

AutoCad- ja CADS Planner- suunnitteluohjelmien automatisoinnin selvittäminen

Veli Suhonen

Opinnäytetyö

Koulutusala Tekniikan ja liikenteen ala	
Koulutusohjelma Kone- ja tuotantotekniikan koulutusohjelma	
Työn tekijä(t) Veli Suhonen	
Työn nimi AutoCad- ja CADS Planner-suunnitteluohjelmien automatisoinnin selvittäminen	
Päiväys 2.5.2012	Sivumäärä/Liitteet 43/1
Ohjaaja(t) Markku Halttunen, Seppo Ryyänen	
Toimeksiantaja/Yhteistyökumppani(t) Proxion Oy	
<p>Tämän opinnäytetyön tavoitteena alun perin oli automatisointimenetelmien selvittäminen AutoCad- ja CADS Planner-suunnitteluohjelmiin Proxion Oy:lle. Tarkoituksena oli selvittää menetelmiä joilla voitaisiin helpottaa ja nopeuttaa Proxion Oy:n turvalaite- ja sähköratasuunnittelijoiden työtä. Automatisointia haluttiin hyödyntää turvalaitteiden ja sähköratapylväiden automaattisessa sijoittamisessa kuvaan, sekä selvittää mahdollista kaksisuuntaista yhteyttä Excel-taulukkolaskentaohjelman sekä edellä mainittujen suunnitteluohjelmien välille.</p> <p>Opinnäytetyön edetessä itse automatisointia alettiin myös toteuttaa turvalaitesuunnittelijoiden käyttämään CADS Planner-suunnitteluohjelmaan. Automatisointi toteutettiin käyttämällä VBA-ohjelmointikieltä. VBA:lla kirjoitettiin ohjelma Exceliin, joka määrittää paikat symboleille sekä laskee niille oikeat kulmat. Lopuksi ohjelma kirjoittaa skripti-tiedoston, joka ajetaan CADS-ohjelman kautta. Skripti on käytännössä lista haluttuja käskyjä ja arvoja symboleille, jotka CADS tulkitsee ja tekee niiden pohjalta määrätyt toimenpiteet. Tässä tapauksessa asettaa turvalaite symbolit paikoilleen oikeassa kulmassa. Opinnäytetyön aikana selvisi, että CADS Planner- ja Excel-ohjelmien välille tehtävä molempiin suuntiin toimiva yhteys, olisi todella haastava tehtävä ja vaatisi mm. uuden ohjelmointikielen opettelemisen. Tämän takia sovittiin, että kyseisen ominaisuuden toteuttaminen jätettäisiin toistaiseksi tekemättä.</p> <p>AutoCad-ohjelman osalta itse automatisointia ei vielä tehty, koska opinnäytetyön aikana ei ollut täysin varmaa, minkälaiseen muotoon työkalu haluttaisiin. Blokkien automaattinen paikalleen asettaminen kuitenkin toteutettiin myös Autocadin osalta tämän työn aikana. AutoCadin osalta tehtiin esimerkkiohjelma, joka selvittää, miten AutoCad- ja Excel-ohjelmien välille muodostetaan yhteys, joka päivittää muutokset ohjelmasta toiseen. Tämä tarkoittaa käytännössä sitä, että kun AutoCad-kuvaan tehdään muutoksia esim. siirretään blokia, siirtää AutoCad uudet sijaintikoordinaatit Excel-taulukkaan. Samoin jos Excel-taulukosta muutetaan blokin koordinaatteja, voidaan ne päivittää AutoCad-kuvaan Excelistä käsin. Esimerkkiohjelmassa on mahdollista siirtää/muuttaa blokin sijaintia, kulmaa sekä dynaamisen blokin parametritietoja. Tulevaisuudessa kyseiset toiminnot siirretään sähköratasuunnittelijoiden käytössä olevaan Excel-tiedostoon.</p> <p>Suunnittelijoille automatisoinnista koituvat hyödyt ovat, työn helpottuminen, työn nopeutuminen sekä virheiden mahdollisuuden pieneneminen.</p>	
Avainsanat AutoCad, CADS, automaatio, blokki, symboli	

Field of Study Technology, Communication and Transport			
Degree Programme Degree Programme in Mechanical Engineering			
Author(s) Veli Suhonen			
Title of Thesis Research on the Automation Possibilities of AutoCad and CADS Planner			
Date	2.5.2012	Pages/Appendices	43/1
Supervisor(s) Markku Halttunen, Seppo Ryyänen			
Client Organisation/Partners Proxion Oy			
<p>The aim of this study was originally to find out automation methods for AutoCad and CADS Planner planning programs for the use of Proxion Ltd's personnel. The goal was to find out methods that could make Proxion Ltd's planners work easier and faster for the planning of electrical railway- and railway safety systems. The goal was to get the blocks and symbols which these planning programs are using on their own places with the automated program. Another goal was to find a way to make these planning programs and Excel spreadsheet program to work bidirectionally</p> <p>Later on the thesis the automation itself was also applied to the CADS planning tool. The automation was carried out by using VBA programming language. Using VBA we wrote a program to the Excel spreadsheet program which defines the coordinates for the symbols used in railway safety systems programming. It also calculates the right angles for these symbols. In the end it will create a script file which can be used through the CADS Planner program. Script file is basically a list of orders and values for the symbols which the CADS Planner can read and put in to use. In this case it will put the symbols and blocks on to the right place and in right angle. During this final thesis it was found out that the bidirectional connection with CADS Planner and Excel would be very challenging to execute and would demand learning a new programming language. Therefore it was agreed that it would be left out of this thesis.</p> <p>What comes to AutoCad planning program the automation was not completed because during the thesis it wasn't clear which kind of tool would be the best choice (Except the automated adding of the blocks. That was also executed in AutoCad during this final thesis) So what comes to the connection between AutoCad and Excel it was solved and an example program was created. This example program shows how this kind of connection can be done. The example program shows how Excel gets information from AutoCad drawing and how by changing this information in Excel it will update the changes in the AutoCad drawing. In practice this means that when AutoCad block is moved, the new coordinates can be sent to Excel and when these coordinates change it moves the block in the drawing. In this example program it is possible to move a block or change its angle. It's also possible to change the dynamic blocks parameters. In future these actions will be moved to Excel which is used by the planners in electrical railway planning.</p> <p>Automation will make the planner's work easier, faster and also it will reduce the possibility of errors.</p>			
Keywords AutoCad, CADS, automation, block, symbol			

ALKUSANAT

Suuret kiitokset Proxion Oy:n sähköratasuunnittelija Jyri Taivaiselle avusta ja ideoinnista tämän opinnäytetyön aikana. Kiitokset Proxion Oy:lle joustavuudesta ja mahdollisuudesta toteuttaa tämä opinnäytetyö. Kiitokset myös opinnäytetyön ohjaajille Markku Halttuselle ja Seppo Ryytäälle, avusta ja tuesta tämän opinnäytetyön aikana. And last but not least thank you Chuck Hardin for the help and all the information you gave me during this final thesis.

31.5.2012 Tihusniemessä

Veli Suhonen

SISÄLTÖ

1	JOHDANTO	7
1.1	Yrityksen esittely	8
1.2	Taustaa	9
1.3	Tavoite	10
2	AUTOMATISOINTIMENETELMÄT	12
2.1	Työkalut.....	17
3	AUTOMATISOINNIN TOTEUTTAMINEN	18
3.1	Ohjelmointikielen valinta	18
3.2	VBA ja tulevaisuus	20
4	TOTEUTUS.....	21
4.1	Selvitystyö	21
4.2	Skriptien selvittäminen	22
4.3	Skripti-tiedoston rakenne AutoCad- ja CADS-ohjelmissa	22
4.4	Esimerkkiohjelman tekeminen.....	24
4.5	Esivalmistelut	31
5	KUINKA TAVOITTEET SAAVUTETTIIN?	34
6	KEHITYSEHDOTUKSIA	36
6.1	CADS-ohjelman kuvien eri reviisioissa tapahtuvien muutosten automaattinen merkitseminen	37
6.2	CADS-ohjelmasta ajettavien useiden eri luetteloiden ajaminen yhteen Excel-tiedostoon	37
6.3	Raidenumeroitien automaattinen paikalleen asettaminen ja oikeiden numeroiden hakemisen automaattisesti	38
6.4	AutoCadin usean tulostuskuvan automatisointi	38
7	MITÄ ITSE OPIN?	39
8	YHTEENVETO	40
	LÄHTEET	42

LIITTEET

Liite 1 CADS Planner käyttöohje

Käytetyt lyhenteet ja termit

AutoCad	CAD-pohjainen (Computer Aided Design) 2D ja 3D-suunnitteluohjelma.
CADS	Suomalaisen Kymdata Oy kehittämä 2D- ja 3D-pohjainen suunnitteluohjelma.
Blokki/Symboli	Suunnitteluohjelmiin tallennettuja valmiita kuvioita.
RATO	Ratatekniset ohjeet
SSR	Sähköistyksen kiinteidenlaitteiden suunnittelu ja rakentaminen ohjekoelma
Turvalaite	Laite, joka turvaa junaradan liikennöintiä
Sähköratapylväs	Pylväs, joka kannattelee sähköradan johtimia
ATU	Avoimen tilan ulottuma. Tila joka rautatiellä on varattu junalle.
VBA	Visual Basic for Application on Microsoftin kehittämä ohjelmointikieli.
AutoLISP	AutoCadin oma ohjelmointikieli.
ObjectARX	Ohjelmointirajapinta AutoCad-ohjelmaan.
Acsii	American Standard Code for Information Interchange. 7-bittinen/128 merkkipaikan laajuinen tietokoneiden merkkistö.
C++	Ohjelmistokehityksessä käytettävä ohjelmointikieli.
C#	C sharp on Microsoftin .NET-konseptia varten kehittämä ohjelmointikieli.
VB.NET	Olioperustainen ohjelmointikieli, joka on VB-kielen nykyinen versio.
Siksak	Ajolanka poikkeaa puolelta toiselle radan keskilinjaan nähden.
Skripti	Komentojoono
Osnap	AutoCad-ohjelman toiminto jolla voidaan määrittää AutoCad hakemaan automaattisesti aloituspiste esim. jo olemassa olevasta kuviosta, tietystä kohtaa.
Dynaaminen blokki	Ominaisuuksiltaan tavallista blokkia monipuolisempi blokki. Dynaamiseen blokkiin on mahdollista lisätä mm. eri parametreja, joilla voidaan asettaa blokki esim. tietylle etäisyydelle jostain pisteestä. Pelkkää parametria muuttamalla blokin etäisyys sitten vaihtelee.
Atribuuttitieto	Blokkiin/symboliin lisättävä tieto, joka voidaan asettaa näkyviin blokin/symbolin viereen kuvassa.

1. JOHDANTO

Tämän opinnäytetyön alkuperäisenä aiheena oli AutoCad- ja CADs-suunnitteluohjelmien automatisointimenetelmien selvittäminen pieksämäkeläiselle Proxion Oy:lle. Opinnäytetyö oli rajattu automatisoinnin selvittämiseen, koska varsinainen automatisointi vaatii ohjelmointiosaamista. Kuitenkin työn edetessä itse automatisointia alettiin myös toteuttaa lopputyön aikana.

Tällä hetkellä suunnittelijat joutuvat lisäämään jokaisen blokin manuaalisesti paikoilleen yksi kerrallaan. Proxion Oy:n suunnittelijat tekevät mm. turvalaite- sekä sähköratasuunnittelua junaraideympäristöön. Turvalaite on laite, jolla turvataan junaradan liikennöintiä. Sähköratapylvästas on pylvästas, joka kannattelee sähköjohtimia, joista juna ottaa virran. Suunnittelijat tekevät blokkeja, määrittävät turvalaitteiden ja sähköratapylvästasien paikat sekä lisäävät blokkeja paikoilleen. Opinnäytetyön tavoitteena on selvittää keinoja tämän prosessin helpottamiseksi ja nopeuttamiseksi. Lisäksi tulisi selvittää, onko mahdollista saada Excel-taulukkolaskentaohjelma kommunikoimaan suunnitteluohjelmien kanssa ns. molempiin suuntiin ja jos on, miten tämä tapahtuu. Kommunikoinnilla molempiin suuntiin tarkoitetaan tässä yhteydessä sitä, että suunnitteluohjelmiin tai Exceliin tehdyt muutokset voidaan päivittää myös toiseen ohjelmaan ilman manuaalista lisäämistä.

Automatisoinnilla pyritään helpottamaan ja nopeuttamaan suunnittelijoiden työtä. Automatisointi myös vähentää virheiden mahdollisuutta. Automatisointia kannattaa toteuttaa samojen kuvioiden ja työvaiheiden toistuessa useasti. Yleisesti ottaen automatisointiin tulisi myös olla jokin sääntö, jonka mukaan kyseinen toiminto tapahtuu. Satunnaisesti tapahtuvaan toimintoon automatisointia on vaikea soveltaa. Suunnitteluohjelmissa puhutaan ns. blokeista sekä symboleista, jotka ovat suunnitteluohjelmiin tehtyjä ja tallennettuja valmiita kuvioita. Näitä blokkeja voidaan sitten asettaa haluttuihin kohtiin, ettei niitä tarvitse piirtää aina uudestaan. Periaatteessa blokkien tekeminen ja käyttäminen on jo itsessään jonkin asteista automatisointia. AutoCadissa puhutaan blokeista ja CADs Planner-ohjelmassa käytetään symboli-nimitystä.

CADs Planner-suunnitteluohjelman automatisointi saatiin tasolle, jolla sitä voidaan testata käytännössä. Luonnollisesti ohjelma vaatii vielä parannuksia ja kehitystä uusien tarpeiden ja käytännön tuoman kokemuksen kautta. AutoCadin automatisointi saatiin toteutettua blokkien sijoittamisen osalta, mutta AutoCadin ja Excelin välistä yhteyttä ei vielä otettu käyttöön, sillä täyttä varmuutta siitä millaiseksi työkalu lopulta halutaan, ei vielä opinnäytetyön aikana ollut.

1.1 Yrityksen esittely

Proxion Oy on vuonna 2005 perustettu asiantuntijayritys, joka tarjoaa asiantuntijapalveluiden sekä suunnittelu- ja rakennuttamisprojektien lisäksi henkilö-, liikenne- ja sähköturvallisuuskokonaisuuksia asiakkaiden kohteisiin. Proxionille tärkeitä tekijöitä ovat laatu, turvallisuus, luottamuksellisuus sekä vahva asiakaslähtöisyys. Yrityksessä työskentelee 55 henkilöä ja pääkonttori sijaitsee Piekämäellä. Proxionilla on myös toimipisteet Helsingissä, Kouvolassa ja Oulussa, sekä projektitoimistot Kokkolassa ja Seinäjoella. Proxion-konserni on jaettu tytäryhtiöihin seuraavasti.

Proxion hyvinvointipalvelut

Asiantuntija- ja koulutuspalveluita hyvinvointi- ja terveyspalveluiden kehittämiseen tarjoavan Proxion Care:n menestyksen avaintekijät ovat yhteiskunnan toimintamekanismien ja taustojen sekä julkisten organisaatioiden toimintatapojen ymmärtäminen. Proxion Care yhdistää soisiaali- ja terveydenhuollon asiantuntijaosaamista insinööriosaamiseen, joten se voi tarjota uusia innovatiivisia näkökulmia mm. kotihoidon prosessien ja toimintamallien kehittämistyöhön.

Proxion koulutuspalvelut

Proxion koulutus- ja asiantuntijapalvelut on urakoitsijasta riippumaton koulutuskeskus, joka tarjoaa monipuolisia turvallisuus- ja pätevyyskoulutuksia rakentamiseen. Kouluttajina toimivat rata-, liikenne-, työ- sekä sähköturvallisuuden ammattilaiset, jotka räätälöivät asiakaskohteisiin sopivat koulutuskokonaisuudet perustuen käytännön työelämässä kohdattaviin ongelmatilanteisiin.

Proxion Train Oy

Proxion Train Oy on yksityinen suomalainen rautatieyhtiö, jonka on tarkoitus aloittaa liikennöintipalveluiden tarjoaminen lähivuosina. Proxion Train Oy:n tavoitteena on luoda kaupallisesti vahva ja itsenäinen toimija Suomen rautatieliikenteeseen. Tehokas organisaatio ja korkea kaluston käyttöaste takaavat liikennöinnin turvallisuuden ja luotettavuuden, jotka ovat Proxion Trainin tärkeitä toimintakriteereitä. Proxion Train keskittyy ensisijaisesti suurteollisuuden tarjoamiin tavarankuljetuksiin.

Proxion ohjelmistopalvelut

Kokonaisvaltaisia ohjelmistopalveluita rautatieympäristössä sekä IT-alalla toimiville asiakkaille tarjoaa Proxion Solutions. Palvelu kattaa ohjelmistojen ja tietojärjestelmien kehityksen määrittelystä ja räätälöinnistä käyttöönottoon ja testaukseen. Proxion Solutionsin erikoisosaamista teknologia-alueittain ovat avoimen lähdekoodin tuntemus, Linux- käyttöjärjestelmät, web-ohjelmistot ja tietokannat (MySQL, PHP, Ajax, Python, Django) sekä Java, Eclipse/GEF-alusta.

Proxion rakennuttamispalvelut

Proxion Power on sähkö- ja turvalaiterakennuttamisen sekä sähköalan asiantuntijatehtävien ammattilainen, jonka ydiosaamista ovat projektinjohto-, suunnitteluttamis-, valvonta- ja asiantuntijapalvelut liikenne- ja teollisuusympäristöjen sähkötekniisiin järjestelmiin.

Kokonaisvaltaisia projektien johtamispalveluita infra-, teollisuus-, ja toimitilainvestoinneissa sekä julkisessa rakentamisessa tarjoava Proxion Infra omaa pitkän kokemuksen vaativista kustannus- ja aikataulusidonnaisten projektien johtamisesta. Tämä takaa tilaajan organisaation rinnalle osaavan ja ammattitaitoisen projektin johdon. Toimeksiannot toteutetaan tiiviissä yhteistyössä tilaajan, urakoitsijoiden ja suunnittelijoiden kanssa.

Proxion suunnittelupalvelut

Proxion Plan tarjoaa turvalaite- ja sähkösuunnittelupalveluja sekä rautateiden sähköistysuunnittelua julkisten ja yksityisen sektorin hankkeisiin. Ammattitaitoinen ja innovatiivinen henkilökunta sekä kehittyneet työkalut toimivat Proxion Planin lähtökohtana. Korkea ammattiosaaminen, asiakaslähtöisyys sekä kokemus aikataulusidonnaisista suunnitteluprojekteista takaavat toimeksiantojen laadukkaan lopputuloksen.

1.2 Taustaa

Proxion Oy:n suunnittelijat tekevät suunnittelutyötä rautatieympäristöön. Suunnittelijat tekevät sähkörata- sekä turvalaitesuunnittelua. Sähköratasuunnittelussa käytetään AutoCad- ja turvalaitesuunnittelussa CADS Planner-suunnitteluohjelmaa. Sijoittaessaan sähköratapylväitä ja turvalaitteita suunnittelijat joutuvat ottamaan huomioon monia määräyksiä ja asetuksia, joiden mukaan turvalaitteet ja sähköratapylväät lopulta sijoitetaan. Sekä turvalaite- että sähköratasuunnittelussa suunnittelijat saavat ns. radan geometriakuvaan ja tiedoston. Geometriakuvaan suunnittelijat lisäävät

sähköratapylväät sekä turvalaitteet. Geometriatiedostossa ilmenee radan geometria, eli miten rata kulkee, millaisia mutkia se tekee yms. Näiden geometrioiden pohjalta suunnittelijat voivat lähteä määrittämään sähköratapylväiden ja turvalaitteiden sijaintia kuvassa.

Turvalaitteiden sijainnille kriteerit määrittää ratatekniset ohjeet (RATO). RATO on liikenneviraston antama ohjeistus, joka käsittää perustiedot radan ja ratalaitteiden suunnittelua, rakentamista, tarkastusta ja kunnossapitoa varten. Suunnittelijat joutuvat määrittämään jokaiselle turvalaitteelle sijainnin manuaalisesti RATO:n määräysten pohjalta.

Sähköratapylväitä sijoitettaessa noudatetaan myös RATO:n sekä SSR:n ohjeistuksia. Myös sähköratapuolelle suunnittelijat joutuvat määrittämään sähköratapylväiden sijainnin ohjeistuksien perusteella. Suunnittelijoilla on käytössä Excel-taulukkoon tehty ohjelma, joka määrittää jokaiselle sähköratapylväälle sijaintikoordinaatit saatujen geometriatietojen pohjalta. Määritettäessä sähköratapylväiden sijaintia täytyy ottaa huomioon monta eri tekijää. Esimerkkinä, jos juna ajaa kaarteeseen niin se kallistuu hieman ja telien välinen osa ”oikaisee”, jolloin pylväs on oltava ATU:n levityksen mukaisesti kauempana raiteesta. Tämä taas vaikuttaa suoraan sähköratapylvään sijaintiin kyseisessä kohdassa. Junan katolla oleva virroitin ottaa virtaa ajojohtimesta, joka riippuu sähköratapylväiden päässä olevista kannatinrakenteista. Ajojohdin on asetettu tekemään ns. siksak-liikettä junaraiteeseen nähden. Tämä sen vuoksi, että junan virroitin kuluisi mahdollisimman tasaisesti joka kohdasta. Junan virroitimen liikkumavara muodostuu useista tekijöistä. Siihen vaikuttavat mm. raiteessa olevat mahdolliset virheet, vetokaluston jousituksesta aiheutuva sivuttainen liike, raiteen kallistuspoikkeama, sekä jo mainittu ajojohtimen siksak.

1.3 Tavoite

Tavoitteena tässä työssä oli selvittää automatisointimahdollisuuksia CADS Planner ja AutoCad-suunnitteluohjelmissa, siten että niistä olisi mahdollisimman paljon hyötyä Proxion Oy:n suunnittelijoiden työssä. Tarkoituksena oli hyödyntää automatisointia sähköratapylväiden sekä turvalaitteiden paikalleen sijoittamisessa. Lisäksi työhön kuului selvittää, onko suunnitteluohjelmia ja Excel taulukkolaskentaohjelmaa mahdollista saada keskustelemaan keskenään ns. molempiin suuntiin. Esimerkkinä, jos AutoCad-kuvassa jotakin sähköratapylvään blokkia siirretään, osaisi Excel päivittää uudet sijaintikoordinaatit taulukkoon ja päinvastoin.

Vaikka opinnäytetyö oli alun perin rajattu käsittämään vain automatisoinnin selvitys, alettiin opinnäytetyön aikana jo toteuttaa automatisointia turvalaittepuolelle. Tavoitteeksi otettiin, että menetelmää päästäisiin ainakin kokeilemaan opinnäytetyön aikana. Kehitys- ja parannustar-

peet ilmenevät käytännön kokemusten kautta. Opinnäytetyön aikana ohjelmaan saataisiin kuitenkin toimiva runko, jota voitaisiin sitten lähteä kehittämään ja parantamaan tarpeen mukaan. Kun opinnäytetyön aikana alettiin tehdä itse automatisointia, sovittiin, että ohjelman käytöstä ja siihen liittyvistä toimintatavoista kirjoitettaisiin myös ohje, jonka perusteella suunnittelijat osaisivat toimia ja käyttää ohjelmaa oikein. Ohjeesta tulisi tehdä mahdollisimman yksinkertainen ja se tuli pyrkiä tekemään niin, ettei väärinkäsityksiä pääsisi syntymään. Sähköratasuunnittelun puolella automatisointi loppujen lopuksi rajoittui sähköratapylväiden sijoittamisen automatisointiin, sekä kaksisuuntaisen yhteyden havainnollistavan esimerkkiohjelman tekemiseen. Tämä sen takia ettei opinnäytetyön aikana vielä ollut täysin varmaa, millaiseen muotoon lopullinen työkalu haluttaisiin.

Ensimmäisenä selvitettiin onko mahdollista saada AutoCad ja Excel siirtämään tietoa ns. molempiin suuntiin, ja jos on niin miten tämä toteutettaisiin käytännössä. Tämä kyseinen selvitystyö ja mahdollinen toteutus oli aluksi tavoitteena myös CADS Planner-ohjelman osalta, mutta selvitystyön aikana ilmeni muutamia seikkoja, jotka toistaiseksi asettivat tämän tavoitteen tulevaisuuteen. Keskustelussa opinnäytetyöstä vastaavan suunnittelijan kanssa tultiin siihen lopputulokseen, että CADS-ohjelman osalta Excelin kanssa tapahtuva kaksisuuntainen yhteys jätettäisiin toistaiseksi toteuttamatta. CADS Planner-ohjelman osalta päädyttiin lopputulokseen, jossa turvalaitesymbolit asetetaan paikoilleen skripti-tiedoston avulla. CADS Planner- ja Excel-ohjelmien välisestä yhteydestä myöhemmin lisää.

AutoCadin osalta yhteyden muodostamiselle ei ilmennyt yhtä suuria vaikeuksia. Esimerkkiohjelman tarkoituksena olisi siirtää blokkien tietoja AutoCad-kuvasta Excel-taulukkoon ja päinvastoin. AutoCad-kuvasta Exceliin haluttiin siirtää blokin nimi, sijaintikoordinaatit, dynaamisen blokin parametritieto sekä blokin kulma. Kun nämä arvot saadaan Excel-taulukkoon, niitä voidaan muuttaa ja päivittää muutokset AutoCad-kuvaan. Näin voidaan nopeasti asettaa blokeille halutut arvot ja sijainnit, joka AutoCadissa vaatisi useita työvaiheita. Parametritietojen muuttamista Excelistä voidaan hyödyntää esimerkiksi, jos halutaan muuttaa dynaamiseen blokkiin asetettuja etäisyysarvoja. Käytännön esimerkkinä voisi mainita sähköratasuunnittelussa olevien sähköratapylväiden orsien pituuden muuttamisen. Yleisesti katsottuna arvojen ja sijaintitietojen muuttaminen Excelistä nopeuttaa työskentelyä ja varsinkin, jos joudutaan muuttamaan useiden blokkien tietoja kerralla tai halutaan muuttaa kaikkien blokkien tiedot samoiksi. Myös sähköratapylväitä kuvaavien blokkien sijoittaminen tulisi automatisoida.

Myös sähköratapylväiden ja turvalaitteiden symbolien sijoittaminen skriptien avulla nopeuttaisi suunnittelijoiden työtä. Manuaalisesti tehtäessä symboleille määritetään paikat ja asetetaan ne sitten yksi kerrallaan paikoilleen. Tämä on suhteellisen hidasta ja työlästä, joten jos symboleiden ja

blokkien sijainnit voitaisiin määrittää ja sitten asettaa ne kerralla paikoilleen, työ nopeutuisi huomattavasti. Tämä käytännössä jättäisi yhden työvaiheen kokonaan pois. Tämä oli tavoitteena ja tätä lähdettiin myös toteuttamaan opinnäytetyön aikana.

Turvalaitesuunnittelussa ei suoraan voida laskea symbolien paikkoja vaan ne täytyy määrittää RATO:n mukaan. Tällä hetkellä turvalaitesuunnittelijat piirtävät kaksi kuvaa. Puhutaan ns. 1:10 000- ja 1:1000-kuvista. 1:10 000-kuvassa rata kuvataan suorana viivana, johon turvalaitteille määritetään sijainnit ja asetetaan ne paikoilleen. 1:1000-kuvassa rata taas kuvataan oikeassa muodossaan. 1:1000-kuva tehdään suunnittelijoiden saamaan geometriakuvaan. Automatisointia skriptin avulla voidaan hyödyntää vaiheessa jossa 1:10 000-kuvan pohjalta aletaan tekemään 1:1000-kuvaa. Automatisoinnin avulla 1:10 000-kuvasta voidaan poimia symbolien sijainnit ja siirtää ne sitten 1:1000-kuvaan, jolloin periaatteessa yksi työvaihe jää kokonaan pois.

2. Automatisointimenetelmät

Tähän kappaleeseen on muutamalla sanalla käyty läpi joitakin automatisointimenetelmiä, joista ajateltiin olevan hyötyä tämän opinnäytetyön aikana. Ohjelmointikielien osalta lisättiin myös yksinkertaiset esimerkkiohjelmat, jotta nähdään hieman kielen rakennetta.

Blokki/Symboli

Blokkit ja Symbolit ovat suunnitteluohjelmissa olevia valmiita piirrettyjä kuvioita, joita voidaan lisätä kuviin. AutoCadissa käytetään nimitystä blokki ja CADS Planner-ohjelmassa puhutaan symboleista. Blokkeja ja symboleita kannattaa käyttää usein toistuvissa kuvioissa. Esimerkkinä sähkörata-pylväät ja turvalaitteet kuten opastimet kannattaa tallettaa ohjelman muistiin, koska niitä joudutaan käyttämään jatkuvasti. Näin ollen niitä ei tarvitse piirtää joka kerta uudestaan.

Skripti

Skripti eli komentojono on käytännössä lista peräkkäisiä komentoja, joita ohjelma lähtee toteuttamaan. Skripti soveltuu pienten ja yksinkertaisten toimintojen automatisointiin. Skriptit kirjoitetaan ASCII-muotoiseen tiedostoon (esim. notepad). Komennot tulee kirjoittaa oikeaan muotoon ja oikeille paikoille, jotta ohjelma ymmärtää ne. Skriptin ongelma onkin juuri oikeinkirjoitus, sillä yksikin välilyönti väärässä paikassa saattaa estää koko skriptin toiminnan. AutoCadin skriptit tulee tallettaa src-päätteellä ja CADSin mac-päätteellä. Skripti on kuitenkin yksinkertainen ja helppo tapa pienten toimintojen automatisointiin. Se on myös hyvä vaihtoehto, jos tarvitsee tehdä useasti toistuvaa ko-

mentoa. Skripti-tiedostoa voidaan käyttää sähköratapylväiden ja turvalaitteiden blokkien automaattisessa paikoilleen sijoittamisessa.

Visual Basic For Application eli VBA

Visual Basic on Microsoftin kehittämä ohjelmointikieli, joka soveltuu niin pienten kuin laajojenkin ohjelmien tekemiseen. Wikipediassa VBA määritellään seuraavasti: ”Visual Basic for Applications (VBA) on isäntäohjelman sisällä toimiva ohjelmointikieli. Se on käytössä mm. Microsoft Office - tuotteissa. Käyttötarkoitukseltaan VBA on makrokieli, jolla ohjataan ja täydennetään isäntäohjelman toimintoja.” AutoCad-ohjelmasta löytyy itsestään VBA-sovellus, CADS Planner-ohjelmassa sitä ei ole.

Alla yksinkertainen VBA:lla toteutettu esimerkkiohjelma, joka hakee Excel-taulukosta koordinaatit joihin AutoCad sitten piirtää pisteet. Ohjelma löytyy osoitteesta:

<http://howtoautocad.com/excel-and-autocad-%E2%80%93-a-match-made-in-heaven-again/>

```
Sub Main()
    Dim ACAD As AcadApplication 'Create ACAD variable of type AcadApplication
    On Error Resume Next 'This tells VBA to ignore errors
    Set ACAD = GetObject(, "AutoCAD.Application") 'Get a running instance of the
class AutoCAD.Application
    On Error GoTo 0 'This tells VBA to go back to NOT ignoring errors
    If ACAD Is Nothing Then 'Check to see if the above worked
        Set ACAD = New AcadApplication 'Set the ACAD variable to equal a new
instance of AutoCAD
        ACAD.Visible = True 'Once loaded, set AutoCAD to be visible
    End If
    ACAD.ActiveDocument.Utility.Prompt "Hello from Excel!" 'Print a message to
the AutoCAD command line
    Dim Coords(2) As Double 'This is an array of double precision floating point
numbers
    ' The array goes from 0 - 2, which will contain our coordinates X, Y and Z
    Dim n As Integer 'Create the variable n as the type Integer
    For n = 1 To 10 'Loop this code, incrementing the value of n from 1 to 10
        Coords(0) = Sheet1.Cells(n, 1) 'Put the Column 1 value into the Coords
array
        Coords(1) = Sheet1.Cells(n, 2) 'Put the Column 2 value into the Coords
array
        ACAD.ActiveDocument.ModelSpace.AddPoint Coords 'Add a point in AutoCAD
at this location
    Next
End Sub
```

AutoLISP

AutoLISP on AutoCadin oma ohjelmointikieli. AutoLISP-ohjelmointikielen avulla pystytään ohjelmoimaan monipuolisesti AutoCadia. AutoLISP-ohjelmointikieli on suhteellisen yksinkertaista ja lyhyttä, ainakin verrattaessa sitä ObjectArx:n käyttämään C++-kieleen.

Alla esimerkki AutoLISP-ohjelmointikielellä tehdystä ohjelmasta. Esimerkkiohjelman avulla voidaan merkata keskipiste ympyrään, ellipsiin tai kaarelle pelkästään hiirellä näyttämällä. Esimerkkiohjelma löytyy osoitteesta

<http://www.lee-mac.com/centreline.html>

```
;;-----=={ Centreline }==-----;;
;;
;; Creates a centreline for Arcs, Circles & Ellipses with
;; a user-defined length and angle.
;;-----
;; Author: Lee Mac, Copyright © 2010 - www.lee-mac.com
;;-----

(defun c:cl nil (c:Centreline))

(defun c:Centreline ( / e a d p1 p2 ) (vl-load-com)
  ;; Lee Mac 2010

  (if (and
    (setq e
      (LM>Selectif
        (lambda ( e )
          (member (cdr (assoc 0 (entget e))) '("CIRCLE" "ARC" "ELLIPSE")))
        )
      entsel "\nSelect Arc, Circle or Ellipse: "
    )
    )
    (setq p2
      (getpoint (setq p1 (trans (cdr (assoc 10 (entget e))) e 1))
        "\nSpecify Length of Centreline: "
      )
    )
  )
  (progn (setq a (angle p1 p2) d (distance p1 p2))
    (mapcar
      (function
        (lambda ( a d )
          (entmakex
            (list
              (cons 0 "LINE")
              (cons 10 (trans (polar p1 a d) 1 0))
              (cons 11 (trans (polar p1 a (- d)) 1 0))
            )
          )
        )
    )
  )
)
```


Alla esimerkki C++-kielellä kirjoitetusta yksinkertaisesta ohjelmasta, joka laskee annetut numerot yhteen ja laskee niistä sitten keskiarvon. Esimerkiohjelma löytyy osoitteesta:

<http://www.codeobsessed.com/c++.html>

```
1 // =====
2 // This program adds two numbers and then averages them.
3 // =====
4 #include <iostream>
5 using namespace std;
6
7 int main ()
8 {
9     double num1 = 0;
10    double num2 = 0;
11    double total = 0;
12    double avg = 0;
13
14    cout << "Please enter a number: ";
15    cin >> num1;
16
17    cout << "Please enter a second number: ";
18    cin >> num2;
19
20    total = num1 + num2;
21    cout << "\nThe total of the two numbers is: " << total << "\n";
22
23    avg = (total / 2);
24
25    cout << "The average of the two numbers is: " << avg << "\n\n";
26
27    return 0;
28 }
29 /* ===== output =====
30 Please enter a number: 88.6
31 Please enter a second number: 75.3
32
33 The total of the two numbers is: 163.9
34 The average of the two numbers is: 81.95
35
36 Press any key to continue . . .
37 */// =====
```


K-ohjelmointikieli

Seuraava K-ohjelmointikielen esittely on lainattu Kymdatán K-ohjelmointikielioppaasta. Sitä ei katsottu tarpeelliseksi muuttaa, sillä työn tekijän puutteellisen ohjelmointiosaamisen takia asia saatettaisiin ymmärtää väärin.

”K-ohjelmointikieli on merkeistä, numeroista ja nimistä koostuva CADS Planner-suunnitteluohjelman oma ohjelmointikieli. K on rakenteinen koneenläheisen ja tehtävänläheisen kielen välimaille sijoittuva ohjelmointikieli. Se ei ole koneenläheinen, koska K-ohjelmassa ei ole yhtään funktiota, jolla päästäisiin suoraan koneen muistiin käsiksi eikä se käsittele dataa akkujen eikä rekistereiden avulla. Se ei ole varsinaisesti tehtävänläheinen, koska se ei ole erityisen älykäs kieli; äly täytyy jokaisen ohjelmoijan itse rakentaa. K:sta tosin löytyy piirteitä, jotka siirtävät sitä tehtävänläheisempään suuntaan, kuten suora liityntä CADSiin, funktio pisteiden lukemista varten, geometriset funktiot, kuvatietokantaliityntä jne. mutta pääasiassa K tarjoaa vain työkalut, joilla sovelluksia voidaan rakentaa.

Ulkoasultaan K muistuttaa ehkä eniten jotakin pseudokieltä, koska siihen on poimittu piirteitä monista eri kielistä tarkoituksella, helppokäyttöisyyden nimissä.

K:n syntaksi on helppo ymmärtää, mikäli käyttäjä tuntee jonkin seuraavista kielistä:

BASIC

FORTRAN77

PASCAL

C

PL/M”

2.1 Työkalut

AutoCad

AutoCad on Autodesk Inc.-yhtiön kehittämä 2D- ja 3D-pohjainen tietokoneavusteinen suunnitteluohjelma. AutoCad on vektorigrafiikkaan perustuva ohjelma. Vektorigrafiikka taas on tietokonegrafiikka, joka perustuu koordinaatistoon sidottuihin objekteihin, suoriin viivoihin, ympyröihin, kaariin, teksteihin jne. AutoCad on yleissuunnitteluohjelma, joka on kuitenkin laajennettavissa useilla sovelluslakohtaisilla laajennuksilla. Ohjelmistolla on myös useita rajapintoja, joilla voidaan luoda uusia

toimintoja eri ohjelmointikieliä käyttäen. Tämä onkin tehnyt AutoCadista suosittua ohjelmistoalustan.

CADS Planner

CADS Planner on suomalaisen Kyndata Oy:n kehittämä 2D- ja 3D-pohjainen suunnitteluohjelma. Ohjelma muistuttaa käytettävyydeltään ja ulkoasultaan hyvin paljon AutoCadia. CADS Planner on perus CAD-ohjelma (computer aided design), johon perustuvat toimialakohtaiset suunnitteluohjelmat, kuten sähkö- ja automaatio suunnittelussa käytetty CADS Planner Electric ja LVIA-suunnittelussa käytetty CADS Planner Hepac. CADS Planner-ohjelmasta löytyy oma ohjelmointikielensä K.

Excel-tilukkolaskentaohjelma

Microsoft Excel on soluihin perustuva tilukkolaskentaohjelma, joka kuuluu Microsoft Office-tuoteperheeseen. Excel sisältää laskemiseen, graafiseen esittämiseen ja tilukointiin soveltuvia työkaluja. Excelissä on myös oma sisäinen VBA-ohjelmointimoduulinsa, jolla sitä voidaan ohjelmoida. Excel on hyvin suosittu ja laajasti käytetty tilukkolaskentaohjelma. Excelin ulkoasu on tilukko, joka koostuu soluista. Solurivit on nimetty numeroilla ja sarakkeet kirjaimilla. Excelin avulla voidaan myös luoda erilaisia diagrammeja, tilukoita sekä tuoda siihen kuvia ulkopuolisista lähteistä. Exceliin voidaan myös tehdä kaavoja, joiden avulla sillä voidaan kätevästi laskea esimerkiksi eri solujen arvoja yhteen. Koska Excelistä löytyy VBA-moduuli, se voidaan ohjelmoida toimimaan yhdessä toisten ohjelmien kanssa.

3. Automatisoinnin toteuttaminen

3.1 Ohjelmointikielen valinta

Koska tämän opinnäytetyön aiheena on selvittää automatisointimenetelmiä Proxion Oy:n suunnittelijoiden työn helpottamiseksi, vaikuttaa ohjelmointikielen valintaan useita eri tekijöitä. Alle on lisätty valintakriteerejä.

- Ohjelmointikielillä pitäisi pystyä tekemään vaaditut automatisoinnit
- Ohjelmointikieltä tulisi pystyä käyttämään kyseisissä suunnitteluohjelmissa
- Proxion Oy:ltä tulisi löytyä osaamista ohjelmoida valitulla ohjelmointikielillä

- Ohjelmointikielellä tehtyä ohjelmaa tulisi pystyä päivittämään tarpeen mukaan uusiin kohteisiin, mikäli tarve vaatii
- Suunnitteluohjelman päivittyessä uudempaan tulisi automatisoinnin toimia ilman suuria muutoksia.
- Käytön helppous.
- Tiedon löytämisen helppous, mahdollisia tulevia päivityksiä varten

Esimerkkihjelmaa AutoCadin ja Excelin kaksisuuntaisen yhteyden toteuttamiseksi alettiin tehdä VBA-ohjelmointikieltä käyttäen. Esimerkkihjelmaa alettiin toteuttaa VBA:lla, koska se on yksinkertainen ja AutoCad ja Excel sallivat molemmat ohjelmien tekemisen VBA:n avulla. Tämä mahdollistaa laajojenkin ohjelmien kirjoittamisen sekä AutoCadin ja Excelin välisten yhteyksien tekemisen. Lisäksi VBA on hyvin suosittu AutoCad-käyttäjien keskuudessa ja aiheeseen liittyen internetistä löytyy paljon tietoa. Tämän lisäksi Proxionin suunnittelijat ovat käyttäneet sitä aikaisemminkin ja tehneet ohjelmia, jotka ovat edelleen käytössä. Lisäksi automatisointia lähdettiin toteuttamaan pohjaan, jossa oli aiemmin VBA-ohjelmointikielellä tehtyjä makroja. Koska VBA on hyvin suosittu AutoCadin ohjelmointiin, löytyi internetin keskustelufoorumeilta paljon tietoa aiheesta. Lisäksi vanhat jäsenet olivat hyvin avuliaita auttamaan VBA:han liittyvissä ongelmissa. VBA:lla kirjoitettua ohjelmaa on suhteellisen helppo lähteä päivittämään ja tietoaikin löytyy melko helposti. Myös keskustelut Proxion Oy sähköratasuunnittelija Jyri Taivaisen kanssa tukivat päätöstä automatisoinnin toteuttamiseksi VBA-kielellä, hänen aikaisemmin tekemiinsä ohjelmiin. Myös VBA.NET-kielen käyttöä harkittiin, mutta keskustelut suunnittelijoiden kanssa johtivat siihen lopputulokseen, että valittiin kuitenkin VBA ohjelmointikieleksi. Mikäli VBA.NET olisi valittu, olisi jo aikaisemmin tehdyt ohjelmat pitänyt muuttaa VBA.NET-kielelle, sillä kaikki ohjelmat haluttiin toimimaan samaan Excel-tiedostoon.

Opinnäytetyön aikana käytiin läpi useita vaihtoehtoja, ennen kuin päädyttiin VBA:han.

AutoCad-ohjelmaan eri vaihtoehtoja oli enemmän tarjolla kuin mitä CADS Planner-ohjelmaan. AutoCad-ohjelman oma sisäinen ohjelmointikieli AutoLISP oli yksi vaihtoehto automatisoinnin toteuttamiseen. Jos ohjelmointia olisi toteutettu AutoLISP-kieltä käyttäen olisi jouduttu joka tapauksessa käyttämään jotakin toista ohjelmointikieltä, mikäli AutoCad ja Excel olisi haluttu siirtämään tietoa molempiin suuntiin. AutoLISP-kielen ongelmana oli myös sen tuntemattomuus Proxionin suunnittelijoiden keskuudessa. Koska itseltänikään ei löydy kokemusta ohjelmoinnista tällä menetelmällä, olisi ohjelmointi vaatinut perusteellista perehtymistä kieleen. Näiden seikkojen takia AutoLISP jätettiin pois. Yhtenä vaihtoehtona automatisointiin ilmeni ObjectArx-rajapinnan käyttö. Tällöin ohjelmointi olisi pitänyt toteuttaa jotakin C-kieltä tai VB.NET-ohjelmointikieltä käyttäen. Löytämäni tiedon perusteella ObjectArx:lla tapahtuva ohjelmointi on kohtuullisen hankalaa ja vaatii perehtymistä sii-

hen. ObjectArx:llä tehdyt ohjelmakomennot ovat pitkiä ja kokemattomalle hankalia. Koska Proxion Oy suunnittelijoilla ja itselläni ei ollut kokemusta ObjectArx:n käytöstä, jätettiin sekin pois.

CADS Planner-ohjelmassa vaihtoehtoja löytyi niukemmin. Internetin kautta ei tietoa CADS Planner-ohjelman automatisoinnista löytynyt juuri lainkaan ja tietolähteenä CADS Plannerin kohdalla toimikin pääasiassa heidän oma tukipalvelunsa. Ainoa CADS Planner-ohjelman ohjelmointimahdollisuus johon opinnäytetyön aikana törmättiin, oli CADS Plannerin oma K-ohjelmointikieli.

CADS Planner-ohjelman ja Excel-taulukkolaskentaohjelman kaksisuuntaisen yhteyden luominen osoittautui haastavaksi. 8.2.2012 kävin sähköpostikeskustelua Kyndata Oy:n tuotantojohtaja Aki Nupposen kanssa. Nupposen mukaan millään ulkoisella ohjelmalla ei ole mahdollista päästä käsiksi CADS Planner-ohjelman kuvatietokantaan. Kuvatietokanta sisältää kaiken mitä CADS Plannerin-kuvissa on. CADS Planner-ohjelman K-ohjelmointikielen avulla päästään kyllä käsiksi kuvatietokantaan, mutta sillä tehtyjä ohjelmia ei voi suorittaa muualta kuin CADS ohjelmasta. Nupponen epäili, että K-ohjelmointikielillä ei ole mahdollista tehdä ohjelmaa, joka reagoisi reaaliajassa muutoksiin, mutta myös käyttäjän käskystä tapahtuva reagointi tulee olemaan erittäin haastavaa. Nupposen mukaan heillä Kyndatalla on omissa sovelluksissaan paljon reagoiteja ulkopuolisten (lähinnä tietokantojen) kanssa, mutta työkalut joita he ovat käyttäneet, ovat kds-rajapinnassa, jota Kyndata ei julkaise ulospäin. Nupposen mielestä molempiin suuntiin oleva yhteys Excelin ja CADS-ohjelman välillä ei ole mahdoton, mutta todella haastava, koska yhteyden aikaan saamiseksi tarvittaisiin ohjelmointia kahdella eri kielellä. K-ohjelmalla CADS Planneriin ja esim. VBA:lla AutoCadiin. Koska K-ohjelmointikieli ei ole Proxion Oy:n työntekijöille eikä itselleni tuttu, jätettiin kyseisten ohjelmien välisen yhteyden toteuttaminen tekemättä opinnäytetyön aikana. CADS-ohjelman kohdalta keskityttiin symbolien automaattiseen sijoittamiseen skriptin avulla.

3.2 VBA ja tulevaisuus

Microsoft päätti 1.7.2007 lopettaa uusien VBA lisenssien levittämisen. Microsoft on ilmoittanut, ettei julkaise VBA lisäyksiä enää tulevaisuudessa. AutoDesk kuitenkin tarjoaa epävirallista tukea ohjelmiinsa ainakin 2012-vuoden julkaisuissaan. Ohjelmistokehittäjät kannustavatkin loppukäyttäjiä siirtymään Microsoft .NET työskentely-ympäristössä toimiviin ohjelmointikieliin kuten VB .NET, C#, C++ jne. AutoCad 2010 version jälkeen VBA-moduuli ei ole enää ollut valmiina AutoCadin mukana, vaan se on pitänyt ladata erikseen AutoDeskin internetsivujen kautta. Vaikka AutoCad ei enää virallisesti tue VBA-ohjelmointikieltä, käytimme sitä kuitenkin esimerkkiohjelman tekemiseen. Keskustelin tästä asiasta opinnäytetyötä ohjaavan Proxionin suunnittelijan sekä opettajan kanssa ja tulimme siihen lopputulokseen, että VBA:ta käytettäisiin. Tulevaisuudessa VBA:lla tehty ohjelma

voitaisiin tarvittaessa kääntää VB.NET-kielille. On kuitenkin todennäköistä, että VBA sovelluksia on saatavilla jatkossakin, sillä VBA on hyvin suosittu loppukäyttäjien keskuudessa. Vierailtuamme Autodeskin sivuilla, josta VBA-moduuli on mahdollista ladata huomasimme, että VBA-moduuli latausversio AutoCad 2013-versioonkin on jo saatavilla.

4. Toteutus

4.1 Selvitystyö

Aluksi työssä selvitettiin, millä tavoilla kyseisiä suunnitteluohjelmia voi automatisoida. Kaikki menetelmät eivät välttämättä sovellu halutun kaltaiseen automatisointiin. Ei myöskään ollut täysin varmaa, että AutoCad- ja CADs-Planner suunnitteluohjelmiin soveltuvat samat automatisointimenetelmät. Lisäksi molemmista suunnitteluohjelmista löytyvät omat sisäiset ohjelmointikielensä, joita voidaan luonnollisesti käyttää vain kyseisiin ohjelmiin.

Selvitystyö lähti liikkeelle etsimällä internetin kautta eri vaihtoehtoja automatisoinnille. Tietoa löytyi mm. CADs Planner-tukipalvelun kautta, vanhoista opinnäytetöistä, Autodeskin sivuilta sekä yksityisten AutoCad-osaajien ylläpitämien kotisivujen kautta. Kuten aikaisemmin mainitsin yhtenä hyvänä tietolähteenä toimi keskustelupalstat, joissa AutoCad-käyttäjät keskustelivat ohjelmasta, siihen liittyvistä ongelmista sekä sen ohjelmoinnista. TheSwamp.org-niminen keskustelupalsta oli yksi parhaista tietolähteistä selvitystyön aikana. Sitä kautta sain itsekin erään ystävällisen käyttäjän minulle lähettämän VBA-koodin, jonka pohjalta lähdettiin opettajan avustuksella tekemään esimerkkiohjelmaa Excelin ja AutoCadin kaksisuuntaista yhteyttä varten. Lisäksi tietoa etsittiin tutustumalla itse ohjelmiin ja tutkimalla niiden eri ominaisuuksia, tämä tarkoitti käytännössä eri toimintojen kokeilemistä sekä ohjeiden etsimistä omista aputietokannoista. Kuitenkaan kummastakaan ohjelmasta ei löytynyt työkalua tai toimintoa, jota olisi suoraan voinut hyödyntää halutunlaisen lopputuloksen aikaansaamiseksi.

Koska CADs Planner on suomalainen ohjelma, ei siitä löytynyt yhtä hyvin tietoa internetin kautta, kuin mitä AutoCadista löytyi. Ainoat CADs-ohjelmaa koskevat tietolähteet internetissä, mitä onnistuin löytämään, olivat vanhoja opinnäytetöitä. Niissäkään ei suoraan käsitelty automatisointia, muuta kuin korkeintaan sivuamalla aihetta. CADs Planner-ohjelman kohdalla tietoa löytyi lähinnä sen omasta aputietokannasta ja sen kehittäjän Kyndata Oy:n tukipalvelun kautta. CADs Planner ei omaa yhtä suurta käyttäjäkuntaa kuin AutoCad ja tämä heijastui suoraan tiedon löytymiseen internetistä.

4.2 Skriptien selvittäminen

Sähköratapylväiden ja turvalaitteiden automaattista sijoittamista kuviin lähdettiin tekemään skriptitiedostojen kautta. Skriptitiedoston osalta tärkein selvitettävä asia oli järjestys, johon skripti tulisi kirjoittaa. Skriptitiedostossa järjestys on tärkeä, sillä jos käskyt eivät ole oikeassa muodossa ja oikeassa järjestyksessä, skripti ei toimi tai toimii väärin. AutoCadin osalta skriptin kirjoittamisesta löytyi kohtuullisen hyvin tietoa. Lisäksi skriptiä selvitettiin kokeilemalla käskyjen syöttämistä itse ohjelmaan. Oikean järjestyksen löydyttyä voitiin Exceliin kirjoittaa VBA-ohjelmointikieltä käyttäen makro, joka kirjoittaisi skriptin sähköratapylväille. Sähköratasuunnittelija Jyri Taivainen oli jo aikaisemmin tehnyt Excel-taulukkolaskentaohjelmaan makron, joka laski geometriatietojen perusteella sähköratapylväiden paikat. Tätä voitiin käyttää hyödyksi myös skriptiä tehtäessä. Taivainen kirjoitti Exceliin uuden makron joka hakisi pylväiden paikat ja kirjoittaisi skriptin. Tämä skriptitiedosto voidaan ajaa AutoCad-ohjelmasta jolloin se suorittaa skriptissä olevat komentojonot ja asettaa sähköratapylväät oikeille paikoilleen.

CADS-suunnitteluohjelmassa skriptitiedosto selvitettiin pääasiassa syöttämällä ja kokeilemalla komentoja ohjelmaan. Internetistä en onnistunut löytämään tietoa CADS Planner-ohjelman skriptistä. Selvittäminen oli siinä mielessä helpompaa, koska mallina voitiin käyttää AutoCadista saatua tietoa ja kokemuksia. Lisäksi CADS Planner-ohjelman suomenkielinen käyttöjärjestelmä helpotti selvittämistä. Skriptin valmistuttua alettiin tehdä ohjelmaa Excel-työkirjaan, joka laskisi tarvittavat sijainnit ja tekisi skriptin kuten AutoCadissa. Koska CADS planner-ohjelmaan ei ollut valmiina vastaavanlaista Excel-taulukkoa, kun mikä AutoCadissa on, alettiin jo olemassa olevan AutoCadin Excel-tiedoston pohjalta tehdä vastaavanlaista CADS Planner-ohjelmaan soveltuva Exceliä.

4.3 Skripti-tiedoston rakenne AutoCad ja CADS ohjelmissa

Kuten on jo tullut ilmi, sähköratapylväiden ja turvalaitteiden sijoittamiseen käytettiin yksinkertaista komentojonoa eli skriptitiedostoa. Skriptiä tehtäessä tulee pylväiden ja turvalaitteiden sijaintikoordinaatit olla selvillä. Tehtäessä skriptiä AutoCadiin, ilmeni joitakin pieniä ongelmia. Esimerkiksi Osnap-toiminto aiheutti skriptiin virhettä hakemalla seuraavan pylväsblokin paikan edellisestä blokkista. Tämä täytyi ottaa huomioon skriptiä kirjoitettaessa. Tehtäessä skriptitiedostoa AutoCadiin kirjoitetaan tiedosto alla olevaan muotoon.

```

osmode
0
_.insert
Blokin nimi
_._Rotate -138.672
_._Scale 1
2440532.5719311,7055801.85493941
text
2440534.98551296,7055798.85117339
3
-48.672
519-6
osmode
7

```

Yllä-olevassa skriptissä osmode-komento käynnistää osnap-toiminnon, joka kuitataan rivinvaihdolla (yhtä kuin enterin painaminen ohjelmassa). Seuraavalla rivillä oleva 0 ottaa osnap:n valitut kriteerit pois päältä (valitsee 0 kpl. toimintoja). Insert-komento lisää kuvaan blokin, joka valitaan kirjoittamalla nimi, jolla blokki on tallennettu AutoCadiin. Insert-komennon edessä olevat _ ja . määrittävät komennon toimivuuden kaikissa ohjelmissa kielestä tai versiosta riippumatta. Rotate-komennolla ja sen perässä olevalla numerolla määritetään blokin kiertokulma. Scale-komento määrittää blokin koon ja koko 1 pitää sen ennallaan. Scale-komennon jälkeen tulevat blokin koordinaatit. Tämän jälkeen tuleva text-komento lisää tekstiä kuvaan, jonka jälkeen tuleva koordinaatti määrittää tekstipaikan. Seuraavana tuleva 3 määrittää tekstin koon, jonka jälkeen merkataan tekstin kulma ja lopuksi itse lisättävä teksti. Viimeisenä kohtana skriptissä on osmode-komento, joka kytkee osnap-toiminnon takaisin päälle. Tämän komennon jälkeen tuleva numero kytkee osnap-toiminnon alta käyttöön halutut ominaisuudet. Scriptiä kirjoitettaessa täytyy ottaa huomioon, että se kirjoitetaan samaan muotoon ja samassa järjestyksessä, kuin mitä kirjoitettaisiin itse AutoCadin komentoriville. Tämän takia esimerkiksi scale-komento on pakko sisällyttää scriptiin.

Skriptissä kukin komento on omalla rivillään tai peräkkäin erotettuna välilyönnillä, joka on skriptissä sama kuin enterin painallus. Skripti kirjoitetaan esimerkiksi notepadille ja tallennetaan src-päätteellä. Tämän jälkeen skriptitiedosto avataan AutoCadista joko komennolla script tai valitsemalla AutoCadin Manage välilehden alta run script-komento. Tämän jälkeen valitaan skriptitiedosto sijainnista johon se on tallennettu.

Sähköratasuunnittelijoiden käytössä on Excel-taulukko, johon on tehty makroja, jotka hakevat mm. sähköratapylvään sijaintikoordinaatit, AutoCad-ohjelman koordinaatit yms. Tähän samaan Excel-taulukkoon on tehty makro, joka kirjoittaa Excelissä olevien tietojen perusteella tarvittavan skriptin. Ensin määritetään lähtökohtatietojen avulla mm. pylvään sijaintitiedot, kulma jossa pylvään tulee olla yms. Kun sijaintitiedot ja koordinaatit ovat selvillä, voidaan ajaa makro, joka muodostaa edellisten tietojen avulla koordinaatit AutoCad-ohjelmaan. Samalla makro tekee scr-päätteisen tiedoston jossa skripti on. Makro tulostaa skriptin yllä olevan esimerkin mukaisesti. Komentojonoja tulee peräkkäin niin monta kuin Excel-tiedostossa on kohtia. Kun skripti on tulostettu, voidaan se ajaa AutoCad ohjelmasta ja sähköratapylväät sijoittuvat oikeille paikoilleen AutoCad-kuvassa.

CADS Planer-ohjelmassa skriptien kirjoittaminen on mielestäni helpompaa verrattuna AutoCadiin. Nyt skripti kirjoitetaan samalla tavalla kuin AutoCadissakin, esim. notepadille. Komennot laitetaan listaksi ja hyväksytään enterillä. Kun komentojono on valmis, se tallennetaan mac-päätteellä samaan kansioon josta CADS Planner-ohjelmakin löytyy. Skripti käynnistetään kirjoittamalla CADS Planner-ohjelman komentoriville skriptitiedoston nimi ja hyväksytään enterillä jolloin skripti käynnistyy. Alla esitetty esimerkiskripti lisää opastimen kuvaan.

```
symboli^#h^#2-PÄÄOPASTIN_O^#2539796.074,7127626.7543,0^#1^#1^#90^#653+275^#655^#
```

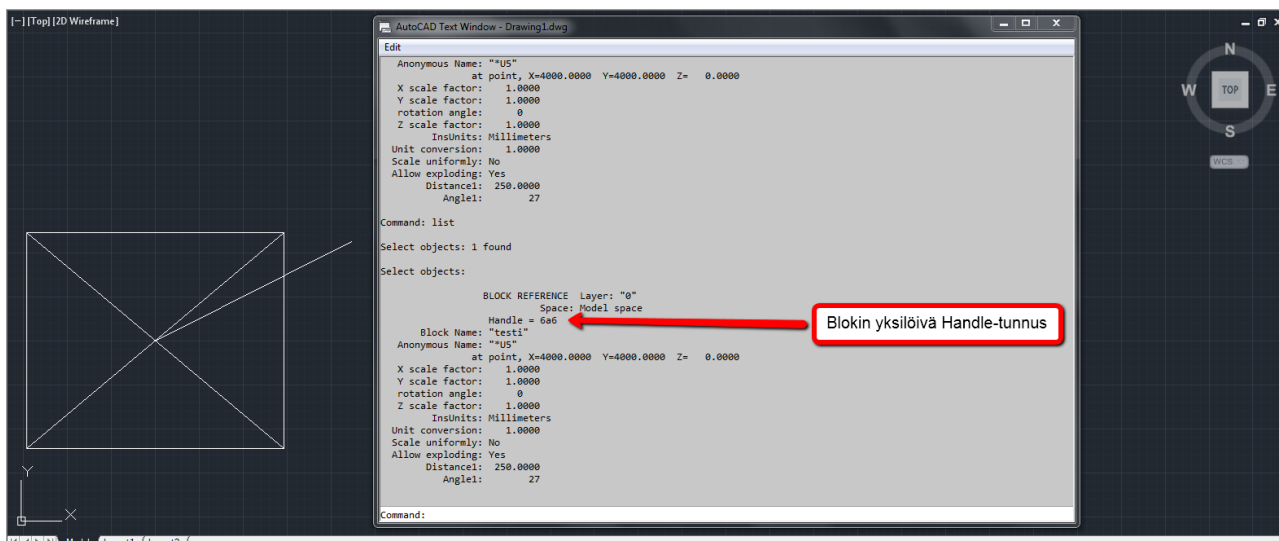
Symboli-komennolla käsketään ohjelmaa lisäämään symboli kuvaan. Komentojonossa ^# merkitsee enterin painallusta, eli se hyväksyy komennon. 2-PÄÄOPASTIN_O on tallennetun symbolin nimi, joka tässä tapauksessa tarkoittaa 2-käsitteistä pääopastinta. 2539796.074,7127626.7543,0 ovat symbolin sijoituskoordinaatit X,Y ja Z-akseleille. Koordinaattien jälkeen tuleva 1 merkitsee kokokerrointa X-akselin suunnassa, eli se määrittää symbolin venymän X-akselin suuntaan. Seuraava 1 on kokokerroin Y-akselin suuntaan, ja venyttää symbolin tarvittaessa Y-akselin suuntaan. Seuraavaksi skriptissä tulee symbolin kiertokulma, joka on tässä tapauksessa 90. Viimeisenä skriptiin kirjoitetaan atribuuttitiedot. Tällä symbolilla attribuutteina ovat sijaintitieto 653+275 (km+m) ja opastimen nimi 655.

4.4 Esimerkiohjelman tekeminen

AutoCadin esimerkiohjelman kirjoittaminen toteutettiin lehtori Markku Halttusen sekä sähkörata suunnittelija Jyri Taivaisen kanssa. Sain TheSwamp.org keskustelufoorumilta Chuck Hardin nimiseltä vanhalta käyttäjältä VBA-koodimalleja, joita apuna käyttäen lähdettiin kirjoittamaan omaa ohjelmaa. Hardin myös opasti aina tarvittaessa ja vastasi kysymyksiin, mitä ohjelmaa tehtäessä

ilmeni. Tarkoituksena oli tehdä esimerkkiohjelma, joka siirtää halutut blokkien tiedot AutoCadista Exceeliin ja päinvastoin.

Ensimmäisenä ongelmana oli se, miten blokit voitaisiin yksilöidä ja sitä kautta tunnistaa. Samassa kuvassa on hyvin usein monta samaa blokkia, jotka pitäisi jollain menetelmällä pystyä erottamaan toisistaan. Ongelmaan löytyi ratkaisu ns. handle-tunnuksen kautta. Selvitystyön aikana ilmeni, että jokaisella blokilla on ns. handle-tunnus. Handle-tunnus on AutoCadin jokaiselle blokille antama oma yksilöivä tunnus, joka erottaa sen muista samanlaisista blokeista. AutoCad antaa handle-tunnuksen blokille sen lisäämisvaiheessa. Handle-tietoa käyttämällä voitiin kirjoittaa ohjelman joka ”ottaa kiinni” tietystä blokista ja siirtää sen tiedot Exceeliin.

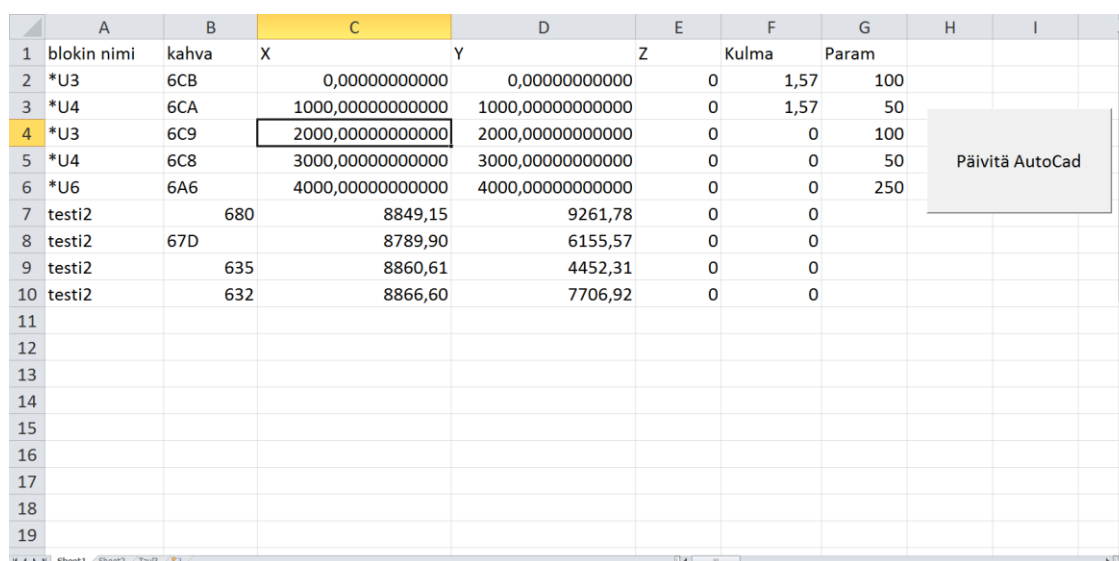


Kuva 1. Kuvassa on avattu List-komennolla informaatio-ikkuna, josta näkee blokin Handle-tunnuksen.

Useiden epäonnistumisien jälkeen saimme ohjelman toimimaan pääpiirteittäin. Päästiin näkemään, miten se toimisi käytännössä ja, että halutun kaltainen yhteys oli mahdollista toteuttaa. Ohjelmassa kuitenkin ilmeni pieniä ongelmia siellä täällä, joita täytyi vielä korjata. Ohjelma mm. saattoi antaa vikailmoituksen syystä, jota emme osanneet sanoa. Lisäksi alkuun ongelmia oli myös Excel-tiedostotyyppien kanssa. AutoCadissa ja Excelissä molemmissa oli oma ohjelmistokoodinsa, jotka tekivät muutoksia myös toiseen ohjelmaan. Excel-taulukon tulee olla tiedostotyyppiä xlsm, joka sallii makrojen käyttöönoton. Jos tiedosto oli tallennettu toisen tyyppiseksi, makrot eivät toimineet. Alkuun näytti siltä, että AutoCadiin tehty ohjelma ei osaa viedä tietoja xlsm-tyyppiseen Excel tiedostoon vaan sitä yritettäessä ohjelma kaatui. Tämän takia aluksi vaikutti siltä, että ohjelman toimimiseen vaadittaisiin kaksi eri Excel-tiedostoa. Toinen xlsm-tyyppinen, joka sallii makrot Excel-ohjelmassa ja toinen xlxs-tyyppiä joka ei salli makroja. Yhtenä ratkaisuna tähän on-

gelmaan suunniteltiin koodia, joka kopioisi xlsx-muotoisesta Excel-taulukosta arvot xlsx-muotoiseen taulukkoon. Tämän ratkaisun haittana vain on se, että yksi ”turha” Excel-tiedosto täytyy olla olemassa ikään kuin väliaikaisena varastona johon AutoCad tuo tiedot kuvasta. Lisäksi tällainen toimintamalli olisi ollut kömpelö käyttää ja virheiden tekeminen olisi ollut mahdollista jos muutoksia olisi tehty väärään työnkirjaan. Tarkemman tarkastelun jälkeen luulimme, että ongelma johtui tietokonekohtaisista asetuksista, koska sama ohjelma ilman minkäänlaisia muutoksia koodiin, toimi moitteetta toisella koneella, mutta ei toiminut toisella koneella. Myöhemmin tämäkin oletamus kuitenkin osoittautui vääräksi ja eräs osa ohjelmakoodista aiheutti tämän ongelman.

Ongelmia aiheuttanut koodinosa oli sijoitettu ns. blokin muokkaustapahtumaan. Alkuun ohjelma toimi siten, että blokia jollain tavalla muokattaessa esim. siirrettäessä tai uusia blokkeja lisättäessä AutoCad siirsi uudet tiedot Exceliin heti muutoksen tapahduttua. Tämä osa ohjelmas- ta kuitenkin jätettiin pois, koska ei katsottu tarpeelliseksi päivittää jokaista blokin muutosta erikseen Exceliin. Lisäksi tällainen toimintaperiaate hidastaisi ohjelman toimintaa skriptiä käytettäessä, koska ohjelma veisi tiedot Exceliin jokaisen blokin lisäämisen yhteydessä ja skriptissä saattoi olla useita kymmeniä blokkeja. Tämän katsottiin olevan tarpeetonta ja kömpelöä. Paljon käytännöllisempi ratkaisu on, että tiedot vietään Exceliin tallennustapahtuman yhteydessä. Tällöin skripti voidaan ensin ajaa kuvaan ja tämän jälkeen tallentaa ja saada tiedot uusista blokeista Exceliin. Tiedonsiir- ron tapahtuessa tallennustoiminnon alla tietojen siirto ei myöskään unohdu, koska kuva täytyy aina tallentaa. Exceliin tehtiin painike, josta muutokset voitiin päivittää AutoCad-kuvaan. Exceliin tehty ohjelma reagoi ainoastaan siihen tehtäviin muutoksiin. Tämä tarkoittaa sitä, että taulukosta löyty- viin tietoihin täytyy tehdä jokin muutos jotta tiedonsiirto on mahdollista.



	A	B	C	D	E	F	G	H	I
1	blokin nimi	kahva	X	Y	Z	Kulma	Param		
2	*U3	6CB	0,000000000000	0,000000000000	0	1,57	100		
3	*U4	6CA	1000,000000000000	1000,000000000000	0	1,57	50		
4	*U3	6C9	2000,000000000000	2000,000000000000	0	0	100		
5	*U4	6C8	3000,000000000000	3000,000000000000	0	0	50		
6	*U6	6A6	4000,000000000000	4000,000000000000	0	0	250		
7	testi2	680	8849,15	9261,78	0	0			
8	testi2	67D	8789,90	6155,57	0	0			
9	testi2	635	8860,61	4452,31	0	0			
10	testi2	632	8866,60	7706,92	0	0			
11									
12									
13									
14									
15									
16									
17									
18									
19									

Kuva 2. Kuvassa näkyy Excel-työkirja, johon AutoCadkuvan tiedot siirtyvät.

Alla on esimerkkiohjelma, jonka avulla päästään muuttamaan AutoCadin dynaamisten blokien parametritietoja. Tätä ohjelmaa käytettiin mallina myös esimerkkiohjelman tekemiseen opinnäytetyössä. Kyseinen ohjelma löytyy osoitteesta:

<http://www.visiblevisual.com/index.php/AutoCad-VBA/edit-dynamic-block-settings.html>

```
Function editblock(ByVal Parametername, ByVal Newvalue, ByVal Blockname As String)
```

```
Dim ent As AcadEntity
```

```
Dim oBkRef As IAcadBlockReference
```

```
Dim oDynProp As AcadDynamicBlockReferenceProperty
```

```
Dim v As Variant
```

```
Dim I As Long
```

```
For Each ent In ThisDrawing.ModelSpace
```

```
    If ent.ObjectName = "AcDbBlockReference" Then
```

```
        Set oBkRef = ent
```

```
        If oBkRef.IsDynamicBlock = True Then 'Check if it is a dynamic block
```

```
            v = oBkRef.GetDynamicBlockProperties 'Get all Dynamic attributes
```

```
            If oBkRef.EffectiveName = Blockname Then
```

```
                For I = LBound(v) To UBound(v)
```

```
                    Set oDynProp = v(I)
```

```
                    ta = oDynProp.PropertyName
```

```
                    If Not IsArray(oDynProp.Value) Then 'Not an array
```

```
                        If oDynProp.PropertyName = Parametername Then
```

```
                            Str (oDynProp.Value)
```

```
                                'If Newvalue > 0 Then
```

```
                                    'Newvalue = Replace(Newvalue, ",", ".")
```

```
'End If
```

```
oDynProp.Value = Newvalue 'Set the new value
```

```
End If
```

```
End If
```

```
Next I
```

```
End If
```

```
End If
```

```
End If
```

```
Next ent
```

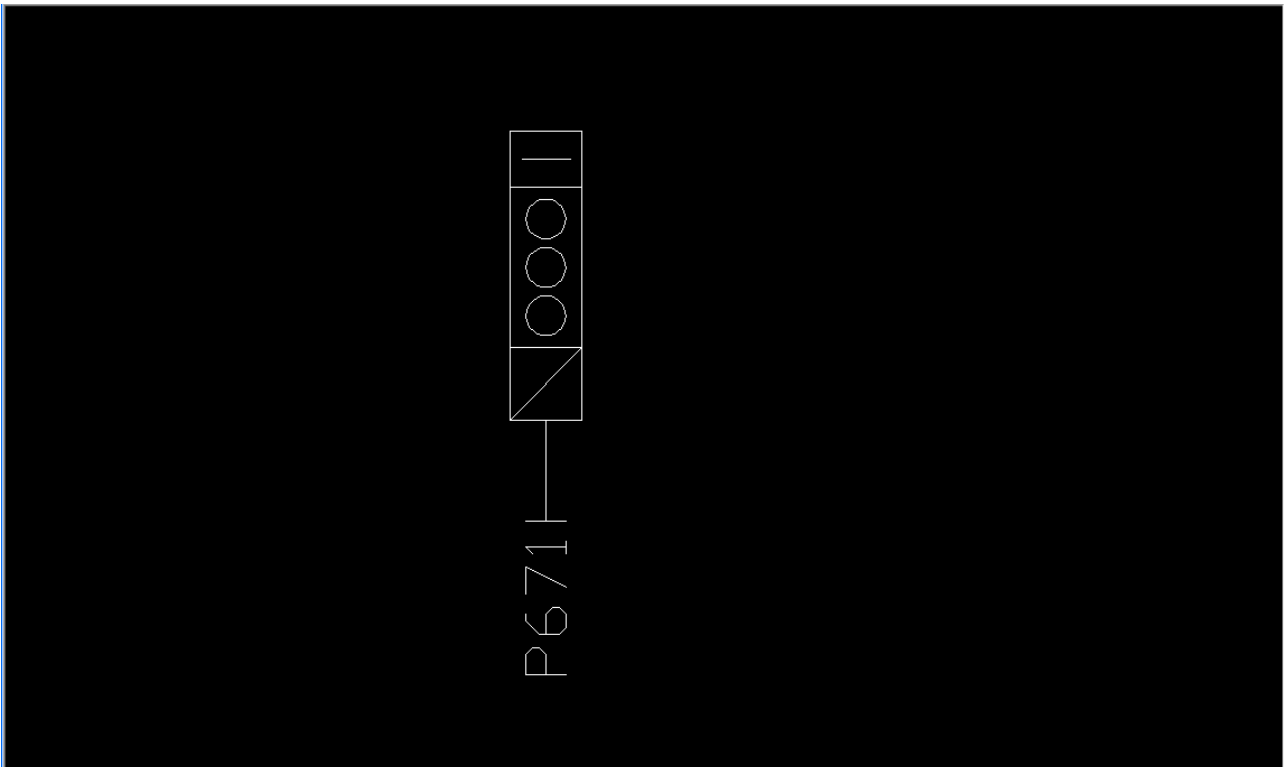
```
End Function
```

CADS Planner-ohjelman kohdalla ei tehty esimerkkiohjelmaa vaan ohjelmointia alettiin tehdä käyttöön tulevaan versioon, jota voidaan päivittää ja kehittää tarpeen mukaan. Kuten jo mainittua CADS-ohjelmassa ei ollut valmista Excel-tiedostoa, johon skriptin kirjoittava makro voitaisiin lisätä, joten sähköratasuunnittelun Excel-työkirjaa käytettiin pohjana myös turvalaitepuolelle. Ohjelman kirjoittaminen turvalaitesuunnittelun osalta toteutettiin Proxion Oy:n sähköratasuunnittelija Jyri Taivaisen kanssa.

Turvalaitesuunnittelijoiden tehtyä 1:10 000-kuvan, päästään hyödyntämään skriptiä blokkien asettamisessa 1:1000-kuvaan. Kun 1:10 000-kuva on valmis, ajetaan siitä määrälaskenta. Tämä tarkoittaa sitä, että CADS Planner-ohjelma tekee piirroksesta taulukon, johon se listaa kaikki siinä olevat symbolit, niiden määrät sekä attribuuttitiedot. Kun tiedot ovat määrälaskentataulukossa, voidaan ne siirtää sieltä Excel-tiedostoon. Sähköratapuolelle tehdyssä Excelissä oli jo valmiina joitakin makroja, joita voitiin käyttää suoraan myös turvalaitepuolen Excelissä. Joitakin makroja jouduttiin kuitenkin muokkaamaan tarkoitukseen sopivaksi, sekä kirjoittamaan kokonaan uusia makroja.

Ensin ohjelma tehtiin toimimaan siten, että geometriasuunnittelijoilta saatu geometria-tiedosto vietiin Excelliin ja määritettiin geometriakuvasta kilometrien pituudet. Tällä tarkoitetaan sitä, että geometria-kuvasta katsotaan miten pitkiä kilometrit siinä ovat, sillä ne eivät aina välttämättä ole tasan 1000 metriä. Näitä tietoja hyväksi käyttäen ohjelma sitten osasi laskea symboleille koordinaatit geometriakuvaan. Tämä toimintatapa piti kuitenkin muuttaa, koska ilmeni, että kaikki geometriatiedostot eivät olleetkaan samanlaisia vaan niissä oli pieniä eroja, joiden takia ohjelma ei olisi toiminut oikein. Tämä ongelma päätettiin ratkaista siten, että geometriatiedosto vietäisiin C-

asemalle tehtyyn kansioon ja nimettäisiin oikean nimiseksi. C-asemalta ohjelma sitten tunnistaisi geometriatiedon ja valitsisi sen mukaan sopivan toimintamallin ja hakisi tarvittavat tiedot Excelliin, jotta symbolien koordinaatit voitaisiin määrittää oikein. Kun turvalaitesuunnittelijat tekevät 1:10 000-kuvan, he antavat symboleille sijaintitiedon, jonka mukaan symbolit asettuvat paikalleen 1:1000-kuvassa. Excelissä on makro, joka hakee radan geometriatiedoista radan X- ja Y-koordinaatit. Määrälaskennassa olevien symbolien sijaintitietojen ja radan geometrian X- ja Y-koordinaattien pohjalta makrot laskevat symbolien sijaintikoordinaatit kuvaan. Excel myös kirjoittaa skriptitiedoston ja tallentaa sen automaattisesti CADS Planner-kansioon. Tämän jälkeen 1:1000-kuva avataan ja käynnistetään skripti, jolloin ohjelma asettaa symbolit automaattisesti paikoilleen kuvaan. Tavallisten symbolien asettaminen kuvaan onnistui suhteellisen helposti. Suurempia vaikeuksia tuotti sellaisten symbolien asettamisessa joihin itseensä piti asettaa toinen symboli seuraavan kuvan mukaisesti.

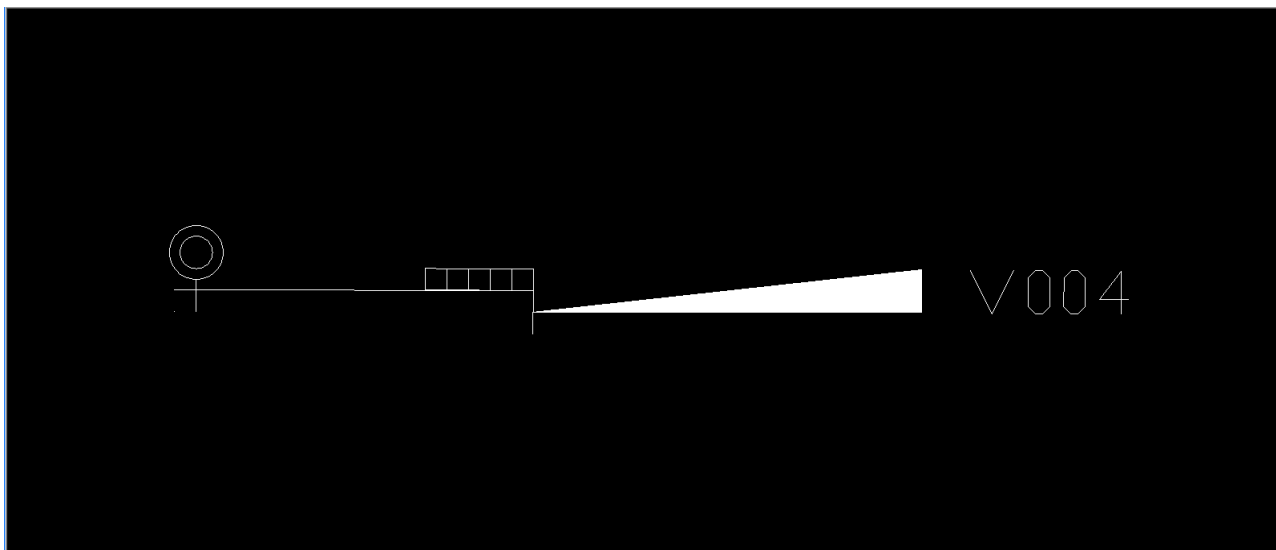


Kuva 3. Opastinta kuvaava symboli.

Kuvassa oleva symboli on opastin, johon on liitetty liikennepaikan alkua kuvaava symboli. Näiden symbolien asettaminen yhteen toteutettiin käytännössä siten, että liikennepaikka-merkin sijainti määräytyi opastimen sijaintikoordinaatin mukaan, jonka jälkeen sen Y-koordinaattiin

lisätään opastin-symbolin pituus. Jotta symboli tulee oikein kuvaan, täytyy sen kulman luonnollisesti olla oikea. Kulmaa joudutaan laskemaan ohjelmaan lisättyjen kaavojen avulla, jotka ohjelma suorittaa ja laskee symbolin kulman. Opastimien kohdalla liikennepaikkamerkit lisätään niin sanottuun lisäosaattribuuttiin 1:10 000-kuvassa. Se on tavallaan yksi symbolin lisätietokenttä, johon voidaan asettaa symboliin kohdistuvia tietoja. Kun liikennepaikkamerkit lisätään 1:10 000-kuvaan opastimen attribuutteihin, osaa ohjelma yhdistää saman nimiset liikennepaikkamerkit ja opastimet 1:1000-kuvaan.

Myös radalla olevia vaihteita kuvaaviin symboleihin tulee liittää osia. Jokaiseen vaihteeseen ei tule samoja osia, vaan ne vaihtelevat erikokoisten vaihteiden kesken. Tämä täytyy ottaa huomioon vaihteiden skriptiä kirjoittavassa ohjelmassa. Ohjelmaan kirjoitettiin osio, joka asettaa vaihteisiin niiden tyypistä riippuen oikeat osat paikoilleen automaattisesti. Kuten opastimien liikennepaikkamerkkien kohdalla, myös vaihteissa lisäosat merkataan 1:10 000-kuvassa vaihteen attribuuttitietoihin. Sieltä makro tunnistaa ne ja asettaa oikeita lisäosia oikean määrän oikeissa kulmissa 1:1000-kuvaan. Lisäosien kohdalla päädyttiin ratkaisuun, jossa RATO 6-ohjeistuksessa olevat esimerkit toimivat malleina lisäosien paikkojen ja määrien suhteen. Lisäosien eri variaatioita vaihteessa voi olla niin paljon, ettei jokaista vaihtoehtoa olisi käytännössä mahdollista tietää, joten lisäosat tulevat esimerkkien mukaan ja suunnittelijat voivat tarvittaessa tehdä niihin muutoksia. Attribuutteja käydään tarkemmin läpi seuraavassa kappaleessa.



Kuva 4. Kuvassa esimerkki vaihteen symbolista lisäosineen.

4.5 Esivalmistelut

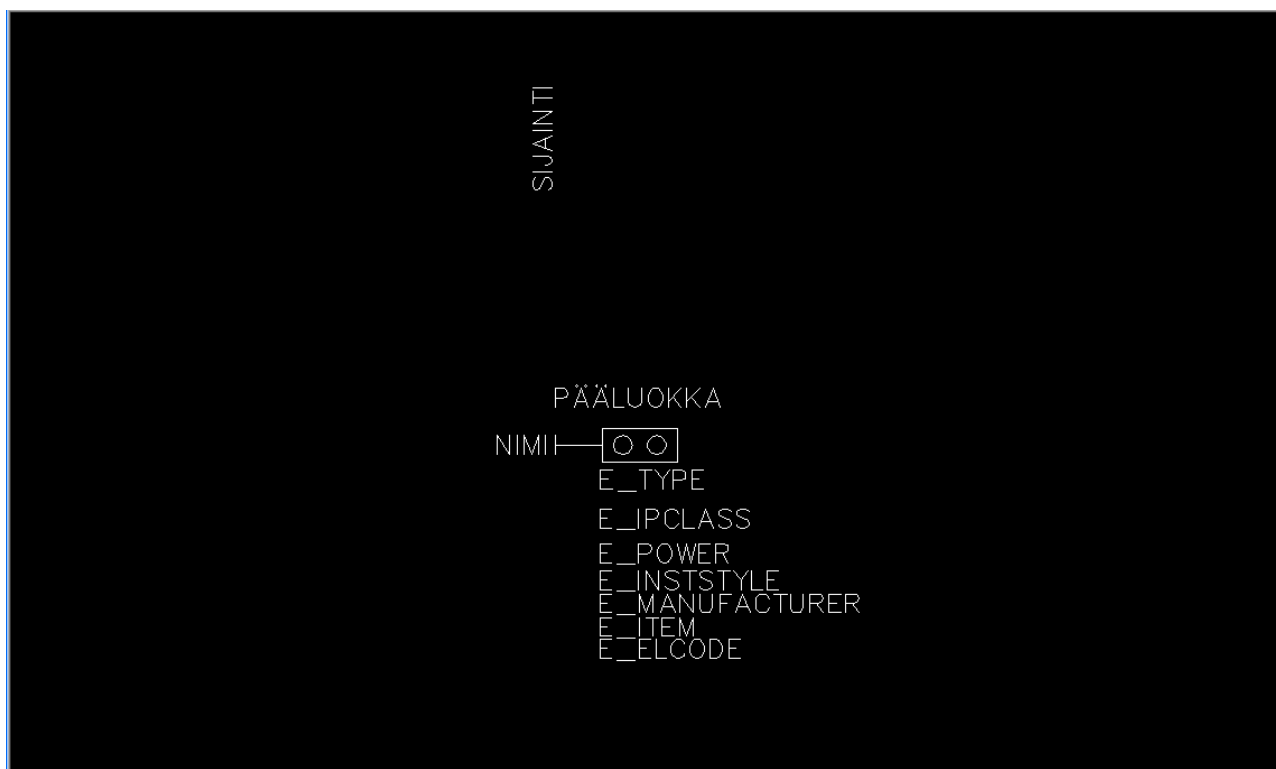
Jotta CADs Planner-ohjelmaan tehty automatisointi toimii, vaatii se hieman esivalmisteluja. CADs Planner-ohjelman symboleilla on ns. attribuuttitietoja, joihin voidaan kirjata esimerkiksi symbolin nimi tai sijaintitietoja. Ennen kuin ohjelmaa voidaan käyttää tulee symbolien attribuutteja muokata.

Kun aloitin opinnäytetyön oli määrälaskentaluetteloon mahdollista valita viisi eri attribuuttia. Tietoja, joita symboleista tarvittaisiin määrälaskentaluetteloon, on kuitenkin enemmän kuin tuo viisi. Tämän takia suunnittelijat ovat joutuneet laittamaan tietoja samojen attribuuttien alle, jotta ne kaikki saataisiin näkyviin määrälaskentaluetteloon. Yleensä suunnittelijat laittoivat nimi-, sijainti-, pääluokka- ja e-type -kohdat normaalisti oikein ja lisäsivät ylimääräiset tiedot maadoitus-attribuutin alle erottamalla aina uudet tiedot toisistaan puolipilkulla.

Kuitenkin opinnäytetyön aikana selvisi, että attribuuttitietoja on mahdollista ottaa käyttöön enemmänkin. Tämä kuitenkin vaatii sen, että attribuutit tavallaan kirjoitetaan symboleihin. Uusia attribuutteja voidaan symboleihin lisätä millä nimellä tahansa, mutta jos halutaan, että CADs tuo ylimääräiset attribuuttitiedot määrälaskentaluetteloon, tulee ne kirjoittaa oikeilla nimillä. Alla on listattu uudet attribuutit ja millä nimillä ne tulee kirjoittaa, jotta CADs tunnistaa ne attribuuteiksi.

Nimi määrälaskennassa:	Nimi jolla attribuutti tulee kirjoittaa symboliin:
IP-Luokka	E_IPCLASS
Teho	E_POWER
Asennustapa	E_INSTSTYLE
Viitetunnus	E_REFID
Symbolin teksti	E_SYMBTXT1
Valmistaja	E_MANUFACTURER
Nimike	E_ITEM
Sähkönumero	E_ELCODE

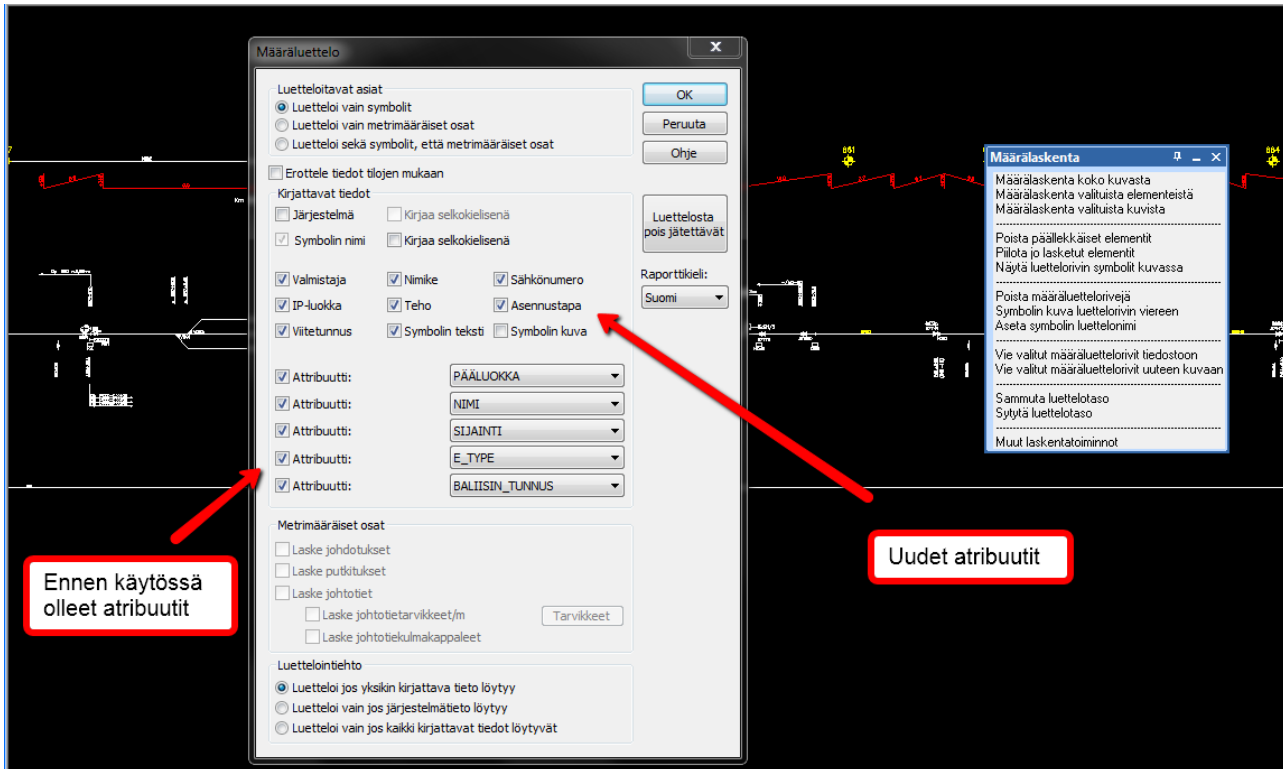
Lisäämällä nämä uudet attribuutit symboleihin saadaan attribuuttien määrä nostettua määrälaskennassa viiden sijasta kolmeentoista. Vaikka attribuutit joudutaan kirjoittamaan E_xxx muotoon, voidaan niille kuitenkin määrittää ns. kehote. Kehotteeksi voidaan laittaa mitä tahansa tietoa, jotta symbolin lisäämisvaiheessa ei tarvitse tietää, mitä mikäkin attribuuttitieto tarkoittaa. Kuitenkin määrälaskentaan CADs planner antaa uudet attribuutit uusilla nimillä. Tämän takia Excel-tiedostoon lisättiin sivu jolla kerrotaan, mitä mikäkin uusi attribuuttitieto vastaa.



Kuva 5. Alla opastinta kuvaava symboli lisättyine attribuuttitietoineen.

Automatisointiin uusia attribuutteja tarvittiin, koska symbolin attribuuttitietoihin lisättiin mm. radan numero jolle se asetetaan sekä lisäosa, johon voidaan merkata aiemmin mainitut symboliin liitettävät mahdolliset lisäosat. Uusien attribuuttien lisäämisen jälkeen keskustelin suunnittelijoiden kanssa, mitkä attribuuttitiedot kannattaa jatkossa valita viideksi attribuutiksi määrälaskentaan. Loput attribuuttitiedot tulevat sitten uusien lisättyjen attribuuttien kautta. Koska eri symboleissa ei välttämättä ole samoja attribuuttitietoja, piti ne aikaisemmin laittaa samojen attribuuttien alle. Nyt kun attribuutteja on käytössä tarpeeksi monta, on määrälaskentaan mahdollista tuoda kaikki symbolien attribuutit omina sarakkeinaan, mikä puolestaan selkeyttää toimintaa. Koska attribuutit pitää kirjoittaa symboleihin, tulee jokainen vanha symboli päivittää uudelleenlaisiksi ja lisätä niihin halutut

atribuutit. Tämä prosessi on kieltämättä työläs, mutta sitä ei tarvitse toteuttaa kuin kerran. Lisäksi siitä saatava hyöty tulee olemaan suuri.



Kuva 6. Kuvassa on esitetty määrälaskentaikkuna ja attribuuttien valintamahdollisuudet.

Eräs erikoinen piirre ilmeni määrälaskentaa ajettaessa. Kun määrälaskenta ajetaan CADS Planner-kuvasta, valitaan halutut tiedot taulukosta ja hyväksytään ne. Kun nämä tiedot on valittu ja niitä lähdetään siirtämään Exceeliin, tiedot jäävät tavallaan leikepöydälle. CADS kysyy tässä vaiheessa, haluatko siirtää valitut tiedot ja pyytää vahvistamaan valinnat enterillä. Jos valinnat kuitataan enterillä, CADS ei jostain syystä vie määrälaskentataulukon otsikkoja leikepöydälle vaan ainoastaan arvot. Jostain syystä myös nimi-attribuutti muuttaa paikkaa Excelissä. Tämä sotki makron toiminnan, joka siirtää tiedon Excelin-määräalaskenta välilehdeltä symbolit-välilehdelle. Symbolit-välilehdellä sijaintikoordinaatit lasketaan ja skripti kirjoitetaan. Tämän takia on tärkeää, ettei enteriä enää paineta määräalaskentaa siirrettäessä Exceeliin. Tässä toimintamallissa on helppo tehdä virheitä, joten Exceeliin kirjoitettiin makro, joka tunnistaa sarakkeet määräalaskenta-välilehdeltä niiden otsikkosarakkeiden perusteella, joten ei ole väliä missä järjestyksessä tiedot sinne tulevat.

Vaihteita kuvaavat symbolit kokivat muutoksia automatisoinnin yhteydessä. Vaihteiden symbolit ovat siitä poikkeuksellisia, että niitä ei kuvata samoilla symboleilla 1:10 000- ja 1:1000-kuvissa, vaan molemmissa kuvissa on omat symbolinsa kuvaamaan kutakin vaihdetta.

Ennen vaihteilla oli 1:1000-kuvassa useampia symboleita kuvaamassa eri asentoja joihin vaihteet saattoivat asettua. Nyt kuitenkin vaihteille tehtiin kaksi symbolia kuvaamaan kutakin vaihdetta. Puhutaan ns. oikean ja vasemman käden vaihteista. Tämä tarkoittaa sitä, että vaihteeseen ajetaan siten, että siitä voidaan jatkaa suoraan tai kääntyä vasemmalle tai oikealle riippuen vaihteesta. Kustakin vaihteesta siis tehtiin kaksi symbolia toinen kuvaamaan vasemmalle kääntyvää vaihdetta ja toinen oikealle kääntyvää vaihdetta. Myös 1:10 000-kuvassa on kaksi symbolia kuvaamaan molempien kätisiä vaihteita. Symbolit on nimetty siten, että 1:10 000- ja 1:1000-kuvassa symbolit ovat muuten saman nimisiä, mutta 1:10 000-kuvassa vaihteen nimen edessä on – merkki. Kun 1:10 000-kuvaan laitetaan symboli paikoilleen hakee, ohjelma vastaavan symbolin 1:1000-kuvaan. Symboleita myös muokattiin siten, että niihin liitettävä niin sanottu kieliviivaa kuvaava symboli laitettiin yhteen itse vaihdetta kuvaavan symbolin kanssa.

5. Kuinka tavoitteet saavutettiin?

Automatisoinnin toteuttaminen kyseisiin ohjelmiin vaatii suhteellisen paljon ohjelmointia sekä niihin perehtymistä. Koska itselläni ei ollut aikaisempaa kokemusta ohjelmien automatisoinnista eikä ohjelmoinnista, oli alkutilanne mielestäni haastava. Alkuun jopa tuntui, ettei keneltäkään löydy varsinaista kokemusta kyseisten ohjelmien automatisoinnista. Varsinkin CADS Planner-ohjelman kohdalla tietolähteitä tuntui olevan niukasti tarjolla. Vähitellen kuitenkin löytyi henkilö, joka saattoi tietää asiasta jotain tai ainakin tiesi henkilön, joka taas saattoi tietää asiasta jotain. Näin pikkuhiljaa edeten löytyi vähitellen tietoa sekä nettilinkkejä aiheesta. Kuten aikaisemmin tuli jo mainittua, AutoCadin kohdalla yllättävän hyviksi tietolähteiksi osoittautuivat eri keskustelupalstat, joissa hieman samantyyppisiin ongelmiin oli jo etsitty ratkaisuja. Kuitenkaan suoraan kyseisenlaisista ratkaisua ei kukaan ollut kysellyt. Niinpä päätin tehdä oman profiilin Swamp.org-nimiselle harrastelijasivustolle, joka ensitutustumisen kautta vaikutti asiantuntevalta ja myös käyttäjäkunnaltaan ystävälliseltä ja halukkaalta auttamaan AutoCad-käyttäjiä heidän ongelmissaan.

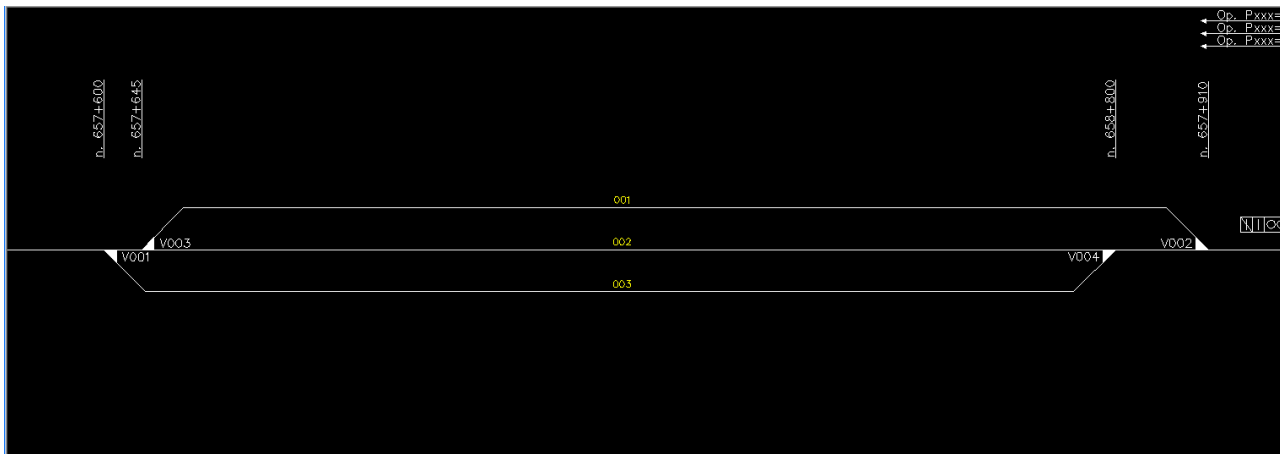
Tämän ansiosta esimerkkiohjelma AutoCadiin alkoi valmistua hyvää vauhtia, koska Chuck Hardin-niminen ystävällinen käyttäjä lähetti esimerkkiohjelman, josta oli hyvä lähteä jalostamaan sitä omiin tarkoitukseen sopivaksi. Lisäksi siitä nähtiin minkälaisella rakenteella halutunlainen yhteys voitiin toteuttaa. Chuck myös hyvin auliisti vastaili esittämiini kysymyksiin ja oli todella avulias. Myös lehtori Markku Halttunen auttoi paljon esimerkkiohjelman teossa ja varsinkin alkuvaiheessa hänen kanssaan tehtiin paljon yhteistyötä ja saatiin yhteys toimimaan molempiin suuntiin. Lopullisesti esimerkkiohjelma saatettiin loppuun Proxion Plan sähköratasuunnittelija Jyri Taisaisen kanssa. Jyrin avustuksella ohjelmaan onnistuttiin luomaan blokin kulman muutos sekä dynaamisen blokin attribuuttitietojen muokkaus Excelistä käsin.

Mielestäni AutoCadin osalta esimerkkiohjelman tekeminen onnistui hyvin. Siihen saatiin Excelin ja AutoCadin välinen molempiin suuntiin toimiva yhteys, joka siirtää halutut tiedot ohjelmasta toiseen. AutoCadin skriptitiedoston tekevä makro syntyi kokonaan Jyri Taivaisen toteuttamana. Hän muokkasi tekemäänsä Exceliä niin, että se myös kirjoittaa scriptin selvittämäni järjestykseen, jolloin sähköratapylväät saadaan oikeille paikoilleen. Tämä ominaisuus on ollut jo käytössä sähköratasuunnittelun puolella oikeissa työprojekteissa.

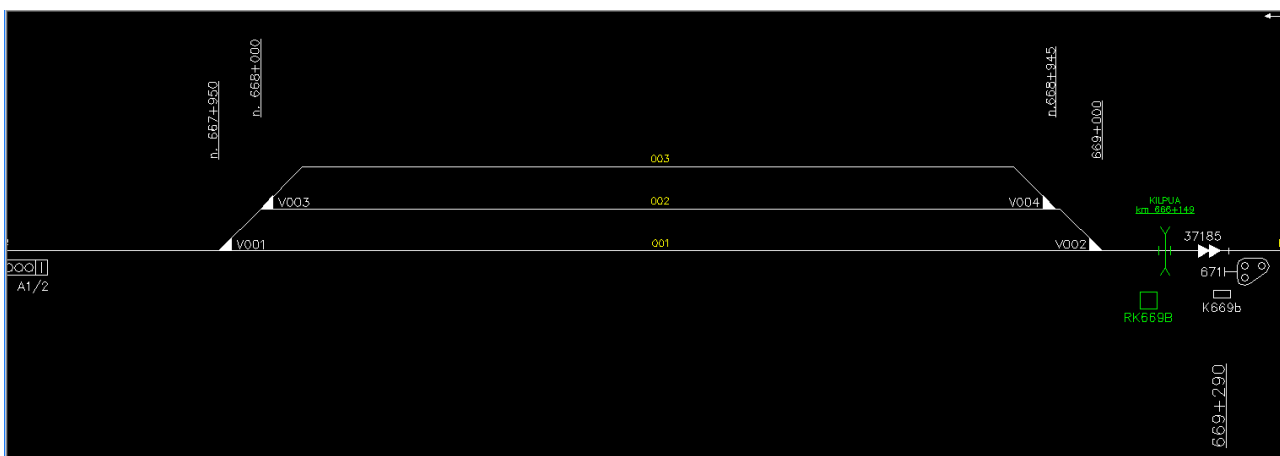
CADS Planner-ohjelman osalta parhaaksi tietolähteeksi osoittautui CADSin oma tukipalvelu johon Proxionilla oli jo olemassa sopimus, ja jota voitiin hyödyntää myös opinnäytetyön tekemiseen. Tukipalvelun kauttakaan ei suoraan osattu antaa apua automatisointiin, mutta heidän ystävällisesti selvitettyään asiaa sain apua Kydata Oy:n tuotantojohtaja Aki Nupposelta. Hänen kanssaan käytyä sähköpostikeskustelua olenkin käynyt läpi jo aikaisemmin tässä raportissa. Hänen kauttaan selvisi, että CADS ja Excel ohjelmien välisen yhteyden toteuttaminen tulisi olemaan todella haastava tehtävä, joka kaiken lisäksi vaatisi myös Proxionin ohjelmoijille uuden ohjelmointikielen opettelemisen. Tämän takia kaksisuuntainen yhteys ohjelmien välillä päätettiin jättää tulevaisuuteen ja keskittyä opinnäytetyön aikana lähinnä skriptien avulla tapahtuvaan automatisointiin. Koska Jyri Taivainen oli jo pystynyt Excelillä luomaan skriptitiedoston AutoCadin puolelle, kannatti tätä luonnollisesti käyttää pohjana myös CADS Plannerin automatisoinnissa. Exceliä jouduttiin muokkaamaan rankalla kädellä ja tietenkin erilaiset vaatimukset sähkörata- ja turvalaitesuunnittelun osalta toivat myös paljon uutta haluttuun lopputulokseen pääsemiseksi. Jo erilaisten blokkien ja symbolien määrässä oli eroa ohjelmien välillä. AutoCadissa tuli lisätä muutama erilainen blokki, kun turvalaitesuunnittelun puolella toisistaan poikkeavia symboleja oli paljon enemmän. Tämän takia ohjelmaan tuli tehdä melkein jokaiselle symbolille oma makro, jotta ne asettuvat oikein kuvassa. Lisäksi jo mainitut lisäosat tuottivat omat haasteensa.

Oman haasteensa asettivat myös raiteen numeron vaihdokset. Jokaisesta raiteesta saadaan oma geometriatiedostonsa, josta ilmenee myös raiteen numero. Raiteen numero määrittää mille raiteelle symboli asetetaan. Symboleihin lisättiin oma attribuutinsa tätä radannumeroa varten, jolla ohjelma sitten osasi asettaa symbolin oikealle raiteelle. Ratapihalla yksi raide saattaa tavallaan jakautua useampaan rataosuuteen, jolloin radan numerointikin saattoi vaihdella. Kaikissa tapauksissa ei ollut varmaa, että ratapihalla läpi menevä raiteen numero pysyi samana ratapihalle tultaessa ja sieltä lähdettäessä. Tämä aiheutti ongelman, joka tuli ratkaista, jotta symbolit asettuivat oikeiden raiteiden varsille. Kustakin rataosuudesta suunnittelijat saavat omat geometriatiedostonsa. Geometriasuunnittelijat numeroivat raiteet tehdessään niitä. Ongelma päätettiin ratkaista siten, että turvalaitesuunnittelijat käyttävät rataosuuksille samoja numeroita, joita geometriasuunnittelijatkin ovat käyttäneet, mutta läpimenevästä raiteesta käytetään aina samaa numeroa, esimer-

kiksi 0. Tällöin läpimenevän raiteen numero pysyy koko ajan samana ja symbolit asettuvat oikeille raiteille.



Kuva 7. Kuvassa on radan ensimmäinen ratapiha. Läpimenevä raide on tässä vaiheessa raidenumero 2.



Kuva 8. Tämä kuva on saman raiteen toinen ratapiha, jossa voi nähdä miten läpi menevän raiteen raidenumero muuttuu. Tässä läpi menevän raiteen raidenumero on 1.

6. Kehitysehdotuksia

Tässä opinnäytetyössä keskityttiin pääasiassa sähköratapylväiden ja turvalaitteiden automaattiseen sijoittamiseen sekä Excelin ja suunnitteluohjelmien väliseen yhteyteen. Kuitenkin suunnittelu työstä löytyy muitakin kohteita, joita olisi vielä mahdollista automatisoida ja kehittää. Tähän osioon olen listannut muutamia työn aikana esille tulleita kehitettäviä kohteita. Näistä mahdollisuuksista

olemme keskustelleet sähköratasuunnittelija Jyri Taivaisen kanssa. Nämä ehdotukset ovat tulleet suunnittelijoiden suunnalta.

6.1 CADS-ohjelman kuvien eri reviisioissa tapahtuvien muutosten automaattinen merkitseminen

Turvalaitesuunnittelussa valmiit kuvat lähetetään tilaajalle ja usein kuviin tulee muutoksia tai lisäyksiä. Kun kuva tulee takaisin, siitä käytetään nimitystä reviisio ja joku kirjain tai numero kuten reviisio A tai 1. Kun näitä reviisioita tulee ja muutoksia tehdään, tulee muutokset merkata omaan Excel-tiedostoonsa. Käytäntönä on ollut, että muutokset merkataan punaisella värillä Excelliin, jotta ne erottuvat muista tiedoista. Ajatuksena tähän malliin olisi, että yksi attribuuttitieto olisi varattu tälle kyseiselle toimenpiteelle. Turvalaitesuunnittelijan tehdessä muutoksia kuvaan, hän samalla voisi merkata reviision tunnuksen muokatun symbolin attribuuttitietoihin. Ajettaessa kuvasta määrälas-kenta ja tietoja vietäessä Excelliin attribuuttitiedossa näkyvä reviisiotunnus siirtyisi tietysti myös Excelliin. Excelliin tulisi merkitä mikä, reviisio on kysymyksessä, jotta ohjelma osaisi tunnistaa tämän hetkisen reviision. Excelliin voitaisiin tehdä makro, joka ensin tarkistaisi minkä reviisiotunnuksen suunnittelija on merkannut Excelliin ja tämän jälkeen etsisi symboleiden attribuuteista vastaavat tunnuksat. Tämän jälkeen se muuttaisi kyseiset tiedot punaiseksi eikä niitä tarvitsisi käydä yksitel-len muuttamassa. Tällä menettelyllä säästettäisiin aikaa ja vaivaa. Lisäksi tämänkaltainen automa-tisointi olisi suhteellisen helppo toteuttaa.

6.2 CADS-ohjelmasta ajettavien useiden eri luetteloiden ajaminen yhteen Excel-tiedostoon

CADS planner-suunnitteluohjelmasta on tähän mennessä jouduttu ajamaan useita eri luetteloita eri tietojen ulos saamiseksi ja siirtämiseksi eteenpäin. Tämä on johtunut pääasiassa siitä, ettei atri-buuttitietoja ole ollut aikaisemmin mahdollista ajaa kuvasta luetteloihin enempää kuin tuo mainittu viisi. Tämän takia kuvasta on täytynyt ajaa useampi eri luettelo, joissa ajettavia attribuuttitietoja on vaihdettu, jotta kaikki tarvittava tieto saadaan luetteloitua. Nyt yhteen luetteloon ajettavien attribuut-titietojen määrän noustessa kolmeentoista on kaikki tarvittavat attribuutit mahdollista saada yhteen luetteloon näkyviin. Tällöin luetteloita ei tarvitse ajaa kuin yksi. Tämä luettelo voidaan ajaa opinnäy-tetyön aikana valmistuneeseen Excelliin. Käytännön kokemukset näyttävät, tulisiko Excelliin kirjoit-taa lisää makroja joitakin uusia toimintoja varten. Tämä menettely luonnollisesti nopeuttaisi työ-skentelyä huomattavasti ja helpottaisi listojen seuraamista, koska kaikki tiedot löytyisivät samasta Excel-tiedostosta.

6.3 Raidenumerointien automaattinen paikalleen asettaminen ja oikeiden numeroiden hakeminen automaattisesti

Turvailaitesuunnittelussa raiteille annetaan raidenumero. Raidenumerolla tässä yhteydessä ei tarkoiteta raidenumeroa, jonka perusteella symbolit hakeutuvat oikeille raiteille, vaan numeroa, joilla eri raideosuudet voidaan paikallistaa. Esimerkiksi vikatilanteessa eri raideosuuksien paikallistaminen on tarpeen. Raideosuudet alkavat ja päättyvät raide-eristykseen, vaihteisiin tai akselilaskijoihin. Raide-eristysten nimet määräytyvät raidenumeroiden mukaan. Esimerkiksi raideosuuden Er101 päättävä ja raideosuuden Er102 aloittava raide-eristys nimetään Er101\102. Tämä voitaisiin automatisoinnin avulla toteuttaa siten, että ohjelma tutkisi edellistä ja seuraavaa raidenumeroa ja nimeäisi raide-eristeen niiden mukaisesti. Myös raidenumeron sijoittamiseen voitaisiin hyödyntää automaatiota. Raidenumeron paikka ei ole niin tarkka, kunhan se on oikealla raideosuudella. Näin ollen ohjelmaan voisi määrittää makron, joka tunnistaisi raide-eristeen ja asettaisi raidenumeron aina tiettyyn etäisyyteen siitä. Jos kävisi niin, että symboli menee toisen kanssa päällekkäin, voisi kuvan tarkistava suunnittelija siirtää sitä hiukan.

6.4 AutoCadin monen tulostuskuvan automatisointi

Proxion Oy:n sähköratasuunnittelussa tehdään kuvia sähköratapylväiden portaaleista. Kustakin portaalista tulee oma AutoCad-kuva, joka täytyy tulostaa erikseen. Tulostaminen on yllättävän aikaa vievää, koska portaaletta saattaa samassa kuvassa olla yli kymmenen. Tulostettaessa kuvia Autocadissa pitää ensin määrittää tulostusasetukset, jonka jälkeen kuvasta maalataan tulostettava alue. Nämä toimenpiteet tulee toistaa jokaisen tulostettavan kuvan kohdalla. Tähän työhön kuluu turhaa aikaa ja siksi tähän työvaiheeseen olisi hyvä kehittää jonkinlainen automatisoiva ominaisuus, jolloin toistuvalla työllä säästytäisiin. Ennen kuin tämä työvaihe voitaisiin automatisoida, tulisi selvittää, miten automatisointi käytännössä onnistuisi. Voisi olettaa, että tulostusasetukset voisi asettaa automaattisesti tai tehdä makron valitsemaan oikeat asetukset. Haasteellisempaa varmasti tulisi olemaan tulostettavan alueen rajaaminen kuvasta. Yhtenä vaihtoehtona voisi olla myös, että AutoCad avaisi yhden tulostettavan kuvan kerrallaan, määrittäisi tulostusasetukset ja valitsisi tulostettavan kuvan. Tulostuksen jälkeen ohjelma avaisi seuraavan kuvan ja toistaisi toimenpiteen. Tämä kaikki on vain pohdintaa ja ilman selvitystä ei ole täyttä varmuutta, kuinka se onnistuu. Tulostaminen on kuitenkin osoittautunut sen verran aikaa vieväksi operaatioksi, että tämän osa-alueen automatisointiin kannattaa jatkossa perehtyä.

7. Mitä itse opin

Kuten on käynyt ilmi, opinnäytetyössä käsiteltiin paljon ohjelmointia. Aikaisempaa kokemusta tai minkäänlaista taustaa ohjelmoinnista minulla ei entuudestaan ollut. Opinnäytetyötä tehdessä eri ohjelmointimenetelmät kuitenkin tulivat jollakin asteella tutuiksi. Koska automatisointia päädyttiin tekemään VBA-kielellä tuli siihen syvennyttyä enemmän. Pääasiassa koodauksen suoritti sähköratasuunnittelija Jyri Taivainen, mutta itsekin tein joitain lyhyitä makroja Jyri Taivaisen opastamana. Lisäksi erilaiset rakenteet kuten silmukat, if-lauseet yms. tulivat tutuiksi ohjelmiin syventyessä. Opinnäytetyön aikana opin kirjoittamaan yksinkertaisia makroja VBA-ohjelmointikieltä käyttäen. Lisäksi opinnäytetyön aikana sai käsityksen siitä, mitä ohjelmoinnin avulla ylipäätään voitaisiin toteuttaa ja mitä ei.

Eri ohjelmointikielien lisäksi opin paljon uutta myös itse suunnitteluohjelmista. AutoCad oli jo opintojen ajalta entuudestaan tuttu. Kuitenkin ohjelmaa tutkiessa ja sen eri ominaisuuksia selvittäessä oppi siitäkin paljon uutta. Erilaisiin toimintoihin ja työkaluihin tutustuessa oppi paremmin ymmärtämään, miten niitä voidaan hyödyntää työskentelyssä. CADS Planner-ohjelma ei ollut entuudestaan tuttu, joten siihen täytyi tutustua. Pääpiirteittäin CADS Planner muistuttaa AutoCadia, mutta eroavaisuuksiakin löytyy. CADS Planner oli kuitenkin suhteellisen kivutonta päästä sisään AutoCad-tuntemuksen ansiosta. Kuten jo aikaisemmin on mainittu CADS planner on suomenkielinen ohjelma, joten se myöskin helpotti ohjelmaan tutustumista. AutoCad- ja CADS planner-ohjelmien osalta uudet opitut asiat eivät niinkään liittyneet graafiseen suunnitteluun tai ohjelman piirtotyökaluihin, vaan pääasiassa ne olivat ohjelmien eri toimintoja ja mekanismeja. Esimerkkinä mainittakoon AutoCadissa oleva ominaisuus, jolla Exceliin tehty taulukko voidaan siirtää AutoCadiin ja numeroarvoja muuttaa Excelistä niin että ne päivittyvät AutoCadiin. Tämä työkalu toimii ainoastaan taulukoiden osalta eikä kuvaan graafiseen puoleen päästä tällä työkalulla käsiksi. CADS Planner oli sinänsä jo uusi kokemus, koska en ollut ohjelmaa käyttänyt, mutta jotkut työkalut jälkeinpäin tarkasteltuna osoittautuivat samanlaisiksi myös AutoCadin puolella, joten molempien ohjelmien käyttäminen ristiin avasi jollain tasolla myös toisen ohjelman työkaluja ja sen ominaisuuksia.

Myös itse suunnittelijoiden työ tuli tutuksi lopputyön aikana. Lopputyön aikana tuli tutustuttua sähkörata- ja turvalaitesuunnittelijoiden työhön. Työn aikana sai ainakin pintapuolisen käsityksen siitä, mitä kaikkea suunnittelijoiden tulee ottaa huomioon työtä tehdessään ja miten suunnitteluprosessi etenee. Monet asiat vaikuttavat toisiinsa, esimerkiksi mainittakoon baliisit jotka ovat ns. vastaanottimia, jotka tunnistavat jos juna ajaa niiden yli ja näin vaikuttavat edessä olevaan opastimeen. Baliisin etäisyys opastimesta tulee aina olla 10 metriä, joten teimme ohjelmaan automatisoinnin, joka asettaa baliisin aina 10 metriä ennen opastinta. Tämä menetelmä jätettiin kuitenkin pois järjestel-

mästä, ainakin toistaiseksi. Kuitenkin, koska lopputyö ei suoraan koskettanut itse suunnittelua, jäi varmasti paljon myös oppimatta itse suunnittelutyöstä.

8. Yhteenveto

Opinnäytetyön alkuperäisenä tavoitteena oli selvittää automatisointimenetelmiä Proxion Oy:n sähkörata- ja turvalaitesuunnittelijoiden työn helpottamiseksi. Sähköratasuunnittelussa käytetään AutoCad- ja turvalaitesuunnittelussa CADS Planner-suunnitteluohjelmaa. Automatisointia tulisi hyödyntää turvalaitteiden sekä sähköratapylväiden paikalleen asettamisessa. Työn aikana tuli myös selvittää, onko mahdollista saada kyseiset suunnitteluohjelmat ja Excel-tilukkolaskentaohjelma päivittämään tietoja molempiin suuntiin. Mikäli tällainen yhteys on mahdollinen, siitä tulisi tehdä esimerkkiohjelma. Automatisoinnin avulla pyritään helpottamaan ja nopeuttamaan suunnittelijoiden työtä. Lisäksi automatisoinnin avulla pyritään vähentämään virheiden mahdollisuutta.

Vaikka alun perin opinnäytetyö oli rajattu vain selvittämiseen, alettiin myös itse automatisointia toteuttaa opinnäytetyön aikana. Automatisointi päädyttiin tekemään VBA-ohjelmointikielen avulla. Valintaan vaikutti mm. kielen helppous, hyvät mahdollisuudet löytää tietoa päivittämistä varten, sekä mahdollisuus käyttää tätä kieltä AutoCad-ohjelmassa. Myös jo olemassa olevat Excel-pohjat, joihin automatisointia alettiin toteuttaa, oli tehty VBA-kieltä käyttäen. Pohjana automatisoinnille käytettiin sähköratasuunnittelussa ollutta Excel-pohjaa, jota muokkaamalla ja jatkamalla saataisiin halutunlainen kokonaisuus. Työn aikana ilmeni, että CADS Planner-suunnitteluohjelman ja Excel-tilukkolaskentaohjelman välinen molempiin suuntiin toimiva yhteys tulisi olemaan haastava tehtävä toteuttaa. Tämän ja muutaman muun seikan takia CADS Planner-ohjelman osalta kaksisuuntainen yhteys jätettiin vielä toistaiseksi toteuttamatta. AutoCadin ja Excel-tilukkolaskentaohjelman osalta yhteys voitiin toteuttaa ja esimerkkiohjelma, joka havainnollistaa yhteyden toimintaperiaatteen saatiin opinnäytetyön aikana valmiiksi. Tätä ohjelmaa ei kuitenkaan vielä opinnäytetyön aikana lisätty sähköratasuunnittelussa käytettyyn Excel-pohjaan, koska ei ollut vielä täysin varmaa minkälaiseksi lopullinen työkalu haluttaisiin. Saattaa myös olla, että turvalaite- ja sähköratasuunnittelussa tultaisiin käyttämään tulevaisuudessa samaa Excel-pohjaa jossa olisi molempiin ohjelmiin automatisoivat makrot lisättynä.

Turvalaitteiden ja sähköratapylväiden automaattinen paikalleen asettaminen toteutettiin skriptien avulla. Exceliin kirjoitetut makrot määrittävät symbolien ja blokkien paikat ja kulmat ja lopuksi kirjoittavat skriptitiedoston, joka voidaan ajaa suunnitteluohjelmista. Suunnitteluohjelma tulkitsee skriptin ja toteuttaa siinä annetut käskyt ja toimenpiteet. Näin symbolit ja blokit saadaan kuvaan ilman manuaalista lisäämistä.

Alkuun opinnäytetyö tuntui haastavalta, koska automatisointi tulisi vaatimaan paljon ohjelmointiosaamista, josta minulla ei ollut aikaisempaa kokemusta. Tämän takia työ olikin alun perin rajattu käsittämään pelkästään automatisoinnin selvittäminen. Kuitenkin työn edetessä automatisointi myös toteutettiin blokien ja symbolien paikalleen asettamisen osalta. Ohjelmointiosuudet toteutettiin lehtori Markku Halttusen ja sähköratasuunnittelija Jyri Taivaisen kanssa yhteistyönä. Jyri Taivaisella oli automatisoinnista visio skriptien osalta, joka tässä työssä toteutettiin. Vaikka opinnäytetyö oli haasteellinen, oli se myös opettavainen ja mielenkiintoinen. Mielestäni oli hyvä, että automatisointia myös toteutettiin opinnäytetyön aikana. Tämä antoi työhön konkreettista tekemistä, ja tavallaan ”oikeita” tuloksia esitettäväksi. Jatkossa mahdollisia kehittämiskohteita tulee varmasti esiin ohjelman käytön tuoman kokemuksen pohjalta. Kaikkea ei ole pystytty ottamaan huomioon ohjelmaa tehtäessä ja kehitettävää tai korjattavaa ilmenee. Kappaleessa viisi esitettyjen kehitysehdotusten lisäksi voi olla, että tulevaisuudessa sähkörata- ja turvalaitesuunnittelussa tullaan käyttämään samaa Excel-pohjaa, johon on lisätty molempia ohjelmia koskevat makrot. Lisäksi AutoCad- ja Excel-ohjelmien välinen yhteys tullaan lisäämään sähköratasuunnittelussa käytettyyn Excel-pohjaan. Mikäli molempien suunnittelijoiden ohjelmat löytyisivät samasta Excel-pohjasta, vähentäisi tämä erilaisten Excel-pohjien tarvetta ja yksinkertaistaisi toimintatapoja.

LÄHTEET

Opinnäytetyö Juho Anttila (Viitattu 2.2.2012)

https://publications.theseus.fi/bitstream/handle/10024/14328/Anttila_Juho.pdf?sequence=1

<http://www.proxion.fi/fi/home/> (Viitattu 2.2.2012)

http://www2.liikennevirasto.fi/julkaisut/pdf4/rato_1_yleiset_perusteet.pdf (Viitattu 3.2.2012)

Kymdata Oy. K-ohjelmointikieli 1998

<http://www.cads.fi/fi/Tuotteet/Muut%20toimialat%20ja%20ohjelmistot/CADS%20Planner/> (Viitattu 13.2.2012)

<http://usa.autodesk.com/adsk/servlet/index?siteID=123112&id=770215> (Viitattu 20.2.2012)

<http://msdn.microsoft.com/en-us/library/kehz1dz1%28vs.71%29.aspx> (Viitattu 23.2.2012)

<http://fi.wikipedia.org/wiki/ASCII> (Viitattu 6.3.2012)

http://fi.wikipedia.org/wiki/C_sharp (Viitattu 6.3.2012)

http://fi.wikipedia.org/wiki/Microsoft_Excel (Viitattu 23.4.2012)

http://en.wikipedia.org/wiki/Microsoft_Excel (Viitattu 23.4.2012)

<http://www.lee-mac.com/centreline.html> (Viitattu 25.4.2012)

<http://www.visiblevisual.com/index.php/AutoCad-VBA/edit-dynamic-block-settings.html>
(Viitattu 25.4.2012)

<http://howtoautocad.com/excel-and-autocad-%E2%80%93-a-match-made-in-heaven-again/>
(Viitattu 25.4.2012)