



# CADS-generointityökalun jatkokehitys

Matti Koskinen

OPINNÄYTETYÖ  
Maaliskuu 2019

Sähkö- ja automaatiotekniikan koulutusohjelma  
Automaatiotekniikka

## TIIVISTELMÄ

Tampereen ammattikorkeakoulu  
Sähkötekniikan koulutusohjelma  
Automaatiotekniikka

KOSKINEN, MATTI:  
CADS-generointityökalun jatkokehitys

Opinnäytetyö 54 sivua, joista liitteitä 5 sivua  
Maaliskuu 2019

---

Opinnäytetyö käsittelee CADs Electric -suunnitteluohjelman ja Excel-taulukko-laskentaohjelman käyttämistä yhdessä tehokkaaseen suunnitteluun. Opinnäytetyön teettäjänä oli Insta Automation Oy, joka toimii laajalti automaatiotekniikan ratkaisujen parissa. Kohdattuja ongelmia kartoitettiin tekemällä haastatteluita suunnittelijoille, ja näihin ongelmiin pyrittiin löytämään ratkaisuja. Työn tavoitteena on kehittää monipuolinen generointikirjasto ja luoda uusia kehitysehdotuksia generointityökalun parantamiseksi.

Generointikirjasto koostuu useista piirikaavioista, jotka ovat jalostettavissa monipuolisesti erilaisiin tilanteisiin asiakkaan tarpeiden mukaisesti. Piirikaavioiden attribuutteihin sijoitettavat generointilinkit luovat yhteyden piirikaavion ja tiedonhakuun käytetyn Excel-tiedoston välille. Generointityökalu on toteutettu Visual Basic for Applications -ohjelmointikielillä, joka hakee useasta lähtötietoja sisältävästä Excel-tiedostosta tiedot yhteen Excel-tiedostoon. Kun lähtötiedot on koottu yhteen tiedostoon, voidaan pohjakuva generoida Excel-tiedoston kanssa. Lopputuloksena syntyy valmiita piirikaavioita sisältäen laitteiden, keskusten ja kaapeleiden tiedot. Generointi tehostaa suunnittelua ja vähentää inhimillisten virheiden määrää.

Generointityökalun täydellinen automatisointi on lähtötietojen erilaisuuden takia lähes mahdotonta. Tietojen täyttäminen tiedonhakuja varten on melko yksinkertainen asia, ja siihen laadittiin ohjeistus tämän kehitystyön aikana. Jatkokehityksenä on mainittu muutamia menetelmiä ohjeistukseen ja käytettävän VBA-koodin parantamiseksi. Generointikirjasto saatiin laajennettua kattavaksi ja käyttökelpoiseksi tulevaisissa projekteissa. Kirjaston laajeneminen uusilla piirikaavioilla parantaa sen käytettävyyttä monipuolisiin kokonaisuuksiin. Nykyisten generointipohjien avulla kirjastoa voidaan pitää ajan tasalla uusien laitteiden yleistyessä.

Opinnäytetyö on julistettu kehitystyön ja liitteiden osalta salaiseksi yrityksen tietojen suojelemiseksi.

## **ABSTRACT**

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

KOSKINEN MATTI:  
Further development of a CADS Generating Tool

Bachelor's thesis 54 pages, appendices 5  
March 2019

---

The purpose of this thesis was to find an effective way an effective way to produce circuit diagrams by using CADS Electric and Excel together. The development was commissioned by Insta Automation Oy, which works extensively in the field of automation technology solutions. The problems encountered were surveyed by interviews with designers and solutions were sought to these problems. The aim of this thesis was to develop a versatile generation library and create new development suggestions to improve the generating tool.

The Generation Library consists of several circuit diagrams that can be refined in a multitude of different situations according to the customer's needs. Generating links to the attributes of circuit diagrams establish a connection between the circuit diagram and the Excel file. The Generation Tool has been made by using the Visual Basic for Applications programming language, which retrieves data from multiple Excel files. From different files, the data is compiled into one file that is generated with the circuit diagram. The end result is a complete circuit diagram containing information about the devices, cables, and electrical distribution center. Automated generation reduces the risk of human error and improves the planning process.

Complete automation of the generating tool is almost impossible due to the differences between various source data. However, filling in the information for information retrieval is fairly simple and guidance material was made at the time of this development. Some methods for visualizing information retrieval and improving the used VBA code were considered as possible areas for further development. The Generation Library was expanded to include comprehensive circuit diagrams and it is useful for future projects. Expanding the library with new circuit diagrams improves its usability. With existing generating bases, the library can be kept up-to-date as new devices become more common.

Confidential information was excluded from the public version of this thesis.

## SISÄLLYS

1	JOHDANTO .....	6
1.1	INSTA AUTOMATION OY .....	7
2	GENEROINTITYÖKALUN OHJELMAT .....	8
2.1	VBA-ohjelmointikieli yleisesti.....	8
2.1.1	Makrojen rakenne.....	9
2.1.2	Ohjauksrakenteet .....	11
2.1.3	Ohjelman hallinnointi .....	15
2.1.4	Nimeämiskäytännöt.....	15
2.2	CADS-tuoteperhe.....	16
2.2.1	Piirto-ominaisuudet.....	18
2.2.2	Tietokanta.....	19
2.2.3	Generointilinkit attribuutteihin .....	20
3	ONGELMIEN KARTOITUS.....	23
3.1	Ongelmien kerääminen .....	23
3.2	Kerätyn datan analysointi.....	24
4	KEHITYSTYÖ .....	27
4.1	Generointikirjasto .....	27
4.1.1	Koneet.....	27
4.1.2	Instrumentit.....	27
4.1.3	Venttiilit.....	27
4.2	Generointityökalun kehitys .....	27
4.2.1	Toiminnallisuus.....	27
4.2.2	Ulkoasu .....	27
5	POHDINTA .....	28
	LÄHTEET.....	31
	LIITTEET .....	32
	Liite 1. Tietokantanumerot (Insta Automation verkkolevy) .....	32
	Liite 2. Tiedonhakuohje.....	32

**ERITYSSANASTO**

CADS	Kymdata Oy:n kehittämä CADS on sähkötekniikassa yleisesti käytetty CAD-ohjelma sähkö- ja piirikaavioiden suunnitteluun
DB	Tässä yhteydessä tietokanta, jota voidaan CADS-ohjelmassa hyödyntää esimerkiksi kaapeli- ja komponenttietojen hakemiseen
Generointi	Generoinnilla tässä yhteydessä tarkoitetaan Excel-solujen sisällä olevien tietojen kopiointia
Makro	Tietokoneohjelman laji, jolla saadaan sovelluksen toimintasarjat suoritettua käyttäjän puolesta
Pivot	Ominaisuus, jolla voidaan luoda analyyseja Excelissä
Rasteri	Piirto-ominaisuus suunnitteluohjelmassa, jolla voidaan liikkua koordinaatistossa tietyn askelluksen mukaisesti
Snap	Piirto-ominaisuus suunnitteluohjelmassa, jolla voidaan tarttua haluttuun elementtiin.
Seula	Suunnitteluohjelman suodatustoiminto, jolla voidaan suodattaa vain halutut elementit
VBA	Visual Basic for Applications on ohjelmointikieli, jota käytetään Microsoft Office -tuoteperheessä
UPS	(Uninterruptible Power Supply) akkuvarmennettu tehonsyöttö.

## 1 JOHDANTO

Opinnäytetyön tarkoituksena on kehittää Insta Automation Oy:n suunnitteluosaston piirikaavioiden generointityökalua ja tuoda uusia kehitysehdotuksia generointityökalun parantamiseksi. Työssä kartoitetaan suunnittelijoiden kokemia ongelmia haastatteluiden muodossa ja pyritään löytämään ratkaisuja niihin. Tavoitteena on yhtenäistää yrityksen työntekijöiden suunnittelua sekä tehostaa sitä generoinnin avulla siinä määrin kuin se on projekteissa mahdollista. Nykyinen generointityökalu ei toimi Windows 10 -käyttöjärjestelmällä ja generointi pitää tehdä virtuaalikoneen avulla. Lähes kaikki suunnittelijat kuitenkin käyttävät CADS-suunnitteluohjelmaa ilman virtuaalikonetta, joten liikkuminen tietokoneiden välillä on hieman työlästä. Tähän ongelmaan etsitään ratkaisua kehitystyön aikana.

Opinnäytetyön tavoitteena on saada generointityökalu toimimaan Windows 10-käyttöjärjestelmällä sekä luoda kattava kirjasto erilaisia generointipohjia suunnittelijoiden työn tehostamiseksi. Generointipohjien avulla saadaan yhtenäistettyä suunnittelutyötä, koska piirikaavioissa moottorit, venttiilit, instrumentit ja niihin liittyvät kojeet sijaitsevat siististi piirikaaviossa. Generointipohjiin syötetään kojeiden attribuutteihin generointityökaluun liittyvät linkit, joiden avulla tiedot haetaan generointityökalun avulla.

Kuvien yhtenäistämiseksi käytettiin vuoden 2018 piirikaavioita tietolähteenä, joista nähtiin tyypillisiä moottori-, venttiili- ja instrumenttipiirikaavioiden esitystapoja. Kuvia oli useita tuhansia, joten niiden perusteella saatiin kattava käsitys erilaisista esitystavoista ja variaatioista esimerkiksi moottoreiden ohjaukseen liittyen. Lisäksi generointipohjien luomisessa käytettiin pohjakuvaan sidottua sapluunaa, jonka avulla esimerkiksi moottori ja sulakkeet saadaan sidottua täsmälleen samaan kohtaan jokaisessa kuvassa. Sapluunan avulla saadaan myös keskuksen ja kentän välinen rajausta yhtenäiseksi eri kuvissa.

## 1.1 INSTA AUTOMATION OY

Insta Automation on yritys, joka kuuluu Insta Group Oy konserniin. Insta-konserni on sähköautomaation, digitaalisen tietoturvan ja puolustusteknologian huippuosaaja, joka rakentaa ja ylläpitää turvallista ja kilpailukykyistä yhteiskuntaa. Insta Group Oy on vuonna 1960 perustettu perheyritys. Edellä mainittuun organisaatioon kuuluvat tytäryhtiöt:

- Insta Automation Oy
- Insta ILS Oy
- Insta DefSec Oy
- Intopalo Digital Oy

(Insta Group Oy 2019)

Insta Automation Oy on teollisuusautomaatioon keskittynyt yritys, jonka liiketoimintaan kuuluvat suunnittelu-, asennus-, keskusvalmistus- ja elinkaaripalvelut. Insta Automation Oy tuottaa palveluita kokonaistoimituksina sekä erillistoimituksina asiakkaan tarpeiden mukaisesti. Insta Automation Oy on kasvattanut liiketoimintaansa lukuisien yritysostojen avulla. Insta Automation Oy yrityksen alaisuuteen kuuluu seuraavat tytäryhtiöt:

- Goodwork Oy
- KMJ-Engineering Oy
- Mattila Porvoo Oy
- Asitek Oy
- Tampereen Automaatiosähkö Oy

Konsernissa työskenteli yhteensä noin 1000 henkilöä vuonna 2017, joista 500 työskenteli Insta Automation Oy:ssä. Insta Automation Oy:n liikevaihto vuonna 2017 oli noin 67 miljoonaa euroa, joka on 62 % koko organisaation liikevaihdosta. Insta Automation Oy:n tärkeimpiä asiakastoimialoja ovat prosessiteollisuus, energian tuotanto, materiaalin käsittely ja vesilaitokset. (Insta Group Oy 2019)

## 2 GENEROINTITYÖKALUN OHJELMAT

Generointityökalun toiminta perustuu Visual Basic for Applications VBA-ohjelmointikielen ja siitä syystä ohjelmointikielen toiminta esitellään tässä siltä osin kuin se on tarpeellista generointityökalun toiminnan ymmärtämisen ja kehitysehdotusten luomisen kannalta. Tässä kappaleessa tutustutaan päällisin puolin VBA-koodin ominaisuuksiin ja käyttökohteisiin erilaisessa tiedon hallinnoinnissa Excel –taulukkolaskentaohjelmassa. Ohjelmointikielillä voidaan rakentaa lisäpotkua kaikkiin Microsoft Officen ohjelmilla tehtyihin sovelluksiin. VBA-kieltä voidaan siis Excelin lisäksi hyödyntää myös PowerPoint- ja Word-ohjelmien tukena, mutta käyttökohteet ovat edellä mainituissa melko vähäiset.

CADS Electric on CADS-suunnitteluohjelman lisäosa, jossa piirikaaviot varsinaisesti luodaan ja Excelin avulla ainoastaan siirretään tarvittavat tiedot haluttuun kohtaan piirikaaviossa. CADS Electric-suunnitteluohjelmalla luodaan raamit generointipohjille, joihin tuodaan tarvittavat tiedot Excelillä suoritettavan tiedonhakutyökalun avulla. Valmiit piirikaaviot syntyvät Excelillä suoritettavan generointityökalun ja CADS –suunnitteluohjelman avulla piirrettyjen pohjakuvien saumattoman yhteistyön tuotoksena.

### 2.1 VBA-ohjelmointikieli yleisesti

Visual Basic for Applications –ohjelmointikieli mahdollistaa erilaisten toimintasarjojen suorittamisen Excel –ohjelmassa halutulla tavalla. Erilaisten makrojen eli toimintasarjojen käynnistämällä voidaan hallinnoida toimintoja Excelissä.

Makrot soveltuvat hyvin käyttökohteisiin, joissa toistuvat rutiininomaiset tiedonsiirto- ja jalostustoimet. Esimerkiksi taloushallinnon viikko- ja kuukausiraporttien luonti, jossa systemaattisesti samasta paikasta haetaan tietoa ja analysoidaan sitä tietyllä ajanjaksolla. Kyseisen toimenpiteen suorittaminen manuaalisesti ilman makroja on työlästä ja samalla kasvaa virheiden määrä. Toinen hyvä esi-



merkki makrojen soveltamisesta on suuren data-aineiston järjestäminen selkeään muotoon jatkojalostusta varten. Datan hyödyntäminen on hankalaa, jos yksittäiset tiedot on ripoteltu eri tiedostoihin. Makro toimii aina samalla tavalla, joten se ei tee virheitä. Tiedonhaussa voi makroista huolimatta tapahtua virheitä, jos tiedosto on nimetty väärin, yksittäisen tiedoston tyyppi on väärä tai oleellisia tietoja puuttuu. Mahdollisista tietokonevirheistä huolimatta rutiininomaiset suoritukset kannattaa automatisoida makrojen avulla, koska ihminen tekee varmasti virheitä toisinaan. (Leino 2016, 160.)

On olemassa muutamia toimintoja Excelissä, joihin ei kannata soveltaa makrojen toimintaa. Excelissä olevien toimintojen suorittaminen makrojen avulla ei tuo juurikaan lisäarvoa, koska ominaisuudet ovat jo olemassa. Käyttöliittymän rakentaminen Exceliin makrojen avulla. Käyttöliittymä on itsessään varsin hyvä, joten sen tekeminen uudelleen on turhaa. Tulostietojen laskeminen raporttiin makrojen avulla, pivot ja kaavat ovat käyttökelpoisempi tapa suorittaa tulostietojen laskeminen. (Leino 2016, 160-161.)

Excel-työhön voidaan lisätä automaattista toimintaa makrojen avulla. Makrot eli tietokoneohjelmat käynnistyessään suorittavat ennalta määritetyn toimintasarjan. Puhekielessä käytetään termiä makro, mutta makron sijaan voitaisiin myös käyttää termejä proseduuri tai ohjelma. (Leino 2016, 161.)

### **2.1.1 Makrojen rakenne**

Makro rakentuu muutamien elementtien yhdistelmällä. Näitä elementtejä käyttäen muodostetaan yhtenäinen kokonaisuus, jossa saadaan suoritettua jakautuminen eri toiminnallisuuksiin, suoritettua aritmeettisia operaatioita, tallennettua tietoa erilaisiin tietoalkioihin unohtamatta ohjelmasuorituksen selostusta kommenttien avulla. Ohjelmointikielinen koodi koostuu seuraavista osista:

- Avainsanat, esimerkiksi Let, Function, If.
- Operaattorit: +, =, &, (.
- Tietoalkioiden nimet eli tunnukset.
- Ohjelmoijan tekemä kuvaus eli kommentit.

Avainsanojen avulla voidaan luoda kontrollirakenteita, joita suoritetaan tietyillä ehdoilla. Esimerkiksi taulukon alkioden läpi käyminen tai tarkistaminen voidaan suorittaa erilaisten silmukkarakenteiden avulla. Silmukkarakenteiden luomiseen käytetään avainsanoja ehdollistamaan tapahtumia. Erilaisilla rakenteilla voidaan määrätä ohjelman toimintaa. Rakenteista puhutaan myöhemmin tässä kappaleessa. Ohjelman algoritmi rakennetaan erilaisten kontrollirakenteiden avulla. Kontrollirakenteita on kolme eri tyyppiä:

- Peräkkäisyys, joka tarkoittaa ohjelmakoodin suorittamista rivi kerrallaan ja suoritus tapahtuu ylhäältä alas. Koodin suorittaminen ei voi kuitenkaan olla ristiriidassa kahden muun kontrollirakenteen kanssa.
- Valinta, jossa haaraututaan ehtojen mukaiseen suoritukseen. Tässä kontrollirakenteessa tulee mukaan If- ja Select Case-rakenteet.
- Toisto: määrätyn silmukan tai toiminnon suorittaminen, kunnes ehto ei täyty tai täytty. Silmukka-rakenne voidaan suorittaa erilaisilla toiminnoilla tilanteen mukaan.

Lausekkeiden avulla suoritetaan erilaiset aritmeettiset toimenpiteet. Aritmeettiset lausekkeet voivat sisältää hyvinkin monimutkaisia laskutoimituksia. Lauseke palauttaa yhden lukuarvon suorituksen jälkeen. Esimerkiksi laskemalla yhteen kahden solun sisältö saadaan lopputuloksena solujen sisältöjen summa. Aritmeettisten operaatioiden suorittaminen tapahtuu hyödyntäen operaattoreita lausekkeissa. Lauseke koostuu muuttujista, vakioista, funktioista ja toimituksen määrävistä operaattoreista. (Leino 2016, 161-166.)

Tietoalkioiden nimeksi eli tunnukseksi kannattaa valita jokin sitä hyvin kuvaava nimitys. Tunnuksia on jaettu niiden käytettävyyden mukaan kahteen pääryhmään. Niiden ero on merkittävä ohjelman toiminnan kannalta, koska lokaaleita tunnuksia ei voi kutsua muista ohjelmista. Lokaaleita tunnuksia voidaan käyttää vain sen makron sisällä, johon ne on luotu. Globaalit tunnukset ovat voimassa kaikissa kyseisen moduulin sisällä olevissa makroissa. (Leino 2016, 166-167.)

Ohjelmakoodin kommentointi erotetaan suoritettavasta koodista puolilainausmerkillä tai vaihtoehtoinen tapa erottaa kommentit varatulla sanalla "rem". Kom-

mentointi tapahtuu rivikohtaisesti. Puolilainausmerkki aloittaa kommentin ja kommentti kattaa loppurivin koodista. Kommenttien vihreä väri helpottaa sen erottamista varsinaisesta ohjelmakoodista. Ohjelmakoodi ei luonnollisesti suorita kommentoituja tekstejä. Kommentit ovat tärkeässä roolissa ohjelmakoodin ymmärrettävyyden kannalta ja siten koodia on helpompi muokata jälkikäteen, kun tiedetään mitä toimintoja kyseinen rivi suorittaa. (Merensalmi 2007, 23.)

### 2.1.2 Ohjausrakenteet

Ohjausrakenteiden avulla voidaan suorittaa monipuolisempia operaatioita kuin makrojen nauhoituksella. Makrojen nauhoitukseen sisältyy ainoastaan asetuslause. Erilaiset ehtolauseet, silmukkarakenteet ja hyppyt kuuluvat myös ohjausrakenteisiin asetuslauseiden lisäksi. Ohjausrakenteiden avulla voidaan välttää paljon manuaalista työtä, jota jouduttaisiin tekemään ilman ohjausrakenteita. (Merensalmi 2007, 77.)

Aetuslauseen avulla määritetään muuttujien arvo koodin suorituksen aikana. Aetuslause mahdollistaa solun sisällön siirtämisen muuttujaan, jotta sitä ei tarvitse toistuvasti poimia koodin suorituksen aikana. Aetuslauseen syntaksissa yhtäsuuruusmerkin vasemmalla puolella on muuttuja, johon tietoa tallennetaan. Yhtäsuuruusmerkin oikealla puolella voi olla esimerkiksi laskutoimitus, jonka lopputulos tallennetaan muuttajan arvoksi. Esimerkiksi Excelin soluun ei voida kirjoittaa kaavaa =A1+3, vaan siitä seuraa virheilmoitus. Ideana olisi siis lisätä solun A1 arvoon luku kolme. Kaavan 1 mukainen esimerkki on kuitenkin tyypillinen tapa suorittaa luvun summaus solun arvoon koodissa:

$$A1=A1+3 \qquad (1)$$

Yhtäsuuruusmerkin oikea puoli suoritetaan kääntäjässä ensimmäisenä, jonka jälkeen A1 soluun tulee uusi arvo, joka on edellinen solussa A1 oleva arvo summatuna luvulla 3. Aetuslauseessa on mahdollista käyttää myös Excelin valmiita vakioita esimerkiksi Date-funktiota, jonka avulla saadaan siirrettyä muuttujaan päivämäärä. (Merensalmi 2007, 78.)

Today-funktio on generointityökalun yhteydessä käyttökelpoinen funktio, jolla voidaan automaattisesti viedä CADs-piirikaavion nimiöön tieto piirikaavion luomispäivämäärästä. Tämän funktion käyttämiseen liittyy kuitenkin pieni ongelma, johon palataan generointityökalun kehityksen yhteydessä.

Asetuslauseiden käytössä on syytä kiinnittää erityistä huomiota käytettävään tietotyyppiin. Väärällä tietotyypin valinnalla voidaan kadottaa dataa ilman, että käyttäjä huomaa sitä. Liukuluvun tallentaminen muuttujaan, jonka tyyppi on kokonaisluku, kadottaa osan datasta. Esimerkkitilanteessa piin likiarvo siirretään ensin muuttujaan M, jonka tietotyyppi on Double. Tämän jälkeen muuttujan M arvo siirretään muuttujaan N, jonka tietotyyppi on Long. Kääntäjän mielestä kaikki sujuu ongelmitta, mutta desimaalit eivät mahdu muuttujaan N väärän tietotyypin takia. Näin voidaan menettää huomaamatta olennaista tietoa myöhempiä toimia ajatellen. (Merensalmi 2007, 78.)

Ohjausrakenteiden ehtolauseet tulee ohjelmoijalle vastaan, jos tarvitaan jonkinlaista ehdollisuutta ohjelman rakenteessa. Ehtolauseessa on kyse haarautuvasta ohjelmarakenteesta, jossa vertaillaan yhden tai useamman ehdon toteutumista ja sen perusteella toteutetaan koodia. Ehtolauseen toteuttamiseen on useita tapoja ehtolauseen monimutkaisuuden mukaan. (Merensalmi 2007, 80.)

Yksinkertaisin ehtolause voidaan kirjoittaa yksirivisellä If-lauseella, jossa jonkin asian ollessa tosi suoritetaan koodi ja muussa tapauksessa ei. Yksinkertaisimmassa ehtolauseessa ei voida tehdä vertailua usean eri tapauksen suhteen ja tästä syystä ehtolauseita usein laajennetaan Else-ehdolla, jonka ansiosta tapahtuu haarautuminen. Jos If-lause on tosi, suoritetaan sen mukainen koodi ja muussa tapauksessa toteutetaan Else-lauseen mukainen koodi. Tämä on ohjelmoinnissa yleisin ehtolause, koska monesti täytyy huomioida Else-osa vaihtoehtojen tilanteiden takia. (Merensalmi 2007, 82.)

Monimutkaisemmissa lausekkeissa voidaan edellisten avainsanojen lisäksi käyttää Elseif-komentoa, joka mahdollistaa useita eri variaatioita ehtolausekkeeseen. Elseif-lausekkeita voidaan lisätä rajaton määrä ehtolausekkeeseen. Elseif-lau-

sekkeen avulla on helppo esimerkiksi määrittää arvosana kokeen tuloksen mukaisesti. (Merensalmi 2007, 82.) Alla on esitetty koodi, joka määrittää arvosanan saatujen pisteiden mukaan.

```

Public Function PalautaLuokka (ByVal Luku As Long) As String
1      Dim OutStr As String
2      If Luku < 20 Then
3          OutStr = "Hylätty"
4      ElseIf Luku < 40 Then
5          OutStr = "Tyydyttävä"
6      ElseIf Luku < 60 Then
7          OutStr = "Hyvä"
8      ElseIf Luku < 80 Then
9          OutStr = "Kiitettävä"
10     ElseIf Luku < 101 Then
11         OutStr = "Erinomainen"
12     Else
13         OutStr = "Virhe"
14     End if
15     PalautaLuokka = OutStr
End function

```

Ehtolauseen rakentaminen tulee tehdä siten, että pienimpään lukuun verrataan ensimmäisenä ja suurimpaan viimeisenä. Jos ensimmäinen tarkastelu olisi luvun tarkistaminen suurimman luvun suhteen, ehto toteutuisi ja arvosana määräytyisi väärin. Ohjelma ei kyseessä olevassa tilanteessa anna virheilmoitusta, koska ensimmäinen ehto on tosi ja ohjelmallisesti ehtolause toimii oikein. Käytännössä vertailu pienempiin lukuihin jäisi suorittamatta ja erinomainen arvosana saavutettaisiin alhaisilla pisteillä. Tästä syystä ehtojen järjestyksellä on ratkaiseva merkitys halutun toiminnan kannalta. (Merensalmi 2007, 82.)

Ehtolauseen kasvaessa vaihtoehtojen ansiosta tulee usein järkevämmäksi käyttää Select Case-rakennetta, joka on vaihtoehtoinen tapa suorittaa If-lause. Select Case-rakenteen avulla säästetään ohjelman pituudessa ja esitystapa on hieman

selkeämpi. Yllä esitetty arvosanan määräytyminen voidaan Select Case-rakenteen avulla suorittaa alla olevan koodin mukaisesti. (Merensalmi 2007, 83.)

```

Public Funktion PalautaLuokkaSelectCase( _ByVal Luku As Long)
As String
1      Dim OutStr As String
2      Select Case Luku
3      Case Is < 20: OutStr = "Hylätty"
4      Case 20 To 39 : OutStr = "Tyydyttävä"
5      Case 40 To 59 : OutStr = "Hyvä"
6      Case 60 To 79 : OutStr = "Kiitettävä"
7      Case 80 To 100 : OutStr = "Erinomainen"
8      Case Else: OutStr = "Virhe"
9      End Select
10     PalautaLuokkaSelectCase = OutStr
End Function

```

Huomataan, että vaihtoehtoisella syntaksilla säästetään 5 riviä koodin kirjoitusta. Koodi on myös kompaktimman näköinen kokonaisuus kuin If-Else-rakenteella tehty vertailu. (Merensalmi 2007, 83.)

Silmukkarakenne on kätevä tapa toteuttaa datan tai Excelin solujen läpikäyminen, jossa samaa koodia tai toiminnallisuutta suoritetaan kerta toisensa perään silmukan ehtojen mukaisesti. Silmukkarakenteet voidaan jakaa toiminnallisessa mielessä kolmeen osaan. Ensimmäistä silmukkatyyppiä voidaan kutsua kiinteäksi silmukaksi. Nimi tulee siitä, että ennen silmukan käynnistämistä on tiedossa kiinteä aloitus- ja lopetuskohta. Tämä silmukkatyyppi soveltuu hyvin esimerkiksi Excelin tietyn sarakkeen läpikäymiseen. Esimerkiksi alkaen solusta B10 ja päättyvän soluun B100. Toista silmukkatyyppiä kutsutaan esiehtoiseksi silmukaksi, jossa aloitus- ja lopetuskohta on tuntemattomia. Esiehtoisella silmukalla voidaan etsiä esimerkiksi ensimmäinen tyhjä solu tietyistä sarakkeista. Tällöin ei ennalta tiedetä silmukan suorituskertoja. Viimeistä silmukkarakennetta kutsutaan lopetusehtoiseksi silmukaksi. Esiehtoiseen silmukkarakenteeseen verrattuna lopetusehtoinen silmukkarakenne suoritetaan ainakin kerran ennen kuin lopettamisehto tarkistetaan. Lopetusehtoinen silmukka toimii pitkälti samalla tavalla kuin

esiehtoinen silmukka, mutta niiden ensimmäisessä toteutuskerrassa on ero. (Merensalmi 2007, 84.)

### 2.1.3 Ohjelman hallinnointi

Etenkin suuremmissa ohjelmakokonaisuuksissa ohjelman jakaminen pienempiin kokonaisuuksiin on tärkeää. Ohjelman jakaminen pääohjelmiin (Public) ja aliohjelmiin (Private) on tärkeää myös sen takia, että aliohjelmat käynnistyvät hallitusti pääohjelman kutsuessa niitä. Pääohjelmat ja aliohjelmat ovat järkevää jakaa toiminnallisiin kokonaisuuksiin. Joskus aliohjelman käynnistäminen ilman pääohjelmaa voi olla jopa haitallista. Aliohjelmien käyttämisellä säästetään myös työtä ja aikaa, kun yleiset aliohjelmat voidaan tallentaa samaan moduuliin, jota voidaan käyttää kaikissa projekteissa. Samaa aliohjelmää ei tarvitse tehdä kahteen kertaan, koska yleiset aliohjelmat ovat kaikkien moduulien käytössä. (Merensalmi 2007, 53.)

### 2.1.4 Nimeämiskäytännöt

Muuttujien ja ohjelmien nimeämisessä on muutamia sääntöjä. Excel VBA –ohjelmointiympäristö varaa omaan käyttöönsä muutamia sanoja, joita ei voi käyttää ohjelmien tai muuttujien nimeämisessä. Varattuja sanoja ovat esimerkiksi tietotyypit kuten: Long, Double, Boolean, Date ja String. VBA sisältää myös joitakin valmiita funktioita, joiden nimiä ei voi käyttää, esimerkiksi Calendar, Error, Fix, Kill, Minute ja Year. (Merensalmi 2007, 56.)

Varattujen nimien lisäksi on muutamia muita nimeämiseen liittyviä rajoitteita. Tiettyjen erikoismerkkien käyttäminen ei ole sallittua muuttujien nimeämisessä. Esimerkiksi välilyönti sanojen välissä tulkitaan kahdeksi erilliseksi muuttujaksi, jolloin siitä tulee virheilmoitus. Myöskään pistettä, pilkkua tai huutomerkkiä ei saa sisällyttää muuttujien nimissä. Sen sijaan alaviiva on sallittu merkki ja sillä on hyvä jakaa pitkiä muuttujien nimiä helpommaksi lukea. Skandinaaviset kirjaimet ovat koodissa sallittuja, joten niitä on mahdollista käyttää vapaasti. Muuttujan nimi täytyy alkaa kirjaimella. Jos muuttujan nimen aloittaa numerolla, VBA-editori tekee

muuttujan alussa olevasta numerosta leiman lukua vastaavalle riville. Muuttujan nimi voi muuten sisältää numeroita, mutta ensimmäinen merkki ei voi olla numerolla mainitun syyn takia. Esimerkiksi nimi "Testi4" on hyväksytty nimeäminen, mutta "4Testi" ei ole. Muuttujan nimeäminen ja esittely voidaan suorittaa vain kerran aliohjelman aikana. Jos muuttujaa ei esitellä, ensimmäinen viittaus tekee automaattisen alustuksen muuttujalle. Tämän jälkeen muuttujaan viittaaminen koodissa osoittaa samaan muistipaikkaan. Kääntäjä ei tee eroa pienten ja isojen kirjaimien välillä. Esimerkiksi aliohjelman tallentaminen nimellä "haku" estää tekemästä samaan moduuliin aliohjelman nimellä "HAKU". Kääntäjä tulkitsee aliohjelmat samannimisiksi isoista kirjaimista huolimatta. (Merensalmi 2007, 56-57.)

## 2.2 CADS-tuoteperhe

CADS on suomalaisen Kyndata Oy:n kehittämä teollisuuden aloille suunnattu ohjelmistotuoteperhe. Yritykseltä löytyy sähkö- ja automaatio suunnitteluun, LVI-suunnitteluun sekä talojen arkkitehti- ja rakennesuunnitteluun omia räätälöityjä sovelluksia. CADS-tuoteperheestä löytyy sovellukset seuraaville osa-alueille:

- Sähkö- ja automaatio
- LVIA
- Kone
- Rakennus
- Prosessikaaviot
- Kylmäteknikka
- Profiilirakenteet
- Maanmittaus (Geotekniikka)

(Kyndata Oy 2019a)

CADS Electric on kokonaisvaltainen sähkö- ja automaatio suunnittelun ohjelmisto. Suunnitteluohjelmistolla voidaan suunnitella esimerkiksi rakennussähköistykseen, teollisuusautomaatioon, keskusten rakenteeseen sekä jakeluverkkoon liittyvät kuvat. CADS Electric –sovelluksesta löytyy alan tarpeisiin soveltuva kirjasto erilaisia valmiita symboleita, jotka helpottavat kuvien piirtämistä. Omien



symbolipakettien luonti on myös sovelluksessa mahdollista, jolloin voidaan luoda valmiita paketteja yleisesti käytetyistä kokonaisuuksista. CADS Electric –sovelluksesta löytyy ominaisuudet seuraaviin sähkö- ja automaatiotarpeisiin:

- Laitteiden ja koneiden ohjaukset
- PLC ja automaatiojärjestelmät
- Instrumentointi
- Kaapit, keskuksat ja kotelot
- Tehdaslayoutit ja reititykset
- Tiedonhallinta
- Järjestelmäintegraatiot
- Dokumenttienhallinta

Tietokannan käyttö on projektin hallintaan hyvä ominaisuus, jolloin esimerkiksi I/O-kortit voidaan luoda kaavioon ennen kuin järjestelmän valinta on tehty. I/O-korttien osoitteet voi jättää suunnitteluvaiheessa tyhjiksi ja järjestelmän valinnan jälkeen määrittää osoitteet tietokannassa, jolloin osoitteet päivittyvät kuviin. Näin saadaan piirikaavioiden suunnittelu tehokkaasti käyntiin projektin alkuvaiheessa. Kenttälaitteiden toiminta tiedetään pääpiirteittäin jo kenttälaitteiden valintojen yhteydessä, joten piirikaaviot voidaan luoda valmiiksi vielä järjestelmän kilpailutuksen aikana. I/O-tiedot voidaan jälkeinpäin tuoda Excelistä CADS –suunnitteluohjelmaan, jolloin piirikaavion puuttuvat tiedot päivittyvät ja piirikaaviot saadaan nopeasti valmiiksi. Datan siirto on kaksisuuntaista Excelin ja CADS –suunnitteluohjelman välillä. Tämä mahdollistaa piirikaavioiden suunnittelun todellisilla I/O-osoitteilla ja niiden tuomisen Exceliin Export-toiminnon avulla. Näin saadaan valmis lista järjestelmän osoitteista esimerkiksi logiikkaohjelman suunnittelua varten. (Kyndata Oy 2019a)

CADS Electricissä keskusten, koteloiden, kaappien ja ohjauspaneelien suunnittelu voidaan tehdä luonnollisilla mitoilla, jolloin on helpompi jo suunnitteluvaiheessa valita optimaalisesti keskuksen tai kotelon suuruus. (Kyndata Oy 2019a) Tämä helpottaa myös keskuksen kokoonpanovaiheessa, koska johdinkourujen ja laitteiden mittasuhte on luonnollisessa koossa. Jos mittasuhte suunnitteluvaiheessa on väärin, komponenttien mahduttaminen keskukseseen on hankalaa.

Myös keskuksen käyttöönotossa johtojen kytkemiselle varattu tila on helpompi huomioida suunnittelun yhteydessä.

### 2.2.1 Piirto-ominaisuudet

Piirto-ominaisuuksissa käsitellään erilaisia työkaluja, joita voidaan hyödyntää piirrettäessä muotoja kuvaan tai komponenttien sijoittamisessa. Työkalujen hyöty korostuu hyvin kuvien muokkaamisessa jälkikäteen. Lisäksi kuvien kojeet ja johdotukset osuvat absoluuttisesti oikeaan kohtaan kuvassa ja kuva on esteettisesti siistin näköinen.

Rasteri eli pakotuspisteytyksellä tarkoitetaan tulostuvaa pisteverkkoa, jossa hiusristikko liikkuu rasterin ollessa käytössä. (Kymdata Oy 2007) Rasterijaon suuruutta voidaan vaihtaa asetuksista, mutta jaollisuus on tärkeä. Rasterin käytössä jaollisuudella saadaan kojeiden kytkentäpisteet osumaan muihin kojeisiin kohdalleen. Esimerkiksi rasterilla piirretty kuva, minkä rasterijako on 3,5 osuu kohdalleen rasterijaon ollessa 1,75. Jaollisuus merkitsee, että 3,5 askellus on jaollinen pienemmällä rasterijaolla ja osamääränä on kokonaisluku. Rasterin käytöllä johdotuksista tulee suoria ja kytkentäpisteet osuvat kuvaan piirretyille elementeille tasaisesti.

Pakotustyökalu on kätevä ominaisuus, kun halutaan piirtää elementtejä kuvaan tiettyssä kulmassa. Pakotustyökalun kulman suuruutta voidaan muuttaa. Esimerkiksi piirikaavioissa on yleisesti käytössä pakotustyökalun kulmana 90°. Pakotuksen ansiosta viivat ja komponentit piirtyvät automaattisesti vaaka- tai pystysuoraan. Toki esimerkiksi viivan kulman voi määrittää myös komentoikkunaan syöttämällä, mutta kätevämpi tapa viivojen piirtoon on pakotustyökalun käyttö. (Kymdata Oy 2019b)

Snap-työkalun ominaisuuksien avulla voidaan sijoittaa viiva tai johdinviittaus keskelle tiettyä symbolia. Snapin käyttö ohittaa aiemmin esitetyn rasterijaon. Jos esimerkiksi snap-toiminnolla asetetaan johdinviittaus keskelle johdinta, johdinviittaus ei välttämättä osu rasterijakoon. Jos johdin on sopivan pituinen ja rasterijako osuu sopivasti, snapilla asetettu johdinviittaus voi osua myös rasterijakoon. Tästä

syystä snap-työkäluu tulee käyttää harkinnan mukaan. Usein on kuitenkin siistimpää asettaa johdinviiittaus johtimen keskikohtaan, mutta asia on tapauskohtainen. Tartuntapisteen osumisella johdinviiittauksiin ei sinänsä ole merkitystä, koska siitä pisteestä ei piirretä viivaa jatkuvaksi. Kojeiden sijoituksessa snapin käyttäminen ei ole suositeltavaa edellä mainitun syyn takia. (Kymdata Oy 2019b)

Seulan voidaan kuvailla toimivan suodattimen tavalla, jolla haravoidaan luodusta kuvasta vain tietynlaiset elementit. Seula valitaan käyttöön elementtikohtaisesti esimerkiksi tekstiin tai viitteisiin. Jos seulaan valitaan teksti, muokkaus kohdistuu pelkästään tekstielementteihin. Muita elementtejä voidaan luoda niiden omilla komennoilla, vaikka seula on käytössä. (Kymdata Oy 2007)

### **2.2.2 Tietokanta**

CADS Electric tiedonhallinnan avulla muutos yhteen kuvaan päivittyy myös muihin kuviin ja tietoihin. Ominaisuus vaatii, että projekti on lisätty tietokantaan (DB). Automaattinen ominaisuus vähentää suunnittelijan työtä, koska yhden tiedon muuttaminen kaikkiin kuviin on varsin työlästä ja kasvattaa riskiä ristiriitojen syntymiseen. Lisäksi samaan projektiin kuuluvalla tiimillä on käytössä yhteiset tuote-tietokannat, projektin laitteet, projektikohtaiset asetukset ja tuotemallikirjastot. Näin varmistutaan siitä, että suunnittelu tapahtuu samoja komponentteja ja asetuksia käyttäen kaikkien suunnittelijoiden toimesta. (Kymdata Oy 2019b)

Johdotusten suunnittelu laitoksen sisällä luonnollisessa koossa mahdollistaa ryhmien pituuksien, oikosulkuvirtojen, tehojen ja jännitealenemien sähköteknisen laskennan automaattisesti. Sovellus antaa suunnittelijalle ilmoituksen ryhmien pituuden, oikosulkuvirran ja jännitteenaleneman ylittäessä rajoitukset. Näin voidaan ennakoivasti vaihtaa kaapelityyppiä tai johdotusreittiä ennen kuin laitosta aletaan ottaa käyttöön. Ominaisuus säästää valtavasti kustannuksia, kun johdotukset tehdään kerralla oikein käyttäen tilanteen vaativia materiaaleja. Jännitteenalenemien ja oikosulkuvirtojen laskeminen isojen laitosten kohdalla on valtava työ ilman valmiiden kaavojen käyttämistä esimerkiksi Excel –taulukkolaskentaoh-

jelmallä. Johdotusreitien pituuden kasvaessa tilanteeseen voidaan reagoida esimerkiksi rakentamalla johdotusreitien väliin erillinen ristikytkentäkotelo, jossa kaapelia jatketaan tai signaalia vahvistetaan. (Kymdata Oy 2019b)

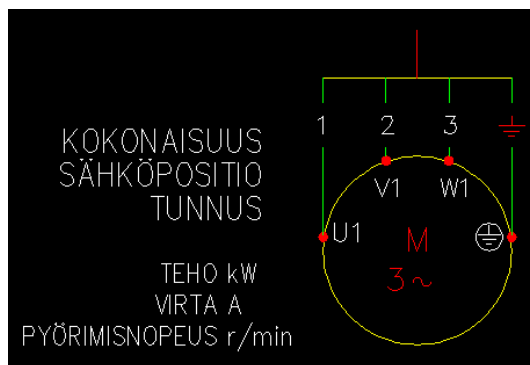
### 2.2.3 Generointilinkit attribuutteihin

Symbolien tiedot saadaan auki klikkaamalla symbolia ja valitsemalla ominaisuudet. Kuvan viereen aukeaa taulukko, jossa on kyseisen symbolin tiedot. Ominaisuuksista löytyy yleiset tiedot, joista ilmenee esimerkiksi taso ja viivan leveys. Geometriasta löytyy symbolin sijainti tason koordinaatistosta ja kokokertoimet eri akseleilla. Symboli kertoo koheen symbolitunnuksen. Symbolitunnuksen avulla voidaan varmistua, että jokaisessa kuvassa käytetään samaa symbolia, koska symbolitunnus on yksilöllinen tunniste. Ominaisuuksista löytyy myös kyseisen symbolin attribuutit, joihin voidaan sijoittaa koheen tietoja. Tiedot vaihtelevat koheiden mukaan. Moottorin attribuutteja ovat esimerkiksi virta, teho ja pyörimisnopeus. Attribuutteja voidaan syöttää manuaalisesti suoraan tai generointilinkkien avulla. Generointilinkkeihin palataan myöhemmin tässä kappaleessa. Symbolin ominaisuuden on esitetty kuvassa 1.

Ominaisuudet	
SYMBOLI	
Yleiset	
Väri	1
Taso	SLEHTI1
Elem. läpinäkyvyys	TASO
Viivatyyppi	EHYT
Viivanleveys	OLETUS
Geometria	
X-koordinaatti	112
Y-koordinaatti	-3,5
Z-koordinaatti	0
Kulma	0°
Kokokerroin X	1
Kokokerroin Y	1
Kokokerroin Z	1
Normaali X	0
Normaali Y	0
Normaali Z	1
Symboli	
Symboli	SRO07001
Attribuutit	
Tunnus:	TUNNUS
Teho:	TEHO
Virta:	VIRTA
Pyörimisnopeus:	PYÖRIMISNOPEUS
Kokonaisuus:	KOKONAISSUUS
Sähköpositio:	SÄHKÖPOSITIO

KUVA 1. Symbolin ominaisuudet

Kuvan 1 esimerkissä attribuutteihin on syötetty linkit manuaalisesti. Osaan attribuuteista tulee yksikkö automaattisesti näkyviin. Esimerkiksi sähköisiin suureisiin tulee moottorin symbolissa yksiköt tehon, virran ja pyörimisnopeuden yhteydessä. Kaikkiin sähköisiä suureita sisältäviin kojeisiin yksikkö ei tule automaattisesti mukaan. Sulakkeiden Sähkö1-attribuutti ei sisällä yksikköä, vaikka virran suureena on ampeeri. Yksiköiden kanssa tulee olla tarkkana syötettäessä attribuuttien tietoja. Attribuuttien esiintyminen symbolin yhteydessä on esitetty kuvassa 2.



KUVA 2. Moottorin attribuutit

Attribuuttien sijainti suhteessa kojeeseen vaihtelee hieman. Kuvan 2 esimerkissä attribuutit sijoittuvat symbolin vasemmalle puolelle. Joissakin tapauksista attribuutit voivat jakautua molemmiin puoliin kojetta. Toiminnan kannalta asialla ei ole suurta painoarvoa, mutta esitystavan on hyvä olla yhtenevä muiden kuvien kanssa.

Generoinnissa tarvittavat linkit sijoitetaan attribuutteihin samalla nimellä kuin tietoa vastaava sarake on esitetty generointityökalun Excel-tiedostossa. Generointilinkille on oma yksilöllinen rivinumero generointitaulukossa, josta se osataan hakea generoinnin yhteydessä. Rivinumero sijoitetaan hakasulkeisiin nimen perään. Generointilinkissä on siis kaksi osaa, joiden avulla tieto haetaan generoinnissa. Nimi kertoo sarakkeen, josta tieto haetaan ja hakasulkeiden sisällä oleva numero kertoo rivinumeron. Näin saadaan määriteltyä tietty solu Excel-tiedostossa. Näiden lisäksi nimi ja numero ympäröidään \$-merkeillä. Alla on esitetty esimerkki generointilinkistä, joka viittaa LAITE TEHO-sarakkeeseen Excelissä rivillä 1.

## \$LAITE TEHO[1]\$

Sijoittamalla esitetty linkki moottorin attribuuttiin kohtaan "Teho:" saadaan laitteen teho haettua Excel-taulukosta oikein. Moottoriin liittyvät muut tiedot saadaan vaihtamalla rivinumeroa edeltävää tekstiä. Yhden laitteen tiedot ovat tyypillisesti samalla rivillä. Samalla rivillä on moottorin tietojen lisäksi esimerkiksi moottorin syöttökaapelin tunnus ja tyyppi sekä signaalikaapelin tunnus ja tyyppi. Syöttökaapelilla tuodaan moottorille käyttöjännite ja signaalikaapeli on kytketty esimerkiksi termistoriin, joka mittaa moottorin käämien virtaa. Syöttökaapelin jännitetaso vaihtelee moottorin mukaan, mutta syöttökaapelin jännitetaso voi olla esimerkiksi 400 VAC ja tyyppinä MCMK 3x6+6. Syöttökaapelin merkintä tarkoittaa, että jokaisen vaihejohtimen poikkipinta-ala on  $6 \text{ mm}^2$  ja suojajohtimen poikkipinta-ala on myös  $6 \text{ mm}^2$ .

Rivinumeroinnissa käytettiin laadittua taulukkoa, johon on jaoteltu erilaisiin käyttökohteisiin ja laitteisiin liittyvät numeroinnit. On huomattavaa, että sama rivinumero esimerkiksi venttiileissä ja koneissa tarkoittaa eri asiaa. Generointikirjaston kuvien generointilinkeissä on hyödynnetty taulukon numerointia. Taulukko on esitetty tämän dokumentin liitteessä 1.

### 3 ONGELMIEN KARTOITUS

Ongelmien kartoituksessa pyrittiin löytämään keskeisiä ongelmia suunnitteluun liittyen ja löytämään uusia näkemyksiä tehokkaampaan suunnitteluun. Haastatteluiden avulla pystytään kehittämään havaittuja ongelmakohtia paremmin.

#### 3.1 Ongelmien kerääminen

Ongelmien kartoittamiseksi pidettiin lyhyitä haastatteluita ja soitettiin puhelimitse eri toimipisteellä toimiville suunnittelijoille. Kysymykset oli laadittu siten, että saadaan hyvä kokonaiskuva tietokannan ja kuvien generoinnin käyttöasteesta. Haastatteluja suoritettiin yhteensä kahdellekymmenelle suunnittelijalle. Insta Automation Oy:n suunnitteluosastolle tullaan tekemään opetusvideoita CADS-ohjelman yleisistä toiminnoista, jotka koetaan haasteellisiksi tai niihin halutaan lisäperehdytystä. Opetusvideoiden tarpeiden kartoitus integroitiin haastatteluihin, koska osa toiveista liittyi nimenomaan generointityökalun esittelyyn.

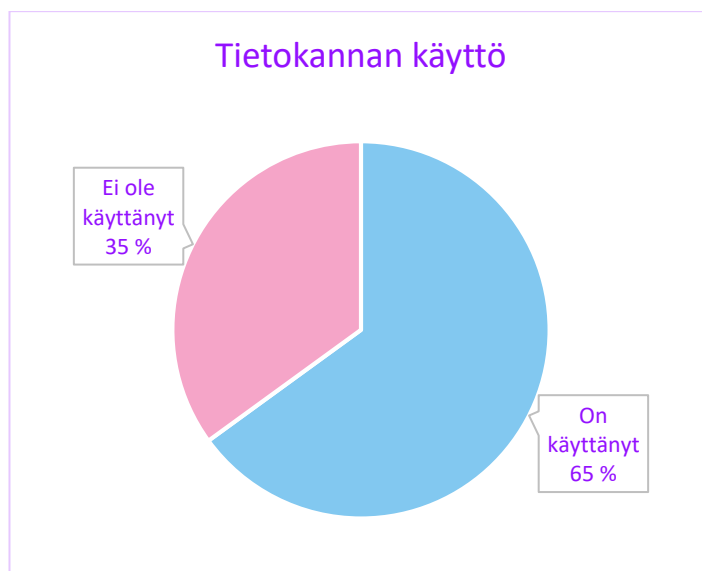
Haastatteluissa suunnittelijat saivat tuoda vapaasti toiveitaan esille opetusvideoihin liittyen. Lähtökohtana on se, että opetusvideoita tehdään tarpeeseen. Haastatteluissa nousi paljon hyviä mielipiteitä, joiden avulla tulevia opetusvideoita luodaan. Opinnäytetyön tarkoitus on kehittää suunnittelua ja ratkaista koettuja ongelmia sekä luoda jotain uutta, mikä tehostaa suunnittelua entisestään. Datan kerääminen suunnittelijoiden kokemuksista on tärkeässä roolissa kehitystyötä. CADS-generointi nykyisellä tavalla koettiin hankalaksi käyttää sekä sen soveltuminen pienempiin projekteihin oli suunnittelijoiden näkemysten mukaan heikko, koska generointityökalun opettelu ja generointitaulukon lähtötietojen täyttäminen vievät aikaa ja pienissä projekteissa aika on rajallista. Lisäksi nykyinen generointityökalu ei toimi uusimmalla Windows 10-käyttöjärjestelmällä, joten tietojen generointi täytyy suorittaa virtuaalikoneella.

Nykyinen generointityökalun hyöty tulee esille suuremmissa kokonaisuuksissa, jossa on hyvät lähtötiedot. Generoinnin ideana on automaattinen tietojen hakeminen eri Excel-tilukoista ja niiden yhdistäminen pohjakuvaan. Jos lähtötiedot

ovat puutteelliset, joutuu suunnittelija syöttämään tietoja paljon manuaalisesti. Itse generointityökalun saattaminen generointia varten valmiiksi vie melko paljon työaikaa, mutta suurissa projekteissa generointityökalu tuottaa valmiita piirikaavioita huomattavasti nopeammin kuin jokaisen kuvan luominen yksilönä. Lisäksi generointityökalu eliminoi inhimillisen virheen määrää kuvissa. Esimerkiksi nimiöiden korjaaminen manuaalisesti useaan kuvaan on työlästä, mutta Excelin avulla tietojen vaihtaminen toimii kätevästi.

### 3.2 Kerätyn datan analysointi

Kerätyn datan perusteella saatiin tietokannan ja generoinnin käytöstä kattava käsitys. Generointi on ollut yleisesti käytössä suurissa projekteissa, joissa kuvat ovat samankaltaisia. Kuvien erona on ollut esimerkiksi I/O-pisteet sekä laitteiden positiot. Näihin projekteihin generointi on luonut merkittävää lisäarvoa nopeuden ja inhimillisen virheen poistaminen ansiosta. Kerätyn datan perusteella muodostettu tietokannan käyttöaste on esitetty kuviossa 1.



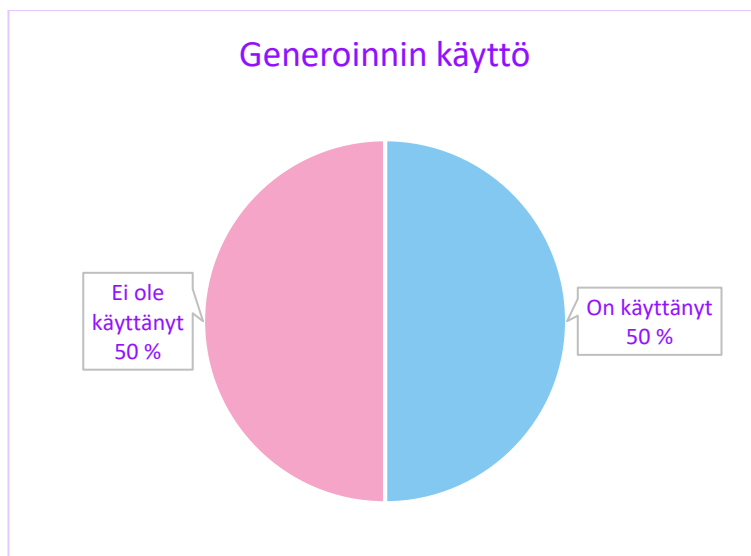
KUVIO 1. Tietokannan käyttöaste

Kerätyn datan perusteella noin kolmannes ei ole käyttänyt CADS-tietokantaa suunnittelun tukena. Yleisesti tietokantaa on käytetty kaapelitietojen ja johdotus-taulukon hakemiseen sekä osaluettelon laatimiseen. Usealta suunnittelijalta tuli toiveita opetusvideoiden tekemiseen tietokannan käytöstä. Vaikka tietokanta on



melko paljon käytetty ominaisuus, siihen toivotaan lisäperehdytystä. Tietokannan perusteella voidaan nopeasti luoda johdotustaulukoita esimerkiksi keskusvalmistajalle.

Kuvien generointi oli hieman vähemmän suunnittelijoiden käytössä kuin tietokannan hyödyntäminen. Generoinnin kokonaisvaltainen hyödyntäminen vaatii hyviä lähtötietoja projektissa ja kuvien samankaltaisuutta. Generoinnissa kerätään useasta Excel-taulukosta komponenttien positiot ja I/O-luettelon pisteet sekä sähkökeskukseen liittyvät tiedot samaan Excel-tiedostoon, joka yhdistetään CADSPohjakuvan kanssa. Lopputuloksena syntyy piirikaavio, joka sisältää kuvan lisäksi myös laitteiden positiot, I/O-pisteet sekä esimerkiksi moottoreiden virta- ja tehotiedot. Kuvien generoinnin käyttöaste on esitetty kuviossa 2.



KUVIO 2. Kuvien generoinnin käyttöaste

Kerättyjen tulosten perusteella huomataan, että vain puolet suunnittelijoista hyödyntää kuvien generointia suunnittelussa. Monelle suunnittelijalle, joka ei kuvien generointia hyödynnä oli työkalun toiminta hieman vierasta. Myös virtuaalikoneen käyttäminen generointiin koettiin ongelmaksi. Aktiiviset generointityökalun käyttäjät kertoivat tiedonhaun olevan melko hidasta suurissa projekteissa. Tähän ongelmaan pohditaan parannuksia myöhemmin generointityökalun kehityksen yhteydessä.

Haastatteluissa tiedusteltiin opetusvideoiden mahdollisia aiheita ja muutama agenda nousi toistuvasti haastateltavien suunnittelijoiden toimesta esille. Yleisin

toive oli tietokannan monipuoliseen hyödyntämiseen liittyvä opetusvideo. Tietokannan hyödyntäminen näyttelee suurta roolia esimerkiksi johdotus- ja komponenttelistojen tekemisessä, siihen tullaan varmasti kiinnittämään entistä enemmän huomiota tulevaisuudessa. Toinen aihe oli yrityksen yhtenevän suunnittelutavan opastaminen uusille suunnittelijoille. Tällä varmistetaan, että kaikkien suunnittelijoiden tuottamien kuvien laatu on asiakkaiden ja ennen kaikkea yrityksen vaatimalla tasolla. Lisäksi kattava opetusvideo vähentää kokeneempien suunnittelijoiden työtaakkaa uusien työntekijöiden opastamiseen, koska uudet työntekijät voivat katsoa opetusvideoita omalla työpisteellä ja tarvittaessa palata opetusvideoiden pariin.

Opetusvideoihin liittyvien kysymysten lisäksi kerättiin näkemyksiä CADS-suunnitteluohjelman parantamiseksi. Insta Automation Oy on jatkuvassa vuorovaikutuksessa Kyndata Oy:n suuntaan, jotta kehitystyötä ohjelman parissa saadaan vietyä eteenpäin. Näkemyksien vaihtaminen yritysten välillä hyödyttää molempia osapuolet, koska käyttäjien parannusehdotusten myötä suunnitteluohjelmasta saadaan kilpailukykyisempi muihin suunnitteluohjelmiin nähden. Joka päiväisessä käytössä on toivottua, että ohjelma toimii halutulla tavalla ja tarvittavia lisäominaisuuksia saadaan jatkossa käyttöön. Suunnittelijat toivoivat uusien komponenttien lisäämistä kirjastoon. Esimerkiksi usein käytettyjä ABB:n ja Schneiderin varokkeita toivottiin päivitettäväksi kirjastoon. Myös osa yleisistä riviliittimistä puuttuu symbolikirjastosta. Tietokannan kehittämistä toivottiin samankaltaiseksi kuin kilpailevalla ALMA Consulting Oy:llä. Kaikki tiedot pitäisi pystyä tuomaan Excel-ohjelman Import-toiminnolla kuten ALMA-suunnitteluohjelmassa. Myös CADS-suunnitteluohjelman jumittuminen ja kaatuilu koettiin turhauttavaksi. Kaatuilu ei ole pelkästään satunnaista vaan osa peräkkäisiä toimintasarjoja aiheuttaa systemaattisesti ohjelman kaatumisen. Ohjelman toiminnassa on myös software-tasolla jonkin verran parannettavaa.

## **4 KEHITYSTYÖ**

### **4.1 Generointikirjasto**

#### **4.1.1 Koneet**

#### **4.1.2 Venttiilit**

### **4.2 Generointityökalun kehitys**

#### **4.2.1 Toiminnallisuus**

#### **4.2.2 Ulkoasu**

#### **4.2.3 Instrumentit**

## 5 POHDINTA

Opinnäytetyön tekopaikka selvisi melko varhaisessa vaiheessa työskenneltyäni Insta Automation Oy:n suunnitteluosastolla kesän 2018. Yrityksestä luvattiin, että he järjestävät opinnäytetyön aiheen. Opinnäytetyö lähti käyntiin joulukuun puolessa välissä, jolloin aihe lopullisesti päätettiin. Työn alkuvaiheessa piirikaavioiden generointikirjaston kehittäminen kuulosti melko hankalalta asialta, koska kokemus piirikaavioiden suunnittelun parissa oli vähäistä. Aikaisempi ohjelmointikokemus C-kielellä antoi hyvät valmiudet opiskella VBA-koodin toimintaa. VBA-koodi oli aiheeseen tutustumisen jälkeen melko helppoa ymmärtää ohjelmointikielen visuaalisuuden ansiosta. Ohjelmarakenteet ja muuttujien määrittelyt tapahtuvat pitkälti samalla tavalla kuin C-kielessä. Piirikaavioiden lukeminen oli tuttua aiemmasta työskentelystä sähkö- ja automaatiokeskusten parissa. Myös käyttöönotkokokemus oli eduksi piirikaavioiden suunnittelussa, koska olin aiemmin työskennellyt kenttälaitteiden ja niihin liittyvien kytkentöjen kanssa.

Varsinainen opinnäytetyön tekeminen alkoi ongelmien kartoittamisella pystypalaverien muodossa, jossa tiedusteltiin suunnittelijoiden kokemia ongelmia ja kehitettäviä kohteita. Ongelmien kartoittamisella saatiin työn kannalta merkittävää etua, koska osattiin alusta lähtien keskittyä nimenomaan koettuihin ongelmiin ja kehitettäviin kohteisiin. Haastatteluiden tuloksena pystyttiin luomaan yhteenveto suurimmista haasteista generointityökalun käyttöön liittyen ja saatiin tietoa toivottuista aiheista opetusvideoihin liittyen. Haastatteluissa kerättyjä tietoja tullaan hyödyntämään opetusvideoiden luomisessa.

Generointikirjastoon perehtyminen lähti käyntiin tutustumalla verkkolevyllä oleviin piirikaavioihin. Piirikaaviot ladattiin verkkolevyllä omalle koneelle, jonka jälkeen jaottelu tapahtui piirikaavioiden osalta instrumentteihin, koneisiin ja venttiileihin. Otannassa käytettiin vuoden 2018 aikana suunniteltuja piirikaavioita. Tarkoituksena oli tutustua erilaisiin esitystapoihin ja ratkaisuihin, joita tulee päivittäisessä suunnittelussa vastaan. Aikaisemmin suunniteltujen piirikaavioiden avulla pystyttiin poimimaan piirikaavioiden hyvät ominaisuudet ja yhdistämään ne generointipohjassa.

Generointipohjien luominen aloitettiin koneista, joihin kuului erilaisia moottorikäyttöjä ja pumppuja. Generointipohjien attribuutteihin sijoitettiin generoinnissa käytettävät linkit ja niiden toimintaa testattiin asettamalla Excel-tiedostoon haluttuja tietoja, joilla voitiin helposti seurata tietojen siirtymistä Excel-tiedosta piirikaavioon generoinnin yhteydessä. Generointipohjissa pyrittiin huomioimaan mahdolliset optiot, jotka liittyvät kyseiseen piiriin. Aluksi suunnitteluohjelman piirrosteknisten työkalujen käytössä oli hieman opettelua. Generointipohjien täytyy olla siistejä ja tarkasti piirrettyjä, koska niiden pohjalta luodaan jatkossa valmiita piirikaavioita. Jos generointipohjassa on virheitä, virhe skaalautuu kaikkiin sen pohjalta tuotettuihin piirikaavioihin. Pelkästään siisti ulkoasu ei riitä, vaan kuvien pitää olla rasterilla piirrettyjä, jotta muokkaaminen olisi helppoa myöhemmässä vaiheessa.

Venttiileitä, peltejä ja instrumentteja koskevat generointipohjat luotiin samaan tapaan kuin koneiden tapauksessa. Generointilinkeissä käytetyt rivinumerot ja niiden selitykset vaihtelivat liitteessä 1 esitetyt taulukon mukaisesti. Aikaisempi kokemus venttiileiden ja instrumenttien käyttämisestä antoi hyvän pohjan ottaa suunnittelussa huomioon kenttälaitteen tärkeitä ominaisuuksia automaatiojärjestelmän osalta.

Generointikuvien lisäksi perehdyttiin generointityökalun toimintaan ja etsittiin parannusehdotuksia työkalun kehittämiseksi. Kehitys jakautui toiminnallisiin ja ulkoasullisiin parannusehdotuksiin. Toiminnallisella kehittämisellä löydettiin muutamia keinoja vähentää koodin suorittavia rivejä. Parannusten jälkeen suoritettavia koodirivejä olisi vähemmän ja koodirivejä suoritettaisiin vain tarvittava määrä. Ulkoasulliseen kehitykseen löydettiin muutamia menetelmiä parantaa haettavien tietojen konkreettisuutta esimerkiksi kuvien ja taulukoiden esimerkkien avulla. Tiedostopolkujen ja sarakkeiden nimien automatisoinnissa on ongelmana lähtötietojen erilaisuus.

Kehitystyön tekeminen oli varsin opettavainen kokemus. Työn aikana opittiin paljon piirikaavioiden suunnittelusta ja tehokkaista toimintatavoista siihen liittyen. Piirikaavioihin perehtymisen lisäksi työn aikana joutui tutustumaan erilaisten kenttälaitteiden toimintaan ja liityntöihin. Alussa asetettuihin tavoitteisiin päästiin mielestäni hyvin. Työn aikana luotiin monipuolinen generointikirjasto uudistuneilla

symboleilla sekä löydettiin muutamia kehitysehdotuksia ulkoasuun ja toiminnallisuuteen. Automaattinen päivämäärän hakeminen saatiin toimimaan generointitiedostossa Kyndata Oy:n asiantuntijoiden avustuksella. Tämä oli hyvä esimerkki siitä, että vuorovaikutus suunnitteluohjelman kehittäjän kanssa antaa molemmille osapuolille lisäarvoa ja auttaa kehittämään tuotteita parempaan suuntaan. Ongelman selvittäminen oli itselleni hieno kokemus asioiden hoitamisesta toisen yrityksen edustajan kanssa. Tulevassa CADS Electric 18.0.4 versiossa on tehty parannuksia generointiominaisuuden parantamiseksi. Asiasta tiedotti Toni Hirvonen, joka toimii asiantuntijana CADS-suunnitteluohjelman parissa.

Lopuksi tein pienimuotoisen ohjeen tiedostopolkujen ja tietojen hakemisesta generointityökaluun. Ohje voi toimia esimerkiksi tulevien opetusvideoiden tukena lähitulevaisuudessa. Ohjeisiin kerättiin olennaisimmat asiat eri tietojen hakemisesta esimerkkien avulla. Syvempään perehtymiseen voidaan tehdä kattavammat opastukset opetusvideoihin.

## LÄHTEET

ABB Oy. 2019. Luettu 15.1.2019.

[https://library.e.abb.com/public/92e7a194b8974b7397c6ba45034191fd/Lyhyt%20katalogi%20Moottorinohjaus%20ja%20sahkonjakelu%20FI14\\_03\\_1SBC100180C1801.pdf](https://library.e.abb.com/public/92e7a194b8974b7397c6ba45034191fd/Lyhyt%20katalogi%20Moottorinohjaus%20ja%20sahkonjakelu%20FI14_03_1SBC100180C1801.pdf)

Blomberg, E. & Lepoluoto, A. 2005. Audiokirja. Audiotekniikkaa ammattilaisille ja kehittyneille harrastajille. Helsinki.

[http://ari.lepoluo.to/audiokirja/Audiokirja\\_luku\\_7.pdf](http://ari.lepoluo.to/audiokirja/Audiokirja_luku_7.pdf)

Insta Group Oy. 2019. Luettu 2.1.2019.

<https://www.insta.fi/>

Kymdata Oy. 2019a. Luettu 19.1.2019

<http://www.cads.fi/>

Kymdata Oy. 2019b. CADS ohjeet. Ohjelman sisäinen Ohje. Luettu 19.1.2019

Kymdata Oy. 2007. CADS Planner Käyttäjän käsikirja. Kotka: Kymdata Oy

<http://fi.lastmanuals.com/kayttoohje/KYMDATA/CADS%20PLANNER%2012>

Leino, T. 2016. Excel-käyttäjän käsikirja. 2. painos. Helsinki: Oy Finn Lectura Ab

Merensalmi, J. 2007. Excel VBA yrityskäytössä. 1. painos. Jyväskylä: WSOYpro/Docendo-tuotteet.

Siemens Oy a. 2019. Luettu 31.1.2019.

[http://www.siemens.fi/fi/industry/teollisuuden\\_tuotteet\\_ja\\_ratkaisut/tuotesivut/pienjannitekojeet/kytkenta\\_suojaus\\_ja\\_ohjaus/termistorireleet\\_sirius.htm](http://www.siemens.fi/fi/industry/teollisuuden_tuotteet_ja_ratkaisut/tuotesivut/pienjannitekojeet/kytkenta_suojaus_ja_ohjaus/termistorireleet_sirius.htm)

Siemens Oy b. 2019. Luettu 31.1.2019.

[http://www.siemens.fi/fi/industry/teollisuuden\\_tuotteet\\_ja\\_ratkaisut/tuotesivut/pienjannitekojeet/kytkenta\\_suojaus\\_ja\\_ohjaus/simocode\\_alykas\\_moottorin\\_suojaus.htm](http://www.siemens.fi/fi/industry/teollisuuden_tuotteet_ja_ratkaisut/tuotesivut/pienjannitekojeet/kytkenta_suojaus_ja_ohjaus/simocode_alykas_moottorin_suojaus.htm)

## **LIITTEET**

Liite 1. Tietokantanumerot (Insta Automation verkkolevy)

Liite 2. Tiedonhakuohje