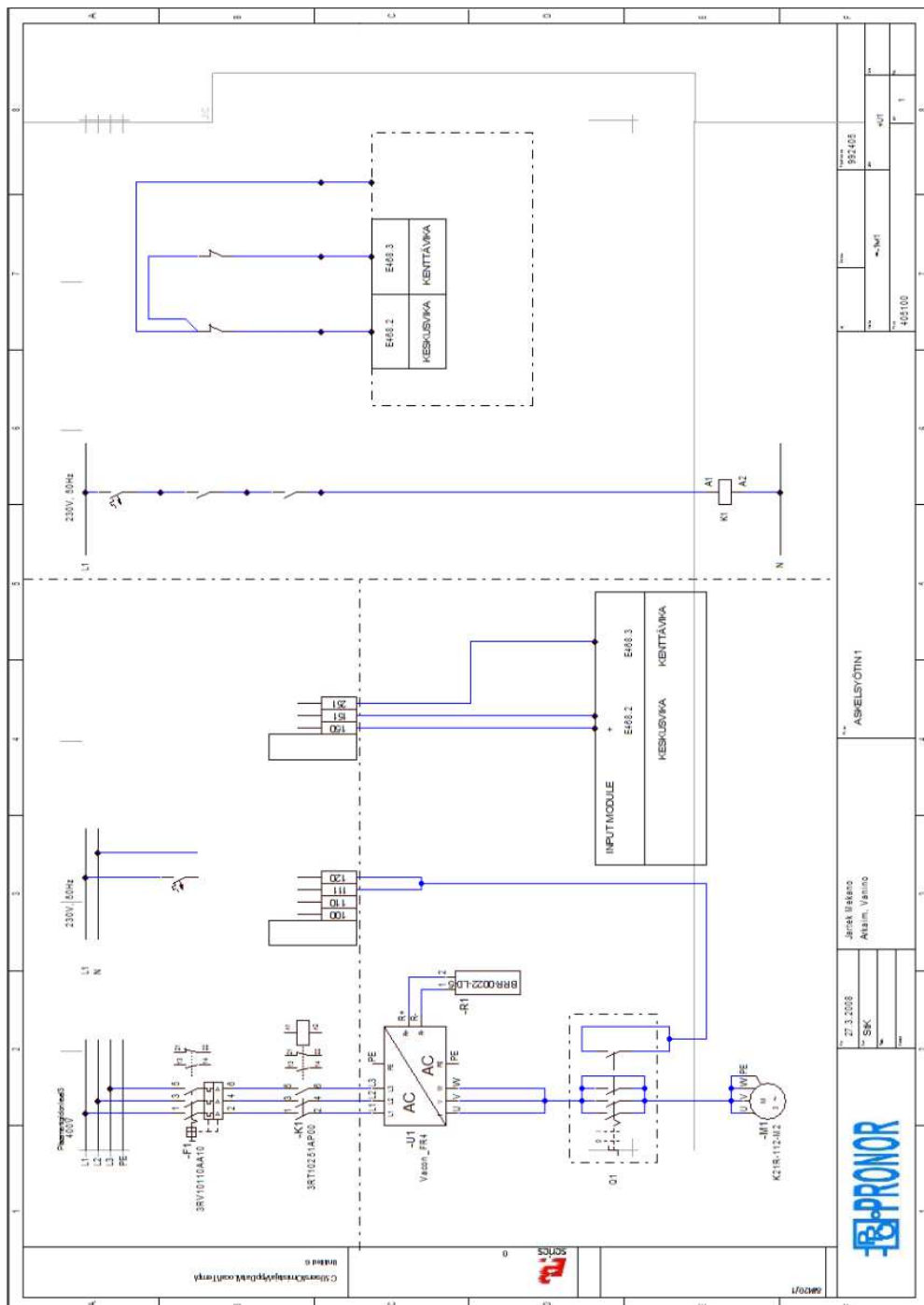
Torvelainen, J. 2012. Piensahat 2010. Vantaa: Metsäntutkimuslaitos, Metsätilastollinen tietopalvelu. Metsätilastotiedote. Viitattu 5.11.2012. http://www.metla.fi/metinfo/tilasto/julkaisut/mtt/2012/piensahat_2010.pdf

Valtanen, E. 2012. Tekniikan taulukkokirja. 19. p. Jyväskylä: Genesis-kirjat Oy.


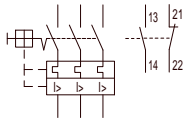
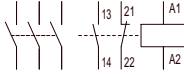
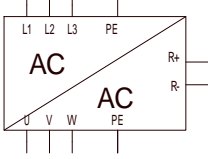


What is WSH? MSDN Library. Artikkelit Microsoftin verkkosivuilla. Viitattu 6.1.2013. [http://msdn.microsoft.com/en-us/library/shzd7dy4\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/shzd7dy4(VS.85).aspx)






LIITTEET

Liite 1. Taajuusmuuttajakäynnisteisen sähkömoottorin piirikaavio



Liite 2. Valitut sähkösuunnittelun symbolit

Komponentti	Symboli	
Kolmivaiheinen oikosulkumoottori	CIM06-08-03	
Moottorinsuojakatkaisin	CIM07-13-23	
Kontaktori	CIM07-02-06	
Taajuusmuuttaja	CIM06-14-01	
Jarruvastus	jARRUVASTUS_BRR_S	
Riviliitin	TermBlock1	

Kytkin, NO	CIM07-02-02	
Kytkin, NC	CIM07-02-04	
Rele	CIM07-15-01	
Ohjaussulake	IEC07-05-07	
Turvakytkin	IEC07-07-11	

Liite 3. Projektinmäärittelytiedosto



CONSUMER LIST

359MCCI
400VAC, 50Hz

Client: Esimerkki Asiakas
Project: Projekti01
Line: Packing

Voltage: 400 VAC		Elect. FC		Motor		Power		Frequency converter	
Line	Part	Qty	Desc	Pos	Power kW	In A	Total power kW	In A	Type
2	-1M1	U1	Механизм равномерной подачи ступенчатый	01	1	15,0	15,0	15,0	Vacon_FR4
3									
4			ASKELSYÖTIN 1				Total 15		
5									
6							OTP32		
7	-2M1	U2	Тр-р электродный 1	02	1	7,5	7,5	7,5	Vacon_FR4
8			KIRAMO 1				Total 7,5	7,5	
9									
10							OTP25		
11									
12	-3M1	U3	Тр-р электродный 2	02	1	7,5	7,5	7,5	VLT2875
13			KIRAMO 2				Total 7,5	7,5	
14									
15							OTP25		
16									
17	-4M1	U4	Транспортер с винтовой роликми 1	03	1	2,2	2,2	2,2	
18									
19			KIERERULLASTO 1				Total 2,2	2,2	
20									
21							OTP16		
22	-5M1	U5	Транспортер с винтовой роликми 2	03	1	3,0	3,00	3,0	
23									
24			KIERERULLASTO 2				Total 3	3	
25									
							OTP16		
			Motor power				Total	kw	
			Frequency converter power				Total	kw	

Liite 4. Piirikaavioiden luontikomentosarja

```
'Tämä komentosarja kysyy käyttäjältä projektinmäärittelytiedoston. Sen jälkeen
'komentosarja luo piirikaavioita sen pohjalta.
,
'komentosarja on jaettu neljään osaan. Jokainen osa on erotettu toisistaan viivalla.
,
'3.1.2013 Pronor Engineering Oy
'=====

'Tässä alkuosassa tehdään seuraavat toiminnot
'-Otetaan yhteys E3seen
'-Luodaan tarvittavat oliot
'-Pyydetään käyttäjää valitsemaan projektinmäärittelytiedosto

'Tämä koodirivi muodostaa yhteyden E3:n. Se toimii vain jos E3 on käynnistetty.
Set E3 = CreateObject("CT.Application")

'CreateJobObject on luokka joka luo projektin E3en. Tämä luokka on toiseksi ylimpänä
'E3n luokkahierarkiassa. Sitä voi verrata Excelin työkirjan luomiseen. Job olio on
'tärkeä ja sen avulla luodaan kaikki muut tarvittavat oliot.
Set Job = E3.CreateJobObject
'Tätä oliota käytetään laitteiden luomiseen.
Set Dev = Job.CreateDeviceObject
'Tällä oliolla toteutetaan sheettiin liittyviä toimintoja, kuten uusien uusien
'sheettien luominen ja pohjakuvien tuominen sheetille.
Set Sht = Job.CreateSheetObject
'Tällä metodilla luodaan graafisia elementtejä. Tässä komentosarjassa sitä on
'käytetty tekstin luomiseen.
Set Gra = Job.CreateGraphObject

'Luo Ikkuna olion josta voi valita halutun tiedoston.
Set Ikkuna = CreateObject("E3.Tools.Dialog")

'Avaa ruudulle näkyviin valintaikkunan josta voi valita halutun
'projektinmäärittelytiedoston. Alaviiva jakaa tekstin kahdelle riville. Teksti
'on laitettu kahdelle riville, koska ylipitkä tekstirivi ei näytä siistiltä.
FileName = Ikkuna.ShowOpen ("Projektinmäärittelytiedoston valinta", FileDir, _
FileName, "EXCEL (*.xlsx; *.xls)|*.xlsx; *.xls")
'Tämä siivoaa ylimääräisen olion pois.
Set Ikkuna = Nothing

'Käynnistää Excelin.
Set ExcelApp = CreateObject("Excel.Application")

'Avaa käyttäjän valitseman työkirjan.
Set ExcelWB = ExcelApp.WorkBooks.Open(FileName)

'Määritellään muuttuja Excel ja avataan "Sheets-Subcircuits" välilehti.
Set Excel = ExcelWB.WorkSheets("Sheet1")
```

```
'Luodaan muuttuja "rivi" ja tallennetaan siihen luku 14. Riviltä 14 ruvetaan
'lukemaan moottoripiirikaavioiden tietoja.
```

```
rivi = 14
```

```
'Tähän muuttujaan on tallennettu sheetin numero missä ollaan menossa. Tätä muuttujaa
'alihjelmien käsittelyyn, joissa sheettejä käsitellään.
```

```
sheet = 1
```

```
'-----
```

```
'Tässä osassa tuodaan tiedot projektinmäärittelytiedoston ensimmäisestä sarakkeesta.
'Tämä osa toteutetaan erillisenä osana, koska valmiissa projektissa on valmiina jo
'sheet 1 ja tämän takia käsittely on erilainen. Loput sheetit luodaan eri osassa ja
'niihin tulee silmukka joka luo samaan tahtiin sheettejä kun niitä on
'projektinmäärittelytiedostossa.
```

```
'Tämä etsii komentosarjan sijainnin ja tallentaa sen muuttujaan FileDir.
```

```
VBSName = wscript.ScriptFullName
```

```
pos = instrrev(VBSName, "\")
```

```
if pos = 0 then pos = len(VBSName) + 1
```

```
FileDir = left(VBSName, pos-1)
```

```
'Excel.Cells metodi lukee tietoa Excelin soluista. Sulkujen sisään syötetään solu
'minkä tiedot halutaan lukea. Tässä tiedot on tallennettu muuttujaan "pohjakuva".
```

```
pohjakuva = Excel.Cells(rivi, "T")
```

```
pohjakuvansijaintix = Excel.Cells(rivi, "U")
```

```
pohjakuvansijaintiy = Excel.Cells(rivi, "V")
```

```
'Jos lähdöllä on useampi pohjakuva ne tallennettu projektinmäärittelytiedostossa
'useammalle eri riville. Excelissä rivi vaihdetaan Alt+Enter painalluksella.
'Split-funktio jakaa pohjakuvan eri rivit yksiulotteisen taulukon eri sarakkeisiin.
'Tässä tapauksessa "pohjakuvaTaulukko" on uusi taulukko, johon tiedot tallennetaan.
'Suluissa ensimmäinen attribuutti on pätkittävä merkkijono ja toinen attribuutti
'on merkki josta ne pätkitään. VBS tulkitsee chr(10) Alt+Enter painallukseksi.
```

```
pohjakuvaTaulukko = Split(pohjakuva, chr(10))
```

```
pohjakuvansijaintixTaulukko = Split(pohjakuvansijaintix, chr(10))
```

```
pohjakuvansijaintiyTaulukko = Split(pohjakuvansijaintiy, chr(10))
```

```
'Jos projektinmäärittelytiedostossa ei ole mitään, komentosarjan ajaminen
```

```
'lopetetaan.
```

```
if pohjakuva = "" then
```

```
    'Näytetään ilmoitus. Parametri "1" saa sen luomaan ilmoitusikkunan windowssiin.
```

```
    'Jos siinä olisi "0", teksti näytettäisiin E3:n omassa ilmoituslaatikossa.
```

```
    E3.PutInfo 1, "Projektinmäärittelytiedosto on tyhjä. Tarkasta että se on" & _
    " oikein täytetty."
```

```
    'komentosarjan ajaminen lopetetaan.
```

```
    wscript.quit
```

```
end if
```

```
'Otetaan sheet aktiiviseksi.
```

```
Sht.Search 0, sheet
```

```

'PlacePart funktio laittaa part filen aktiiviselle sheetille. Ensimmäinen
'attribuutti on polku josta part fileä etsitään. "1" on part filen versio.
'Seuraavat kaksi attribuuttia ovat x ja y koordinaatit johon pohjakuva laitetaan.
'0 on "rot" parametri, joka ei ole vielä tämän E3 version käytössä. Jätä se
'arvoon 0.
Sht.PlacePart FileDir & "\\drawings_e\" & pohjakuvaTaulukko(0) & ".e3p", "1", _
pohjakuvansijaintixTaulukko(0), pohjakuvansijaintiyTaulukko(0), 0
vaihdTiedot

'Tutkitaan taulukon kokoa. Ubound funktio ilmoittaa taulukon suurimman indeksin.
'Tämä tieto tarvitaan, jotta tiedetään kuinka monta pohjakuvaa Exceliin on
'tallennettu ja kuinka monta sheettiä täytyy luoda.
taulukonSuurinIndeksi = Ubound(pohjakuvaTaulukko)

'Jos pohjakuvia on enemmän kuin yksi luodaan vastaava määrä sheettejä lisää.
If taulukonSuurinIndeksi > 0 Then
    'Tämä on for-toistosilmukka. Se luo yhtä monta sheettiä kun Exelin solussa
    'on pohjakuvia.
    For n = 1 To taulukonSuurinIndeksi
        'Tämä luo uuden sheetin.
        Sht.Create 0, n + 1, "pronor_a3_v", Sht.GetId, 0
        'Otetaan juuri luotu sheet aktiiviseksi...
        Sht.Search 0, n + 1
        '...ja tuodaan siihen pohjakuva.
        Sht.PlacePart FileDir & "\\drawings_e\" & pohjakuvaTaulukko(n) & ".e3p", _
        "1", pohjakuvansijaintixTaulukko(n), pohjakuvansijaintiyTaulukko(n), 0
        'Ajetaan aliohjelma "VaihdTiedot". Katso lopusta kuinka se toimii.
        VaihdTiedot
        'Lisätään muuttujaan sheet 1. Tästä pidetään kirjaa, jotta ohjelma tietää
        'mitkä sheetit on luotu ja missä ollaan menossa. Näin sheetit tulee nätisti
        'nimettyä numerojärjestykseen.
        sheet = sheet + 1
    'Tähän sanaan päättyy aina for-silmukka.
    Next
End If

'-----

'Tässä osassa luodaan loput sheetit ja tuodaan tiedot niihin, jos ne on määritelty
'projektinmäärittelytiedostossa.

'Tämä muuttuja luotiin vain tätä silmukkaa varten. Tältä riviltä aletaan laskemaan
'lähtöjen määrää projektinmäärittelytiedostosta.
b = 11

'Tämä silmukka laskee projektinmäärittelytiedoston sheettien määrän.
do
    ekasarake = Excel.Cells(b,"A")
    if ekasarake <> "" then
        sheettienlkm = sheettienlkm + 1
        b = b + 1
    end if
end do

```



```

    end if
'Silmukka käydään läpi niin monta kertaa kun solussa ei ole enää mitään tietoa.
'Merkintä "<>" tarkoittaa: Erisuuri kuin.
loop while ekasarake <> ""
'Sheettien lukumäärä jaetaan viidellä, koska projektinmäärittelytiedostossa
'yhdeällä lähdöllä on 5 riviä.
sheettienlkm = sheettienlkm / 5

'Jos sheettejä on enemmän kuin yksi niin toteutetaan silmukka.
If sheettienlkm > 1 Then
    'Tämä silmukka kutsuu aliohjelmaa, jossa luodaan loput sheetit ja tuodaan tiedot
    'niihin.
    For d = 2 To sheettienlkm
        rivi = rivi + 5
        'Kutsutaan aliohjelmaa "LaitaPohjakuva". Katso lopusta kuinka se toimii.
        LaitaPohjakuva
    Next

End If

'Kutsutaan aliohjelmaa "SiivoaOliot". Se tuhoaa ylimääräiset oliot pois muistia
'kuormittamasta.
SiivoaOliot

'PÄÄOHJELMA PÄÄTTYY TÄHÄN.

'-----

'Tässä osassa on aliohjelmien toiminnot. Aliohjelmia käytetään ohjelmassa usein
'toistuvien asioiden tekemiseen.

'Tämä aliohjelma laittaa pohjakuvan sheetille.
Sub LaitaPohjakuva

'Luetaan pohjakuva ja sen koordinaatit projektinmäärittelytiedostosta.
pohjakuva = Excel.Cells(rivi, "T")
pohjakuvansijaintix = Excel.Cells(rivi, "U")
pohjakuvansijaintiy = Excel.Cells(rivi, "V")

'Pätkitään pohjakuva ja sen koordinaatit taulukkoon.
pohjakuvaTaulukko = Split(pohjakuva, Chr(10))
pohjakuvansijaintixTaulukko = Split(pohjakuvansijaintix, Chr(10))
pohjakuvansijaintiyTaulukko = Split(pohjakuvansijaintiy, Chr(10))

'Tutkitaan taulukon kokoa.
taulukonSuurinIndeksi = Ubound(pohjakuvaTaulukko)

'For-toistosilmukka joka ajetaan yhtä monta kertaa kuin on pohjakuvia.
For w = 0 To taulukonSuurinIndeksi
    'Luodaan sheet.
    Sht.Create 0, sheet + 1, "pronor_a3_v", Sht.GetId, 0
    'Otetaan juuri luotu sheet aktiiviseksi.

```

```

Sht.Search 0, sheet + 1
'Asetaan pohjakuva siihen.
Sht.PlacePart FileDir & "\\drawings_e\" & pohjakuvaTaulukko(w) & ".e3p", "1", _
pohjakuvansijaintixTaulukko(w), pohjakuvansijaintiyTaulukko(w), 0
'Kutsutaan aliohjelmaa "Vaihd tiedot". Se vaihtaa komponenttien nimet ja
'PLC-osoitteet sheetillä.
VaihdTiedot
'Kasvatetaan muuttujaa sheet yhdellä.
sheet = sheet + 1

```

Next

End Sub

'Tämä aliohjelma etsii komponenttien nimiä, ylemmän järjestelmän tunnuksen,
'lähdön numeron sekä PLC osoitteet projektista ja muuttaa ne samanlaisiksi kun
'mitä projektinmäärittelytiedostoon on merkitty.

Sub VaihdTiedot

'Tässä tallennetaan tietoa projektinmäärittelytiedostosta.

```
assignment = Excel.Cells(rivi - 2, "B")
```

```
location = Excel.Cells(rivi - 2, "C")
```

```
piirustuksennimi = Excel.Cells(rivi, "D")
```

```
asiakas = Excel.Cells(5, "D")
```

```
projekti = Excel.Cells(6, "D")
```

```
projektinro = Excel.Cells(3, "R")
```

```
pvm = Excel.Cells(5, "R")
```

```
suunnittelija = Excel.Cells(4, "R")
```

```
piirustuksennro = Excel.Cells(6, "R")
```

```
moottorinsuojakatkaisija = Excel.Cells(rivi - 2, "AE")
```

```
kontaktori = Excel.Cells(rivi - 2, "AF")
```

```
taajuusmuuttaja = Excel.Cells(rivi - 2, "N")
```

```
jarruvastus = Excel.Cells(rivi - 2, "O")
```

```
moottori = Excel.Cells(rivi - 2, "X")
```

```
PLCosoite01 = Excel.Cells(rivi - 2, "Y")
```

```
PLCosoite02 = Excel.Cells(rivi - 2, "Z")
```

```
PLCosoite03 = Excel.Cells(rivi - 2, "AA")
```

```
PLCosoite04 = Excel.Cells(rivi - 2, "AB")
```

```
PLCosoite05 = Excel.Cells(rivi - 2, "AC")
```

'Vaihdetaan sheetin ylemmän järjestelmän tunnus ja lähdön numero.

```
Sht.SetAssignment assignment
```

```
Sht.SetLocation location
```

'Tallennetaan käsiteltävän sheetin ID tunnus muuttujaan shtid. ID tunnusta tarvitaan
'tekstin kirjoittamisessa sheetille.

```
shtid = Sht.GetId
```

'Luodaan tekstilaatikoita sheetille.

```
Gra.CreateText shtid, piirustuksennimi, 189, 22
```

```

Gra.CreateText shtid, piirustuksenro, 318, 7
Gra.CreateText shtid, asiakas, 120, 25
Gra.CreateText shtid, projekti, 120, 20
Gra.CreateText shtid, suunnittelija, 89, 19
Gra.CreateText shtid, pvm, 89, 25
Gra.CreateText shtid, projektinro, 370, 24

```

```

'Funktio laskee laitteiden (Device) lukumäärän projektissa ja tallettaa ne muuttujaan
'"Devicenlkm". "DevIds" on yksiulotteinen taulukko johon on tallennettu kaikkien
'laitteiden ID tunnuksset. Huom. toisin kuin yleensä ohjelmoinnissa, taulukon
'ensimmäinen luku tallennetaan taulukon kohtaan (1) eikä (0).
Devicenlkm = Job.GetDeviceIds(DevIds)

```

```

'Tämä silmukka etsii komponentin nimen ja korvaa sen projektinmäärittelytiedostoon
'merkityllä komponentilla.

```

```

for i = 1 to Devicenlkm
  Dev.SetId DevIds(i)
  'Tallennetaan laitteen nimitys eli "Device Designation", joka löytyy laitteen
  'properties valikosta muuttujaan komponentindesignation muuttujaan.
  komponentindesignation = Dev.GetName

  'Jos laitteen nimitys on "-LAITE01" niin...
  if komponentindesignation = "-LAITE01" Then
    '...vaihdetaan sen nimeksi mikä laite löytyy projektinmäärittelytiedostosta...
    Dev.SetComponentName moottorinsuojakatkaisija, 1
    '...ja vaihdetaan laitteen nimitys "-F1".
    Dev.SetName "-F1"
  end if

  if komponentindesignation = "-LAITE02" Then
    Dev.SetComponentName kontaktori, 1
    Dev.SetName "-K1"
  end if

  if komponentindesignation = "-LAITE03" Then
    Dev.SetComponentName taajuusmuuttaja, 1
    Dev.SetName "-U1"
  end if

  if komponentindesignation = "-LAITE04" Then
    Dev.SetComponentName jarruvastus, 1
    Dev.SetName "-R1"
  end if

  if komponentindesignation = "-LAITE05" Then
    Dev.SetComponentName moottori, 1
    Dev.SetName "-M1"
  end if
next

'Vaihdetaan PLC osoitteet-

```

```

'Tämä funktio laskee sheetillä olevien tekstielementtien määrän ja tallentaa niiden
'ID-tunnukset. Muuttujaan graphcount tulee elementtien määrä ja graphIds on taulukko
'johon on tallennettu kaikki ID-tunnukset.
graphcount = Sht.GetGraphIds(graphIds)
'Toistosilmukka käydään läpi niin monta kertaa kun sheetillä on graafisia elementtejä.
'Siinä etsitään merkkijonoja esim. PLC001, PLC002, jne ja vaihdetaan niiden paikalle
'projektinmäärittelytiedostosta löytyvä osoite.
for m = 1 to graphcount
    Gra.SetId graphIds(m)

    PLCOsoite = Gra.GetText

    If PLCOsoite = "PLC001" Then
        Gra.SetText PLCOsoite01
    End If

    If PLCOsoite = "PLC002" Then
        Gra.SetText PLCOsoite02
    End If

    If PLCOsoite = "PLC003" Then
        Gra.SetText PLCOsoite03
    End If

    If PLCOsoite = "PLC004" Then
        Gra.SetText PLCOsoite04
    End If

    If PLCOsoite = "PLC005" Then
        Gra.SetText PLCOsoite05
    End If

next

End Sub

'Siivotaan pois ylimääräisiä olioita.
Sub SiivotaOliot
Set ExcelApp = Nothing
Set ExcelWB = Nothing
Set Excel = Nothing
Set Job = Nothing
Set Dev = Nothing
Set Sht = Nothing
Set Gra = Nothing
Set E3 = Nothing
End Sub

```

Liite 5. Kaapeliluettelon luontikomentosarja

```
'Tämä skripti luo kaapeliluettelon, jonka E3 luo Tools -> Reports -> Excel -> Cables
'valikosta. Skriptin tiedoston nimeksi pitää vaihtaa CableExcel.vbs ja se pitää
'laittaa \Zuken\E3.series_2011\reports kansioon. Muista ottaa varmuuskopio
'CableExcel.vbs tiedostosta, koska tämä skripti ei sisällä kaikkia toiminnallisuuksia
'mitä alkuperäisessä on.
'
'30.11.2012 Pronor Engineering Oy
'=====

'Tämä koodirivi muodostaa yhteyden E3:n. Se toimii vain jos E3 on käynnistetty.
Set E3 = CreateObject("CT.Application")
'Tätä metodia käytetään kaapeleiden, liittimien ja muiden laitteiden luomiseen.
Set Job = E3.CreateJobObject
'Tällä metodilla luodaan sheet.
Set Sht = Job.CreateSheetObject

'Käynnistää Excelin.
Set objExcelApp = CreateObject("Excel.Application")

'Tämä rivi saa sen näkyväksi.
objExcelApp.Visible = True

'Tämä avaa kaapeliluettelon mallipohjan.
Set objExcelWB = objExcelApp.Workbooks.Open("Z:\Moottoripiirikaavioiden " & _
"suunnittelun automatisointi -opinnäytetyö\Kaapeliluettelon " & _
"mallipohja\Kaapeliluettelon mallipohja.xlsx")

'Tämä avaa avatun workbookin ensimmäisen välilehden.
Set objExcelWS = objExcelWB.Worksheets(1)

'Tämä funktio laskee sheettien määrän projektissä. Muuttujaan sheettienlkm on
'talletettu sheettien lukumäärä projektissa. SheetIdArr on yksiulotteinen taulukko
'johon on tallennettu kaikkien sheettien Id tunnus.
sheettienlkm = Job.GetSheetIds(SheetIdArr)

'Määritellään yksiulotteinen taulukko jonka koko on 3. Tähän taulukkoon
'tallennetaan kojeiden tunnuksia M1, Q1 ja K1.
Dim kojetaulukko(2)

'Tässä silmukassa täytetään kaapeliluettelo tiedoilla. Siinä on kaksi sisäkkäistä
'silmukkaa. Ulompi silmukka toistetaan yhtä monta kertaa kun projektissa on
'sheettejä. Sisempi silmukka toistetaan kolme kertaa, koska taajuusmuuttajasta
'lähtee kolme johtoa.
for n = 1 to sheettienlkm - 9

    'Avataan sheet 1, 2, 3, jne.
    Sht.Search 0, n
```

```

'Tallennetaan kaapeli tyyppejä muuttujiin jotta niitä on helpompi kirjoittaa
'myöhemmin.
kaapeli01 = "MCMK 3x6+6"
kaapeli02 = "MMJ 3x1.5S"

'Tallennetaan kojetaulukon eri sarakkeisiin komponenttien tunnuksia.
kojetaulukko(0) = "M1"
kojetaulukko(1) = "Q1"
kojetaulukko(2) = "K1"
'Haetaan ylemmän järjestelmän tunnus muuttujaan assignment.
assignment = Sht.GetAssignment
'Haetaan lähdön numero muuttujaan location.
location = Sht.GetLocation

'Tässä on sisempi silmukka. Se täyttää kaapeliluetteloön tarvittavat tiedot. Se
'käydään läpi kolme kertaa koska taajuusmuuttajasta lähtee kolme kaapelia.
for i = 1 to 3
  'Viedään ylemmän järjestelmän tunnus ja lähdön numero sarakkeeseen 9A.
  objExcelWS.Cells(n * 3 + i + 5, 1) = assignment & location & "-" &
  kojetaulukko(i - 1)
  'Viedään ylemmän järjestelmän tunnus sarakkeeseen 9B ja 9E.
  objExcelWS.Cells(n * 3 + i + 5, 2) = assignment
  objExcelWS.Cells(n * 3 + i + 5, 5) = assignment
  'Viedään lähdön numero sarakkeeseen 9C ja 9F.
  objExcelWS.Cells(n * 3 + i + 5, 3) = location
  objExcelWS.Cells(n * 3 + i + 5, 6) = location
  'Viedään taajuusmuuttajan tunnus "U1" sarakkeeseen 9D.
  objExcelWS.Cells(n * 3 + i + 5, 4) = "U1"
  'Viedään oikosulkumoottorin tunnus "M1" sarakkeeseen 9G.
  objExcelWS.Cells(n * 3 + i + 5, 7) = kojetaulukko(i - 1)
  'Viedään kaapelityyppejä sarakkeeseen 9H. Eli jos kaapeli on menossa
  'laitteelle M1...
  if kojetaulukko(i-1) = "M1" then
    '...laitetaan sarakkeeseen "kaapeli01" ja...
    objExcelWS.Cells(n * 3 + i + 5, 8) = kaapeli01
  else
    '...muuten, "kaapeli02".
    objExcelWS.Cells(n * 3 + i + 5, 8) = kaapeli02
  end if
next
next

```