

Paananen Rauli

**Suunnittelujärjestelmien vertailu ja
kehitysmahdollisuudet lasijulkisivurakentamisessa**

Opinnäytetyö

Kevät 2013

Tekniikan yksikkö

Rakennustekniikan koulutusohjelma



SEINÄJOEN AMMATTIKORKEAKOULU

Opinnäytetyön tiivistelmä

Koulutusyksikkö: Tekniikan yksikkö

Koulutusohjelma: Rakennustekniikka

Suuntautumisvaihtoehto: Talonrakennustekniikka

Tekijä: Rauli Paananen

Työn nimi: Suunnittelujärjestelmien vertailu ja kehitysmahdollisuudet lasijulkisivurakentamisessa

Ohjaaja: Martti Perälä

Vuosi: 2013

Sivumäärä: 33

Liitteiden lukumäärä: 1

Insinööritö tehtiin HTR-Steel Oy:lle. Lasin käyttö rakennusmateriaalina on yleistynyt ja monipuolistunut. Lasirakenteiden suunnittelu ja valmistaminen on tämän seurauksena kohdannut uusia haasteita. Kiristyneet vaatimukset erityisesti lämmöneristyksen suhteen ovat johtaneet profiilijärjestelmien voimakkaaseen tuotekehitykseen. Tämä on tuonut samalla uusia vaatimuksia julkisivurakenteita valmistaville yrityksille.

Työn lähtökohta oli etsiä vaihtoehtoisia suunnittelujärjestelmiä alumiini- ja teräsprofiilirakenteiden suunnitteluun. Tarkoituksena oli löytää keinoja suunnitteluautomaation hyödyntämiseen suunnittelutyön nopeuttamiseksi ja suunnitelmien laadun parantamiseksi.

Työ aloitettiin kartoittamalla erilaisia ohjelmistovaihtoehtoja. Vertailuun otettiin mukaan sekä profiilirakenteiden suunnitteluun räätälöityjä ohjelmistoja, kuten Logikal, että myös yleiskäyttöisiä ohjelmistoja, kuten Tekla. Käytännössä vertailu tapahtui vertaamalla muita ohjelmia yrityksen nykyiseen suunnitteluohjelmaan. Työn aikana alettiin myös tutkia mahdollisuuksia yrityksen nykyisen suunnittelujärjestelmän kehittämiseen.

Insinööritöön tuloksena saatiin tietoa erilaisista lasirakentamiseen tarkoitetuista suunnittelutyökaluista sekä mahdollisuuksista niiden tehostamiseen.

Avainsanat: CADS, rakennusautomaatio, lasiteollisuus

SEINÄJOKI UNIVERSITY OF APPLIED SCIENCES

Thesis abstract

Faculty: School of Technology

Degree programme: Construction Engineering

Specialisation: Building Construction

Author: Rauli Paananen

Title of thesis: The design comparison of systems design and development opportunities for the glass facade of a building

Supervisor: Martti Perälä

Year: 2013

Number of pages: 33

Number of appendices: 1

The thesis was made for HTR-Steel Oy, which is a glass façade structures manufacturing company. The use of glass as a building material has increased and diversified. This has led to new challenges with the design and manufacturer. The increasing demands on properties, especially thermal insulation have resulted in a strong product development.

The starting point for the thesis was to look for alternative design systems for aluminum and steel profile structures. The aim was to find ways of utilization of automation in the design and planning to accelerate the plans to improve quality.

The work began with searching for a variety of software options. For comparison were included in the profile, as well as the structures of the design of tailor-made software, such as Logikal that the general-purpose software such as Tekla.

In practice, the comparison was made by comparing the other software company to the current design of the program. During the work the possibilities of the company's current planning system development were taken under inspection.

The result of the thesis provided information on a variety of design tools intended for glass construction and the possibilities for their improvement.

Keywords: CADs, building automation, glass industry

SISÄLTÖ

Opinnäytetyön tiivistelmä.....	2
Thesis abstract.....	3
SISÄLTÖ.....	4
1 JOHDANTO.....	6
2 LASIRAKENTEISSA KÄYTETYT PROFIIlijÄRJESTELMÄT.....	7
2.1 Alumiiniprofiilit.....	7
2.2 Teräsprofiilit.....	8
3 OHJELMISTOT.....	10
3.1 AutoCad.....	10
3.2 CADS Planner.....	10
3.3 Logikal ja Pursocal.....	12
3.4 Tekla Structures.....	13
3.4.1 Yleistä Tekla Structures suunnitteluohjelmasta.....	13
3.4.2 Teklan käyttö alumiinirakenteiden suunnittelussa.....	14
3.4.3 Oven mallinnus komponenttina.....	16
3.4.4 Piirustuksen luominen.....	17
3.4.5 Yhteenveto.....	18
4 CADS PLANNERIN LAAJENNUSMAHDOLLISUUDET.....	19
4.1 Lasin painon laskeva apuohjelma.....	20
4.2 Elementin lämmönläpäisykertoimen laskeva ohjelma Excel-taulukkoon....	21
4.3 Lämmönläpäisykertoimen laskentaohjelma CADS Planneriin.....	24
5 IDEAT JOITA EI TOTEUTETTU.....	28
5.1 Käyttöliittymän toteutus omaan ohjelmaan Cads Plannerissa.....	28
5.2 Profiilisovellus ObjectArx-ohjelmointikielellä.....	29
6 YHTEENVETO.....	31
LÄHTEET.....	33

LIITTEET

Liite 1: Ohjelmakoodi Lasip.k

1 JOHDANTO

Kaupunkirakentamisessa metallilasiseinät ovat olennainen osa arkkitehtuuria. Näitä lasirakenteita ovat esimerkiksi kauppojen näyteikkunat ja kerrostalojen porrashuoneiden lasiseinät. Ihmiset kulkevat päivittäin metallirunkoisten lasiovien kautta kauppoihin, kouluihin ja toimistoihin.

Näiden jokapäiväisten rakenteiden taakse kätkeytyy teollisuudenala, joka on kehittynyt viimeisten vuosikymmenten aikana paljon, mutta pysynyt samalla perusajatukseltaan samanlaisena. Julkisivurakenteiden valmistaminen on yhä käsityövaltaista, vaikka erilaiset automaattikoneet ovatkin osin syrjäyttäneet käsityötä, erityisesti metallin mekaanisessa työstössä. Työ on myös yhä luonteeltaan projektiluontoista. Lasirakenteet ovat eräänlainen rakennuksen käyntikortti, jonka yksilöllisyyteen arkkitehdit kiinnittävät erityistä huomiota. Tämän vuoksi sarjavalmistuksen kaltainen valmistustapa on lasirakenteissa vaikea toteuttaa. Jokainen rakennuskohde on lasirakenteita valmistavalle yritykselle oma, yksilöllinen prosessinsa.

Tämän insinööriyön näkökulma on suunnitteluprosessin tehostamisessa. Väistämättä tämä näkökulma johtaa myös tuotannollisten näkökulmien huomioon ottamiseen. Koska julkisivurakenteet ovat hyvin yksilöllisiä ja usein monimutkaisiakin, jokaisen kohteen suunnittelu on aloitettava puhtaalta pöydältä. Valmiita malliratkaisuja voidaan harvoin sellaisenaan käyttää. Tämä merkitsee sitä, että valmistusprosessissa suunnittelutyö on hyvin merkittävässä asemassa. Ilman yksityiskohtaisia valmistuskuvia on mahdotonta valmistaa mitään. Tuotteen toteutuneet valmistuskustannukset määräytyvät hyvin pitkälti suunnittelijan tekemistä ratkaisuksista ja myös suunnittelun lopputuotteena syntyvän piirustuksen laadukkuudesta.

Vertaillen lasijulkisivurakentamisessa käytettyjä suunnittelujärjestelmiä esimerkiksi teräsrakentamisessa käytettyihin vastaaviin ohjelmistoihin ei voi välttyä ajatukselta, että lasirakentamisen suunnitteluprosessi on käynyt vanhanaikaiseksi. Teräs- ja koneensuunnittelussa tapahtunut merkittävä murros, siirtyminen 2D-ympäristöstä 3D-mallinnukseen, on jäänyt lasirakentamisessa tapahtumatta. Vertailu teräsrakentamisen suunnitteluprosessiin antaa selkeän kuvan siitä, että mahdollisuuksia suunnitteluprosessin tehostamiseen voisi olla runsaasti. Samalla tämä voisi johtaa myös valmistusprosessin tehostumiseen.

2 LASIRAKENTEISSA KÄYTETYT PROFIIlijÄRJESTELMÄT

2.1 Alumiiniprofiilit

HTR-Steel käyttää Purso Oy:n valmistamia profiileja, jotka tunnetaan tuotenimellä Purso. Markkinoilla on myös muiden profiilivalmistajien tuotteita, kuten Nordic Aluminium, SAPA ja Schuco. Toisin kuin esimerkiksi rakenneteräksissä, alumiiniprofiileissa ei ole mitään standardoitua mitoitusjärjestelmää tai muotoilua, vaan eri profiilitoimittajien järjestelmät poikkeavat toisistaan. Ulkonäöllisesti niitä on valmiissa rakenteessa usein vaikea erottaa toisistaan, mutta erilaiset mitoitukset ja liitosratkaisut tekevät niistä valmistuksen ja suunnittelun kannalta täysin erilaisia. Tämä on johtanut siihen, että lasijulkisivuvalmistajat ovat yleensä hyvin sitoutuneita omiin profiilitoimittajiinsa. Profiilivalmistajien valikoimasta löytyy erilaisia profiilijärjestelmiä eri käyttötarkoituksia varten. Esimerkiksi Purson tämänhetkisessä valikoimassa käytetyimmät profiilisarjat ovat P50L ulkoikkunasarja, LK78 lämpökatkaisu ovisarja, P80 palo-ovisarja ja P50 sisäovisarja (Purso 2013). Runkoprofiilien lisäksi profiilisarjaan kuuluvat lasilistat, tiivisteet, liitoskappaleet, liitosruuvit, sarnat ja muut vastaavat osat. Normaaleja rautakauppatuotteita ovat oikeastaan vain eristeet ja pellit.

Alumiinin selkeimpiä etuja teräkseen verrattuna on sen ruostumattomuus. Alumiinin leikkauspintoja ei tarvitse erikseen ruostesuojata, eivätkä maalivauriotkaan aiheuta korroosiovaaraa. Etenkin sateelle alttiissa seinärakenteessa tämä on tärkeä tekijä. Alumiinin pintakäsittelyyn ei myöskään vaikuta se, tuleeko elementti ulkoilmaan vai sisälle. Toinen selkeä etu on alumiinin keveys. Alumiinirakenteiden painosta aiheutuva kuorma ympäröiville rakenteille on melko merkityksetön. Keveyden ansiosta suuretkin rakenteet ovat liikuteltavissa ihmisvoimin, mistä on asennusvaiheessa etua. Alumiiniprofiilien huonona puolena on niiden vaatimaton kuorituksen kestävyys. Koska alumiini on materiaalina kallista, profiileissa on pyritty minimoimaan materiaalin käyttö. Tästä syystä seinämävahvuudet ovat hyvin ohuita. Alumiini on myös materiaalina noin kolmasosan terästä heikompaa. Sen myötölujuutta vastaava $R_{p0,2}$ -venymäraja on 130 N/mm^2 , ja kimmokerroin E on $70\,000 \text{ N/mm}^2$, kun normaalilla rakenneteräksellä vastaavat luvut ovat 355 N/mm^2 ja 210

000 N/mm². Tavallista onkin, että alumiiniprofiileista valmistetuille suurille seinille joudutaan tekemään erilliset tukirakenteet teräksestä. Toinen rajoittava tekijä on alumiinien liitostapa. Koska liitoksia ei hitsata, liitosten käyttö on rajattu siihen, millaisia liitospaloja on saatavissa. Toisaalta liitostapa antaa vapauksia suurten elementtien kuljetuksiin, koska ne voi helposti kuljettaa pienempinä lohkoina ja koota yhteen työmaalla.

Alumiiniprofiilien liitokset tehdään tavallisesti erillisillä liitospaloilla. Liitos voi olla toteutettu puskusaumalla tai jiiriliitoksena. Liitos voidaan tehdä ruuvikiinnitteisenä tai puristamalla. Profiilitoimittajilta saa valmiita liitospaloja ja liitosprofiileja erityyppisiin liitoksiin. Jos valmista liitospalaa ei löydy, se on tehtävä itse. Tämä on kuitenkin melko harvinaista. Liitoksien laadussa sahausjälki ja profiilien mittatarkkuus ovat avainasemassa. Epätarkkuutta liitoksissa on mahdotonta peittää.

Alumiiniprofiileja ei tavallisesti hankita varastoon, vaan kaikki tilataan määrämittäisenä ja valmiiksi maalattuna projektikohtaisesti. Profiilien toimitusaika on yleensä useita viikkoja. Tämän vuoksi suunnittelijalta vaaditaan suurta tarkkuutta profiilien tilauksissa, yhden profiilin unohtaminen voi merkitä koko projektin viivästymistä usealla viikolla. Materiaalitalauksissa pyritään alle 5%:n materiaalihukkaan. Yleensä profiilien tilaus on projektin ensimmäisiä vaiheita.

Alumiinirakenteista ei tavallisesti piirretä erillisiä osakuvia, vaan kaikki sahaukset tehdään kokoonpanokuvan mitoituksen mukaan. Joissakin tapauksissa voidaan tehdä erilliset sahauslistat. Liitoksien vaatimiin työstöihin profiilitoimittajilta on saatavissa erilliset työstöohjeet, joten niitä ei tarvitse sisällyttää valmistuskuviin.

2.2 Teräsprofiilit

Kuten alumiiniprofiileissa, myös teräsprofiileina käytetään eri profiilivalmistajien profiilisarjoja. Nämä ovat poikkeuksetta kylmämuovattuja, ohutseinäisiä profiileja. Saatavissa on lämpökatkaisu- ja lämpökatkaisemattomia profiileja sekä osastoviin rakenteisiin tarkoitettuja palonkestäviä profiileja. Teräsprofiileissa tilanne on lasirakentajan kannalta sikäli helpompi, että valmistukseen ei välttämättä tarvita profiilitoimittajan omia erikoistyökaluja. HTR-Steelin käyttämät profiilijärjestelmät ovat

Forster Unico, Forster Fuego, Forster Presto ja Voest Alpine 60. Myös rakenneteräksiä voidaan käyttää, mutta niistä tehdyt rakenteet tulevat lähes poikkeuksetta kalliimmiksi.

Teräsprofiilien liitokset tehdään tavallisesti hitsaamalla ja työstöt voidaan tehdä yleismallisella työstökoneella tai käsityökaluilla. Lämpökatkaistut profiilit ovat tavallisesti sinkittyjä ja lämpökatkaisemattomat profiilit ovat pintakäsittlemättömiä tai korroosionestopohjamaalattuja. Työstöjen ja hitsauksen jälkeen elementit maalataan lopulliseen väriin. Teräsprofiileista on helpompi tehdä vakioratkaisuista poikkeavia tuotteita kuin alumiinista. Koska teräsprofiilit hitsataan toisiinsa, muodoillaan erilaiset profiilit voidaan liittää helposti toisiinsa. Myös vinot rakenteet on helpompi toteuttaa teräksestä kuin alumiinista.

Suunnittelijan kannalta teräsprofiilit ovat hieman helpompia kuin alumiiniprofiilit. Profiileja voi tarvittaessa jatkaa hitsaamalla ja ylijääneitä profiileja voi käyttää muissa projekteissa.

3 OHJELMISTOT

3.1 AutoCad

AutoCad on AutoDesk Inc. -yhtiön kehittämä 2D/3D-pohjainen, yleiskäyttöinen suunnitteluohjelmisto. Ohjelmisto esiteltiin 1982 ja siitä tuli nopeasti johtava CAD-ohjelmisto (Wikipedia, [viitattu 16.04.2013]). Vaikka tietomallipohjaiset mallinnusohjelmat ovatkin syöneet AutoCadin asemaa, on sen asema yhä kiistaton. AutoCadin merkitystä kuvastaa hyvin se, että sen tiedostformaatti ”dwg” on yhä de facto -standardin asemassa 2D-suunnittelussa. Merkittävä tekijä AutoCadin suosiolle löytyy sen laajennettavuudesta. Ohjelmassa on useita ohjelmointirajapintoja, kuten Lisp ja ObjectArx. Näiden rajapintojen avulla AutoCadiin on voinut lisätä toimialakohtaisia sovelluskokonaisuuksia. Näitä lisäsovelluksia ovat pääsääntöisesti tehneet kolmannen osapuolen ohjelmistoyritykset, kuten esimerkiksi Progman Oy, jonka kehittämä MagiCad on merkittävä talotekniikan ohjelmisto. Lasijulkisivurakentamiseen suunnattuja, AutoCadin päällä toimivia ohjelmistoja ovat esimerkiksi Syscad ja Athena. Molemmat ovat Suomessa melko tuntemattomia eivätkä ne sisällä täällä tavallisesti käytettyjä profiilisarjoja, joten niitä ei tutkittu tarkemmin tämän opinnäytetyön aikana. AutoCadilla voi piirtää lasirakenteiden kuvia ilman erillistä sovellustakin, mutta näin tehtynä suunnittelutyö on liian hidasta.

3.2 CADS Planner

CADS Planner on suomalaisen Kyndata Oy:n kehittämä tekninen piirustusohjelma. Ohjelma muistuttaa toiminnaltaan ja ulkoasultaan melko paljon AutoCadia. Perusohjelmiston lisäksi Kyndata kehittää toimialakohtaisia laajennuksia moneen tarkoitukseen, kuten sähkösuunnitteluun tarkoitettu CADS Electric ja rakennesuunnitteluun tarkoitettu Talon CADS. Profiilirakenteiden suunnitteluun on tarjolla sovellus Profiili CADS (Kyndata 2013). Ohjelma sisältää valmiit työkalut profiilirakenteiden suunnitteluun yleisimmillä Suomessa käytetyillä profiilijärjestelmillä. Myös profiilijärjestelmien lisääminen itse on mahdollista. Profiilien piirtämiseen on sovelluksessa monipuoliset työkalut. Yksittäisen profiilin lisäksi voi määritellä kokonaisen

profiilipaketin, kuten esimerkiksi Purson P50L-sarjassa on tarpeen. Profiilipaketti sisältää määrityksien mukaan rungon, lämpökatkon, lasilistan ja peitelistan. Nämä voi profiilipaketin määrittelyn jälkeen piirtää yhdellä kertaa. Rakenteesta voi myös ottaa leikkauskuvannot. CADS piirtää profiilit suljettuna moniviivana, jossa on attribuutteina tiedot profiilisarjasta ja profiilin tyypistä. Koska profiili on moniviiva, sitä voi muokata samoin kuin muitakin piirustusobjekteja, esimerkiksi venyttämällä. Leikkauskuvannon CADS tekee etsimällä symbolikansioista profiilin *tunnus*-attribuutin mukaista symbolia. Tämä symboli sisältää kuvannon profiilin poikkileikkauksesta. Lasi ja lasilista ei sisälly profiilin poikkileikkaussymboliin, vaan niistä on omat symbolinsa.

Profiileista voidaan tehdä katkaisuluettelot ja optimoida vaadittavat salkomäärät tilausta varten. Profiileista tulevat kirjatuksi luetteloihin ainoastaan ne, jotka on piirretty kuvaan. Poikkeuksena tästä on oviprofiilien lasilista, jonka katkaisupituudet ovat lasisymbolissa attribuutteina. Lasi ja umpiosa sijoitetaan kuvaan omalla toiminnollaan. Ensin piirtäjä näyttää profiilirakenteesta aukon, johon lasi sijoitetaan. Sen jälkeen määritellään lasin tunnus ja paksuus. Lasi sijoitetaan kuvaan symbolina, jonka attribuuteista löytyy tiedot lasin koosta, lasilistojen tyypistä ja pituudesta, leikkauskuvansymbolin nimi ja mahdollinen positiotunnus. Laseista voidaan tehdä tilausta varten litterat omalla toiminnollaan.

CADS Plannerin etuna on sen suomenkielisyys ja monipuoliset työkalut. Heikkoutena taas on 2D-ympäristön tuomat rajoitukset. Kolmiulotteisten rakenteiden piirtäminen on vaikeaa eikä ohjelma sisällä minkäänlaista automatiikkaa esimerkiksi teräsrakenteiden osakuvien tuottamiseen. Kaikista aputoiminnoista huolimatta paljon suunnittelutyötä jää suunnittelijan harteille. Profiilien salkomääriä ja lasiluetteloita lukuun ottamatta lähes kaikki materiaalmäärät suunnittelijan on laskettava itse. Myös kaikki osavalmistukseen tarvittavat kuvat on piirrettävä ja valmistusmäärät laskettava itse. CADS Planner ei myöskään sisällä mitään automatiikkaa peruskoonpanojen, kuten esimerkiksi ovien piirtämiseen. Kaikki rakenteet on suunnittelijan piirrettävä alusta lähtien, tai sitten käytettävä jotain vanhaa piirustusta mallipohjana. Ohjelman muokkaaminen on myös melko työlästä. Esimerkiksi uuden profiilin lisäämiseksi on muokattava useita tekstimuotoisia määritystiedostoja, joiden toimintalogiikka on melko vaikeasti hahmotettavaa.

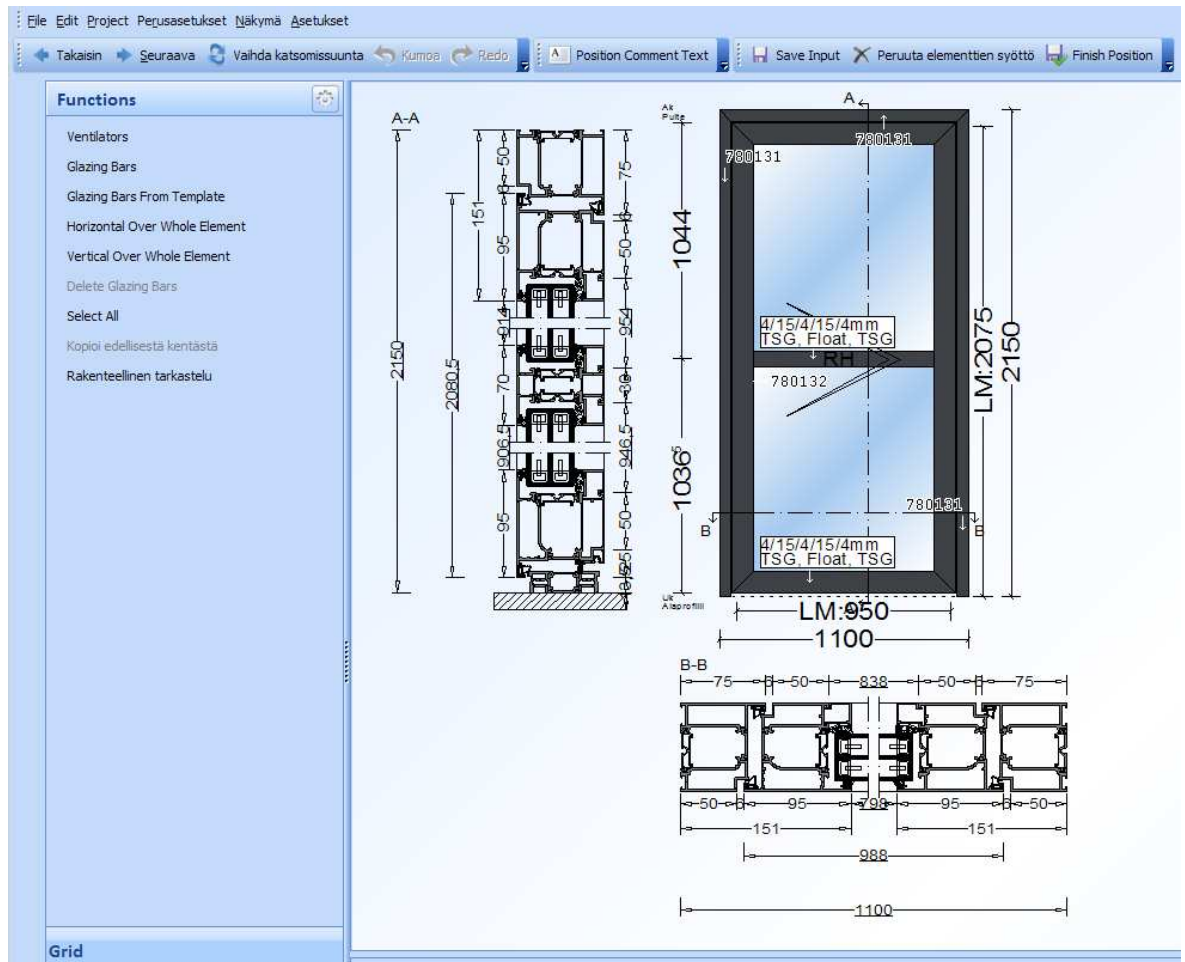
3.3 Logikal ja Pursocal

Logikal on saksalaisen Orgdata-nimisen yhtiön kehittämä profiilijärjestelmien suunnitteluohjelmisto. Jotkin profiilivalmistajat ovat muokanneet ohjelmasta omat versionsa ja toimivat samalla ohjelman jälleenmyyjinä. Suomessa ohjelmaa myy Purso Oy omille asiakkailleen nimellä Pursocal (kuvio 1). Purson toimittama ohjelmisto on melko hyvin suomennettu ja siitä löytyy yleisimmät Purson profiilit. Pursocalin kehitys on melko alkuvaiheessa ja muutoksia ja päivityksiä tulee siksi nopeaan tahtiin. Logikal ei ole perinteinen piirustus- tai mallinnusohjelmisto, sillä sen käyttö ei perustu piirtämiseen. Ohjelman käyttö perustuu siihen, että käyttäjä määrittää ohjelmalle parametrit, joiden mukaan ohjelma muodostaa elementistä mallin. Tästä mallista voidaan ottaa erilaisia raportteja ja listauksia ja myös valmistuspiirustus.

HTR-Steel on käyttänyt on Pursocalia noin vuoden verran. Tämän jakson aikana ohjelman hyödyt ja heikkoudet ovat tulleet hyvin esille. Tavallisen perusoven tai vastaavan elementin mallintaminen on ohjelmalla helppoa. Prosessi etenee dialogi-mallin mukaisesti askel kerrallaan. Ensin määritetään elementin profiilisarja. Sen jälkeen elementtiin tulevat kentät. Esimerkiksi ovi, jossa kiinteä sivupieli, sisältää leveyssuunnassa kaksi kenttää ja korkeussuunnassa yhden. Tämän jälkeen määritetään kenttien mitat, sitten kenttiä kiertävät profiilit. Lopulta tuloksena on valmis elementti. Ohjelmalla voi myös tarkistaa elementtien taipuman ja kuormitukset tuulikuorman ja lasipainon kannalta. Ohjelma ei kuitenkaan laske profiilien jännityksiä, joten ne on laskettava käsin taivutusvastuksen ja suurimman momentin avulla. Ohjelmasta löytyy myös 2D-optimointi, jolla voi tehdä optimoidun leikkausluettelon pelleistä.

Ohjelman muokattavuus on kuitenkin erittäin rajoitettua. Omien profiilien tuominen ohjelmaan on käytännössä mahdotonta. Myös ohjelmaan määriteltyjen profiilien käyttö on mahdollista vain niissä rajoissa, jotka on ohjelmaan valmiiksi koodattu. Usein on kuitenkin tarpeellista käyttää profiileja tavoilla, joihin ei löydy valmiita profiilivalmistajien ohjeita. Tällaisissa tilanteissa Pursocalin mielekäs käyttö ei onnistu, joten valmistuskuvat on tehtävä jollain toisella ohjelmalla. Pursocalin valmistuskuvat on mahdollista viedä esimerkiksi dwg-muotoon muokattavaksi toisella CAD-ohjelmalla, mutta tämä toimii vain yksisuuntaisesti. Toimintamalli, jossa osa työkuvis- ta voidaan tehdä yhdellä ohjelmalla ja loput on tehtävä jollain toisella ohjelmalla,

on melko epäkäytännöllinen. Tästä syystä Pursocalia ei HTR-Steelilla ole otettu käyttöön tuotantokuvien laatimiseen. Hyvin toimivaa 2D-optimointia on käytetty levyoptimointiin. Hinnaltaan Pursocal on melko edullinen CADSiin verrattuna.



Kuvio 1. Pursocalin käyttöliittymä (Pursocal).

3.4 Tekla Structures

3.4.1 Yleistä Tekla Structures suunnitteluohjelmasta

Tekla Structures on suomalainen rakennesuunnitteluohjelmisto. Sen on kehittänyt Tekla Oyj, joka on vuonna 1966 perustettu ohjelmistotalo. Nimi on lyhenne yrityksen alkuperäisestä nimestä Tekninen Laskenta Oy (Tekla 2013). Tekla Structures on noussut maailmanlaajuisesti yhdeksi merkittävimmäksi ohjelmistoksi teräsrakennesuunnittelussa, käyttäjiä on yli 80 maassa. Aiemmin ohjelmisto tunnettiin ni-

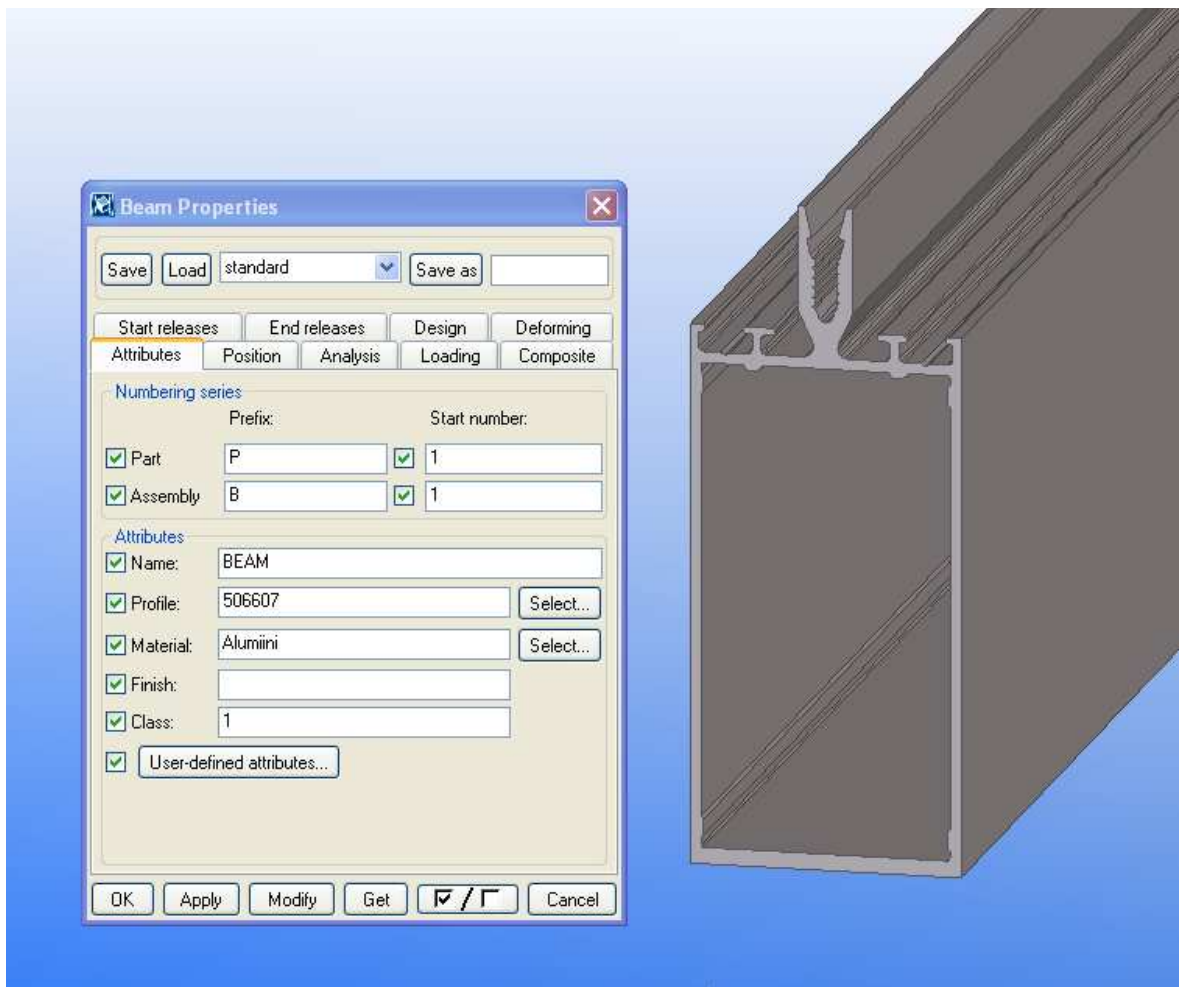
mellä Xsteel. Tekla Structures -ohjelman lisäksi yhtiöllä on muitakin ohjelmistotuotteita, kuten tietomallipohjaisen yhteistyön avuksi kehitetty Tekla BIMsight ja energia- ja infratoimialoille kehitetty sovelluskokonaisuus Tekla Solutions.

Tekla Structures on tietomallinnuspohjainen ohjelmisto, eli rakenteet mallinnetaan kolmiulotteiseksi tilavuusmalliksi. Tästä mallista voidaan ottaa tarvittavat osakuvat, kokoonpanokuvat, määräluettelot ja muut tarvittavat tulosteet. Malli voidaan myös yhdistää muiden osapuolien tuottamien mallien kanssa esimerkiksi IFC-tiedostoformaattia käyttäen (Tekla BIM). Ohjelmalla tuotettuihin valmistuskuviin ei yleensä ole tarvetta piirtää mitään, vaan kaikki tarvittava tieto saadaan mallista. Tämä tekee työskentelytavasta merkittävästi erilaisen verrattuna perinteiseen 2D-piirtämiseen. Mallinnuksessa tärkein työvaihe on saada malli tehtyä oikein, kun taas perinteisessä piirtämisessä päähuomio on piirustuksissa. Mallinnuksen jälkeen piirustukset syntyvät ohjelmalla nopeasti. Osakuvat ohjelma osaa usein tehdä niin hyvin, että suunnittelijan ei ole tarvetta tehdä niihin muutoksia. Kokoonpanokuvaan täytyy yleensä jonkin verran täydentää mitoitusta ja ottaa tarvittavia leikkauskuvia ja osasuurenoksia. Piirustuksiin voi lisätä tietokenttiä, joihin voi kerätä tarvittavaa tietoa mallista. Tietokenttä on ikään kuin suodatin, jonka kautta haluttua tietoa tuodaan mallista piirustukseen. Tällaisia tietokenttiä on esimerkiksi kokoonpanokuvassa oleva *osaluettelo*, jossa voi näkyä esimerkiksi kunkin osan tunnus, profiili, materiaalin laatu ja kappalemäärä. Myös *nimiö* on tällainen tietokenttä, johon tarvittavat tiedot haetaan malliin syötetystä tiedosta. Tietokenttiä käyttäjä voi itse muokata ja lisätä haluamallaan tavalla. Tätä varten Teklasta löytyy sovellus Template Editor. Jos malliin tehdään muutoksia, myös kuvat päivittyvät sen mukaisesti. Tarvittaessa kuva voidaan myös jäädyttää, jolloin se ei päivity automaattisesti.

3.4.2 Teklan käyttö alumiinirakenteiden suunnittelussa

Koska Tekla Structures toimii hyvin teräsrakenteiden suunnittelussa, päätettiin sitä myös testata alumiinirakenteiden suunnitteluun. Ohjelman profiilikirjasto ei sisällä valmiita alumiiniprofiileja, joten ensimmäinen tehtävä oli luoda tarvittavat alumiiniprofiilit siihen. Tekla Structures sisältää profiilien luomiseen editorin, jolla voi tehdä parametrisia profiileja. Koska alumiiniprofiilit ovat tyypillisesti hyvin monimutkaisia

ja koska eri dimensiot poikkeavat usein myös muodoltaan, ei sitä ollut tarkoituksenmukaista käyttää. Sen sijaan profiilit tuotiin dwg-kuvista suoraan. Pohjina käytettiin profiilitoimittajan suunnitteluaineistosta saatuja dwg-kuvia. Ne eivät kuitenkaan olleet valmiita sellaisenaan käytettäväksi. Profiilien ääriviivat piti muokata yhtenäiseksi moniviivaksi ja joitain muotoja yksinkertaistettiin. Dwg-pohjaisen profiilin luomiseen Tekla Structures sisältää komponentin nimeltä DWG profile to library, jolla profiilin poikkileikkaukuvannosta voidaan tuoda profiili ohjelman profiilikirjastoon (Kuvio 2).



Kuvio 2. Purso 506607 profiili tuotu Teklaan dwg-kuvasta (Tekla Structures).

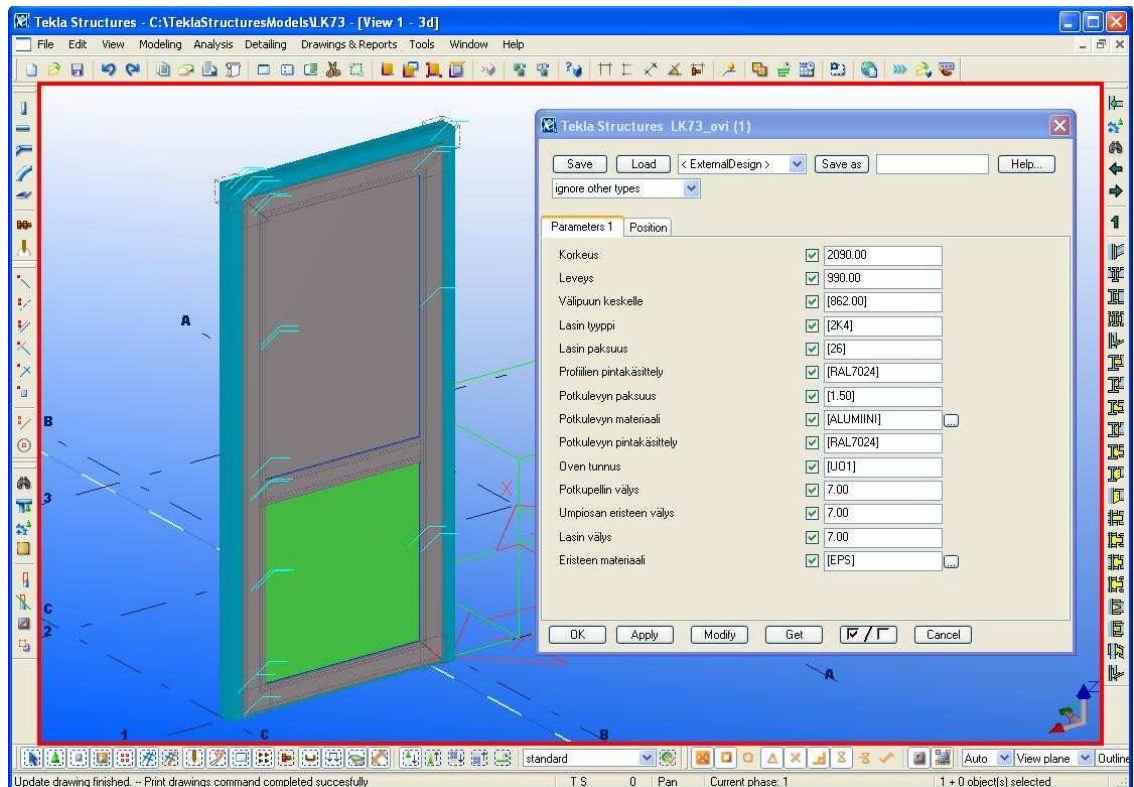
Koska tarkoituksena oli testata Tekla Structures -ohjelman soveltuvuutta tuoteautomaatioon, elementtien mallinnus perinteiseen tapaan profiili kerrallaan ei käynyt kysymykseen. Sen sijaan päätettiin kokeilla elementin luomista komponenttina.

3.4.3 Oven mallinnus komponenttina

Tekla Structures sisältää mittavan valikoiman teräsrakenteisiin tarkoitettuja, valmiita liitosratkaisuja, joita kutsutaan makroiksi. Niiden käyttö säästää suunnittelijan aikaa, sillä detaljeja ei tarvitse mallintaa osa kerrallaan, vaan niissä voi käyttää valmiita ratkaisuja. Tyypillisimpiä makrojen käyttökohteita ovat esimerkiksi pilarien pohjalevyt ja vetosauvojen ja ristikoiden pulttiliitokset. Myös käyttäjä voi luoda omia komponenttejaan. Niitä ohjelmassa kutsutaan nimellä Custom Component. Oman komponentin luominen tapahtuu tyypillisesti niin, että ensin mallinnetaan komponenttiin tulevat osat ja sitten niistä luodaan komponentti. Tätä komponenttia voidaan käyttää muualla mallissa tarvitsematta kopioida tai mallintaa kaikkia osia erikseen. Jos komponenttia muokataan myöhemmin, päivittyvät kaikki mallissa olevat komponentit sen mukaan. Aluksi täytyi siis mallintaa komponentin prototyyppi. Päätettiin luoda Purson LK73 -sarjaa oleva ovi, koska juuri ovissa tuoteautomatation hyöty olisi suurin. Prototyyppiksi tehtiin M10x21 kokoinen yksilehtinen ovi, jossa alaosa on umpinainen ja yläosassa on lasi. Kun ovi oli mallinnettu, tehtiin siitä detail-tyyppinen komponentti. Ohjelmassa komponenteilla voi olla erilaisia tyyppisiä riippuen niiden käyttötarkoituksesta.

Komponentin luomisen jälkeen suurin haaste on tehdä siitä parametrinen. Tämä tarkoittaa sitä, että käyttäjä voi muuttaa komponentin ominaisuuksia haluamukseen vaihtamalla parametrien arvoja. Muokattavia ominaisuuksia on oltava ainakin koko, lasin paksuus ja umpiosan korkeus. Tätä varten komponenttiin alettiin lisätä rajoitteita. Rajoitteilla voidaan sitoa profiilien alku- ja loppupisteitä suhteessa toisiin profiileihin tai mallin koordinaatistoon. Tässä tapauksessa määräävimmin arvoina ovat elementin ääriimitat. Se toteutettiin niin, että oikean puoleisen pystykarmin etäisyys komponentin sijoituspisteestä on sama kuin annettu elementin leveysmitta ja yläkarmin etäisyys sijoituspisteestä on sama kuin elementin korkeus. Näin karmiprofiilit ovat oikeassa kohtaa. Tämän jälkeen jatkettiin sitomalla ovilehtiprofiilit suhteessa karmiprofiileihin sekä lasi ja umpiosa suhteessa ovilehteen. Työ osoittautui melko monimutkaiseksi. Tässä tapauksessa rajoitteita on yhteensä lähes 200 kappaletta. Niiden hallitseminen on melko vaikeaa. On vaikeaa päätellä, miten komponentti käyttäytyy erilaisilla parametreilla, joten työ täytyi tehdä melko pitkälti kokeilemalla. Lopulta saatiin kuitenkin aikaan komponentti, joka toimii kuten oli

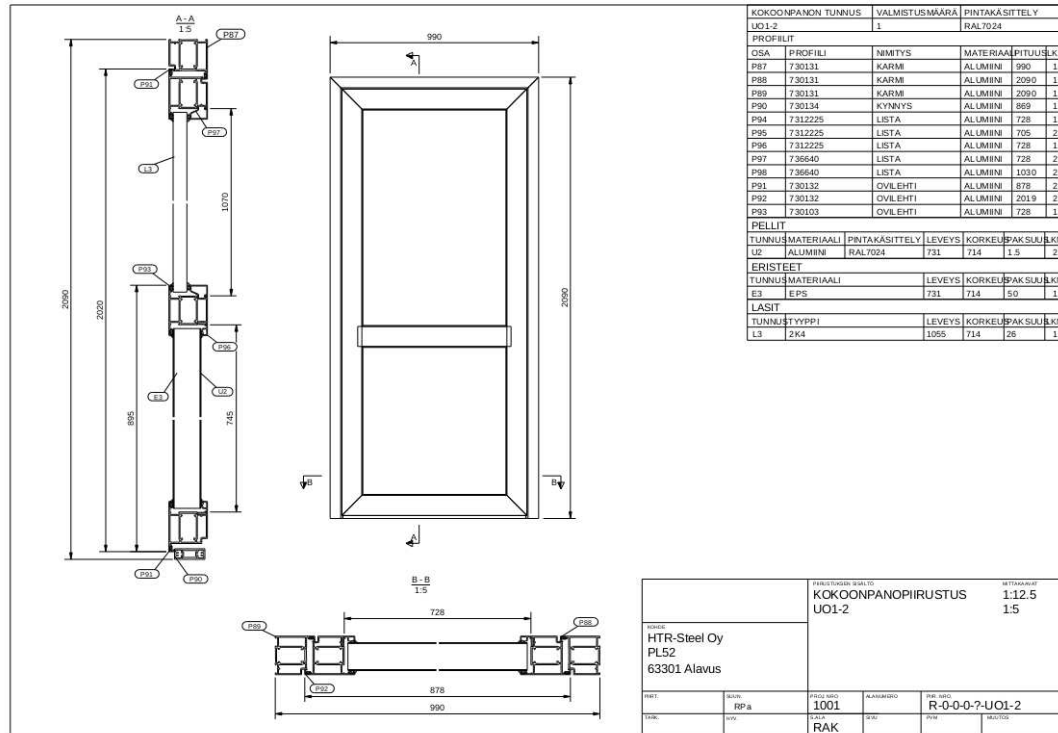
suunniteltu. Komponenttiin tehtiin dialogi-ikkuna, jossa komponentille voidaan antaa halutut arvot (Kuvio 3). Äärimittojen muuttaminen päivittää myös lasin ja umpiosan mitat. Lasipaksuuden vaihtaminen vaihtaa myös lasilistan oikean kokoiseksi.



Kuvio 3. Ovi-komponentti ja sen ominaisuusikkuna (Tekla Structures).

3.4.4 Piirustuksen luominen

Seuraava tehtävä oli luoda ovelle piirustus pohja. Kokoonpanopiirustusta varten luotiin uusi A3-kokoinen piirustus pohja, johon tehtiin Template Editorilla sopivat tietokentät. Lomakkeen ylänurkassa on elementin perustiedot. Sen alla on osaluettelo, lasiluettelo ja umpiosaluettelo. Ovesta riittää tavallisesti naamakuvannon lisäksi yksi vaaka- ja yksi pystyleikkaus. Normaalisti 2D CAD -ohjelmassa leikkauskuvannot ovat yleensä niin pieniä, että pieniä yksityiskohtia ei pysty näkemään. Niinpä työkuviin on yleensä lisättävä luonnollisessa mittakaavassa olevan leikkauskuvannon lisäksi myös leikkauskuvan suurennokset. Tekla Structures ratkaisee tämän ongelman, koska siinä leikkauskuvantoa voi litistää ilman että tämä vaikuttaa mittalukemiin. Näin leikkaus voi olla mittakaavaltaan suurempi kuin oven naamakuvanto (Kuvio 4).



Kuvio 4. Oven kokoonpanopiirustus (Tekla Structures).

3.4.5 Yhteenveto

Tekla Structures on nykyaikainen ja luotettavasti toimiva ohjelmisto. Sen käyttäminen on melko loogista ja yksinkertaista ja sen muokattavuusominaisuudet ovat erinomaiset. Jos tätä opinnäytetyötä jatkettaisiin, seuraava vaihe olisi luultavasti jonkin vaihtoehdoisen automatisointimenetelmän tutkiminen. Tällainen voi olla esimerkiksi .NET-ohjelmointirajapinta. Myös tässä opinnäytetyössä tutkittu Custom Component on käyttökelpoinen tekniikka, mutta komponentin luominen kaikille vaadittaville profiilisarjojen luominen sillä on työläs urakka, koska jokaisella profiilisarjalla työ pitää aloittaa alusta. Ohjelman tutkiminen päätettiin kuitenkin jättää tähän. Suurin syy on ohjelman korkea hankintahinta. Yrityksen suunnittelujärjestelmän uusiminen kaikkine ohjelmistohankintoineen ja koulutuksineen on mittava investointi, johon ei tässä tilanteessa haluttu lähteä.

4 CADS PLANNERIN LAAJENNUSMAHDOLLISUUDET

Työn aikana alettiin myös tutkia CADS Plannerin laajennusmahdollisuuksia. Tämä on sikäli järkevä vaihtoehto, että se on yrityksessä käytössä ja kaikille suunnittelijoille tuttu ohjelma. Varsinaista CADS Plannerin ohjelmointirajapintaa, jota kutsutaan nimellä KDS, Kyndata ei luovuta ulkopuolisille (Suhonen 2012, 20). CADS Planner kuitenkin sisältää oman, K-kieleksi kutsutun BASIC-tyyppisen ohjelmointikielensä. Sillä ei ole mahdollista päästä käsiksi koneen muistiin eikä sisällä kehittyneempien ohjelmointikielien kaltaisia rakenteita, kuten luokkia, perintää tai kapselointia. Sillä ei ole myöskään mahdollista tehdä ohjelmalle graafista käyttöliittymää, vaan kaikki ohjaus on tapahduttava CADS Plannerin komentoriviltä tai määritystiedostoista. Myös tietorakenteet ovat melko alkeelliset, esimerkiksi kaksiulotteiset taulukot ja struktuurit puuttuvat. Yksinkertaisuutensa vuoksi K-kieli on kuitenkin helposti opeteltavissa, vaikka ennestään ei olisikaan ohjelmointikokemusta. Se ei myöskään vaadi erityistä ohjelmointiympäristöä, vaan kääntäminen onnistuu yksinkertaisimmillaan komentoriviltä. Ohjelmakoodin voi kirjoittaa vaikka Microsoft Windowsin Muistiolla. Tarvittava kääntäjä ja otsikkotiedostot tulevat CADS Plannerin asennuspaketin mukana. Tässä työssä käytettiin PSPad tekstieditoria, jolla voidaan tehdä myös ohjelman käännös.

Jo hyvin alkuvaiheessa kävi selväksi, että mitään varsinaista tuoteautomaatiota ei K-kielen avulla ole järkevästi toteutettavissa. Tämä johtuu K-kielen alkeellisuudesta. K-kieli ei tue kooditiedostojen linkitystä, joten kaiken ohjelmakoodin on oltava samassa tiedostossa. Myös tietorakenteiden puutteellisuus monimutkaistaa ohjelmaa, sillä osakokonaisuuksien tietoja on jaettava useaan eri muuttujaan. Tämä aiheuttaa se, että ohjelman kasvaessa riittävän monimutkaiseksi, sen hallittavuus käy mahdottomaksi. Kehittyneemmissä ohjelmointikielissä tämä on ratkaistu jakamalla ohjelmakoodi pienempiin osiin, jolloin kunkin toiminnallisuuden ylläpito on helpompaa. Myös graafisen käyttöliittymän puuttuminen ohjelmakirjastosta on ongelma. K-kieltä voidaan kuitenkin hyödyntää pienempien rutiinien automatisoimiseen. Tässä opinnäytetyössä päätettiin tehdä kaksi ohjelmaa. Toisen tarkoitus on lisätä valmistuskuviin lasielementtien paino ja toisen lisätä kuviin elementin lämmönläpäisykertoimen arvo.

4.1 Lasin painon laskeva apuohjelma

Lasielementin paino on asennuksien kannalta tärkeä tieto. Myös suunnittelijan tarvitsee tietää lasin paino tarkistaakseen vaakaprofiilin kestävyuden. Normaalisti suunnittelijat laskevat lasin painon likiarvon kaavalla:

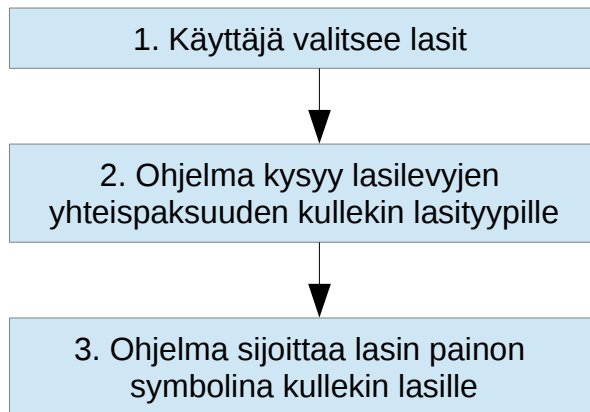
$$W_{\text{lasi}} = A_{\text{lasi}} * T_{\text{lasi}} * 2,5 \quad (1)$$

Jossa W_{lasi} on lasin paino kilogrammoina

A_{lasi} on lasin ala neliömetreinä

T_{lasi} on lasilevyjen yhteispaksuus millimetreinä

CADSin profiilisovellus sijoittaa lasielementin kuvaan symbolina (Kuvio 5). Symbolin prototyyppi on kooltaan 1*1mm, ja sen koko kuvassa määritellään antamalla symbolin kokokertoimiksi lasin leveyttä ja korkeutta vastaavat arvot. Lasin koko kirjataan myös lasisymboliin attribuuttina, joka on muotoa leveys*korkeus. Profiilisovellus lukee lasin koon attribuuttitiedosta, joten samaa logiikkaa käytettiin myös lasipaino-ohjelmassa.



Kuvio 5. Ohjelman toimintakaavio.

Vaikka käyttäjälle ohjelman toiminta näkyy yksinkertaisena, ohjelmaan sisältyy kuitenkin paljon tietojen käsittelyä, ennen kuin ohjelman suoritus on valmis. Valittujen elementtien joukosta pitää poimia lasisymbolit, tallettaa kunkin lasin koko, paksuus ja sijaintikoordinaatti taulukoihin. Kullekin lasille lasketaan sen paino, joka sijoitetaan lasipainosymboliin ja tallennetaan kuvaan. Lisäksi pitää vielä etsiä sen

piirustusraamin mittakaava, jonka sisällä lasi on, jotta lasipainosymboli tulisi kuvaan oikeassa mittakaavassa. Ohjelman asetukset luetaan erillisestä tiedostosta, jotta niitä voitaisiin muuttaa ilman ohjelman uudelleenkäntämistä. Ohjelmakoodia tuli yhteensä 390 riviä (Kuvio 6).

```

/*****
Sijoitetaan symboli kuvaan
*****/

FUNCTION AddLabel(): INT;
INT n_Ind, n_Weigth, n_Scale;
STRING*100 str_Help, str_Question, str_Layer;
BEGIN
  n_Ind = 1;
  n_Scale = 0;
  ArrangeGlasses();
  Getvar( "TASO", str_Layer);
  IF( FindLayer() == 0 ) THEN
    Cads("TASO LUO^#"str_Settings_Value[9]^# 1^#);
    Cads("TASO VAIHDA^#"str_Settings_Value[9]^#);
    str_Question = "Anna lasityypin ";
    Strset( str_Question, strlen(str_Question)+1, str_GlassType[n_Ind]);
    Strset( str_Question, strlen(str_Question)+1, " lasien yhteenlaskettu paksuus mm: ");
    Input( str_Question, n_GlassThicness );
    str_Help = str_GlassType[n_Ind] ;
    REPEAT
      IF( Strcmp( str_Help, str_GlassType[n_Ind]) <> 0 ) THEN
        BEGIN
          str_Question = "Anna lasityypin ";
          Strset( str_Question, strlen(str_Question)+1, str_GlassType[n_Ind]);
          Strset( str_Question, strlen(str_Question)+1, " lasien yhteenlaskettu paksuus mm: ");
          Input( str_Question, n_GlassThicness );
        END
      n_Scale = FindScale(r_GlassLDX[n_Ind],r_GlassLDY[n_Ind]);
      IF( n_Scale == 0 ) THEN n_Scale = 20;
      n_Weigth = Round(r_GlassWidth[n_Ind] * r_GlassHeight[n_Ind] * n_GlassThicness * 2.5 * 0.000001);
      Cads("TASO VAIHDA^#"str_Settings_Value[9]^#);
      Cads("SYMBOLI HAE lasip^#"(r_GlassLDX[n_Ind]+Strval(str_Settings_Value[5])*n_Scale),
(r_GlassLDY[n_Ind]+Strval(str_Settings_Value[6])*n_Scale) n_Scale n_Scale 0 n_Weigth^#);
      Cads("TASO VAIHDA^#"str_Layer^#);
      str_Help = str_GlassType[n_Ind] ;
      n_Ind = n_Ind + 1;
    UNTIL(n_Ind > n_NumOfGlass);
    AddLabel = 0;
  END

```

Kuvio 6. Koodin osa, missä lasipainosymboli sijoitetaan kuvaan (K-Kieli).

4.2 Elementin lämmönläpäisykertoimen laskeva ohjelma Excel-taulukkoon

Nykyaikana rakenteiden lämmöneristykseen kiinnitetään koko ajan enemmän huomiota. Tämä koskee myös ovia ja ikkunoita. Metallirakenteiden kohdalla tilanne on hankala, sillä profiilien lämmöneristys on yleensä ollut melko huono. Kiristyneet

määräykset ovat pakottaneet myös profiilivalmistajat kehittämään lämmöneristykseen parempia ratkaisuja. Tästä huolimatta ovirakenteiden kohdalla ollaan usein siinä tilanteessa, että määräysten mukaista lämmönläpäisykerrointa ei voida saavuttaa. Profiilirakenteita valmistavissa yrityksissä lämmöneristysvaatimuksien kiristyminen näkyy myös lasien kehittymisenä. Aikaisempina vuosina yleisesti käytetyt kaksinkertaiset eristyslasit on käytännössä kokonaan korvattu kolminkertaisilla. Kaikki eristyslasit toimitetaan nykyään yhdellä tai kahdella selektiivikalvolla (Lasifakta 2012, 16). Usein myös eristyslasien välilistaksi on valittava RST- tai komposiittirakenteinen välilista, joilla saavutetaan parempi lämmöneristävyys kuin perinteisellä alumiinirakenteisella välilistalla (Lasifakta 2012, 74). Suunnittelijan työssä lämmöneristysvaatimuksien kiristyminen näkyy myös siten, että yhä useammin on piirustuksien lisäksi esitettävä tilaajalle laskelmat elementtien lämmöneristävydestä. Tästä syystä HTR-Steel on päättänyt, että lämmöneristävyden arvo kirjataan valmistuskuviin. Lämmönläpäisykerroimen laskeminen on kuitenkin siinä määrin työlästä, että päätettiin kehittää suunnittelujärjestelmään apuväline sen laskemista varten. Lämmönläpäisykerroin lasketaan RakMK C4 mukaan kaavalla 2:

$$U_{cw} = \frac{A_g * U_g + A_p * U_p + A_f * U_f + I_g * \Psi_g}{A_{cv}} \quad (2)$$

Missä U_{cw} on Elementin keskimääräinen lämmönläpäisykerroin W/m^2K

A_g on Lasin pinta-ala m^2

U_g on Lasin lämmönläpäisykerroin W/m^2K

A_p on Umpiosan pinta-ala m^2

U_p on Umpiosan lämmönläpäisykerroin W/m^2K

U_f on Karmin yhteenlaskettu pinta-ala m^2

A_f on Karmin lämmönläpäisykerroin W/m^2K

I_g on Lasipiiri m

Ψ_g on Viivamainen lisäkonduktanssi W/mK

A_{cv} on Elementin kokonaispinta-ala m^2

Aluksi lämmönläpäisykertoimen laskuohjelma tehtiin Excel-taulukkoon. Ohjelma sisältää kaksi välilehteä, joista toisella on varsinainen laskentalomake ja toisella ohjelman taulukoidut lähtötiedot, kuten eri profiilisarjojen lämmöneristävyyksien mitoitusarvot. Laskentalomakkeessa käyttäjä voi valita alasettelusta käytettävän profiilisarjan, välilistan tyyppin, eristemateriaalin ja lasin lämmönläpäisykertoimen. Muut numeeriset arvot syötetään niille varattuihin soluihin (Kuvio 7). Ohjelma osoittautui toimivaksi ja käyttökelpoiseksi, mutta päivittäiseen työhön se on turhan hidas käyttää. Käyttäjän täytyy ensin itse laskea pinta-alat ja lasipiiri, mikä kuluttaa työaikaa.

U-arvolaskuri Ver. 0.9

Profiilit ja sisäosat

Profiilisarja: Purso LK78H

Lasipaketin välilista: Al

Umpiosan eristemateriaali: EPS

Umpiosan paksuus mm: 50

Lasin U-arvo: 0,60

Lasin paksuus mm: 46

Pinta-alat ja piiri

Elementin kokonaispinta-ala: 2,20 m^2

Lasipinta-ala: 1,22 m^2

Umpiosan pinta-ala: 0,17 m^2

Lasipiiri: 4,80 m

U-arvot ja lisäkonduktanssi

Karmin U-arvo: 1,450 W/m^2K

Lasin U-arvo: 0,600 W/m^2K

Umpiosan U-arvo: 0,820 W/m^2K

Välilistan lisäkonduktanssi: 0,106 W/m^2K

Rakenteen kokonais U-arvo: 1,16 W/m^2K

Kohde: _____

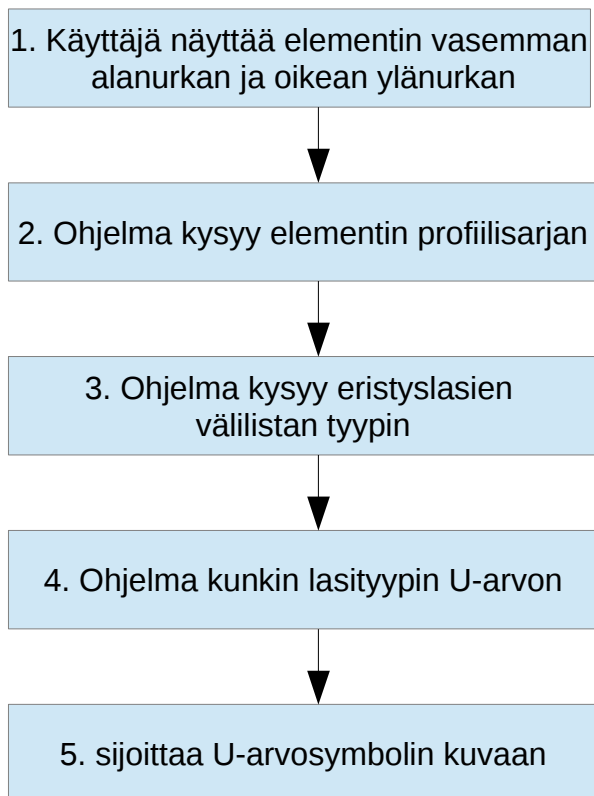
Elementti: _____

Lisätiedot: _____

Kuvio 7. Lämmönläpäisykertoimen laskuohjelma Excel-ohjelmassa (Excel laskuri)

4.3 Lämmönläpäisykertoimen laskentaohjelma CADS Planneriin

Koska Excel-laskentaohjelman käyttö on hidasta, ohjelma päätettiin liittää osaksi CADS Planneria (Kuvio 8). Ohjelman suunnittelu osoittautui melko työlääksi. Aluksi ajatuksena oli laskea elementeistä profiilien juoksumetrimäärät ja laskea sen perusteella karmirakenteen pinta-ala. Tähän ei kuitenkaan löytynyt luotettavaa keinoa, sillä kuvassa saattaa olla piirrettyinä monta profiilia päällekkäin. Lopulta toimintaa päädyttiin yksinkertaistamaan niin, että käyttäjä näyttää elementin kaksi nurkkapistettä, joiden perusteella lasketaan elementin pinta-ala. Tästä alasta vähennetään lasien ja umpiosien yhteenlaskettu pinta-ala, ja erotuksen oletetaan olevan kokonaan karmirakennetta. Lasityyppien lämmönläpäisykertoimien arvot on kysyttävä käyttäjältä, koska kuvassa ei sitä tietoa ole. Samoin myös välilistan tyyppi on kysyttävä käyttäjältä. Eristeen tyyppi ja paksuus saadaan umpiosan symbolista.



Kuvio 8. Ohjelman toimintakaavio

Ohjelmaan liittyy neljä määrittystiedostoa. Uarv.ini sisältää ohjelman yleiset määrittökset. Uarv_eriste.dat sisältää käytettävien eristetyyppien lämmönläpäisykertoimi-

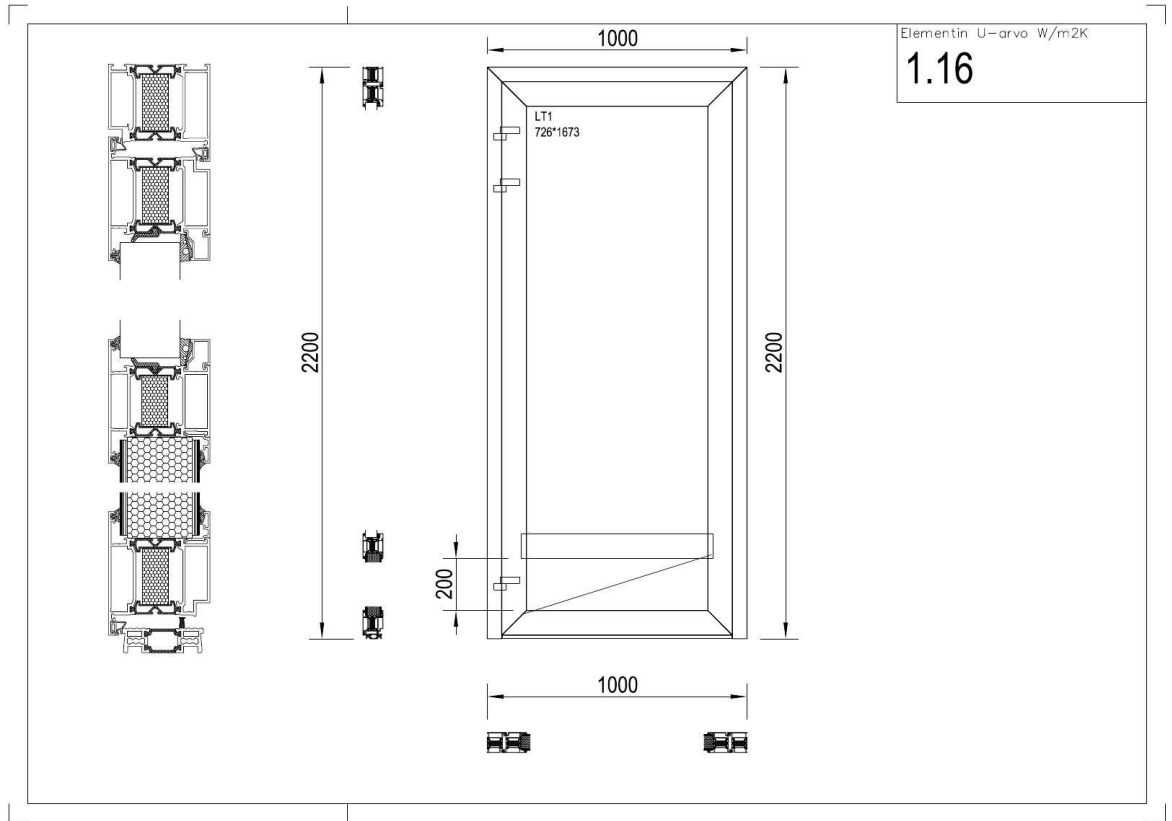
met. Eristeen tyyppi saadaan kuvasta umpiosasymbolista ja sitä verrataan tiedostosta luettuihin eristetyyppeihin. Jos vastaavaa tyyppiä ei löydy, ohjelma keskeytyy virheeseen. Uarv_profiili.dat sisältää profiilisarjojen lämmönläpäisykertoimet. Tiedostossa on myös sarake eri lasipaksuuksille, koska joillakin profiilisarjoilla lasipaksuus vaikuttaa lämmönläpäisykertoimeen. Lasin paksuus luetaan kuvan lasisymbolista. Lisakond_valilista.dat sisältää välilistatyyppien lisäkonduktanssit. Tuloksen ohjelma liittää symboliin, joka sijoitetaan kuvaan käyttäjän näyttämään paikkaan piirustusraamin mukaisessa mittakaavassa (Kuvio 10). Ohjelmakoodia kertyi kaikkiaan 990 riviä (Kuvio 9). Ohjelman toiminta oli helppo tarkistaa vertaamalla tuloksia Purso Oy:n julkaisemiin laskelmiin ovirakenteista (Kuvio 11). Jos ohjelmaa kehitettäisiin edelleen, olisi seuraava vaihe tulosten kerääminen tiedostoon, josta ne olisi helposti tuotavissa Excel-taulukkoon tulosten kirjallista esittämistä varten.

```

FUNCTION CalculateUValue(): REAL; //Lasketaan elementin U-arvo
REAL r_Profile_UValue, r_UValue, r_UValueGTotal, r_UValuePTotal;
INT n_Ind, n_Ind2, n_Help, n_Find;
STRING*100 str_Layer, str_Help, str_Question;
BEGIN
  r_UValueGTotal = 0;
  r_UValuePTotal = 0;
  n_Help = Questions();
  r_Profile_UValue = FindProfileUValue();
  n_Ind = 1;
  str_Help = str_GlassTypes[n_Ind] ;
  WHILE( n_Ind <= n_NumOfGlassTypes ) DO
  BEGIN
    str_Question = "Anna lasityypin ";
    Strset( str_Question, strlen(str_Question)+1, str_GlassTypes[n_Ind]);
    Strset( str_Question, strlen(str_Question)+1, " U-arvo: ");
    Input( str_Question, r_UValue );
    r_UValueGTotal = r_UValueGTotal + r_GlassArea[n_Ind] * r_UValue;;
    str_Help = str_GlassTypes[n_Ind] ;
    n_Ind = n_Ind + 1;
  END
  n_Ind = 1;
  n_Ind2 = 1;
  REPEAT //Umpiosien U-arvot
    n_Ind2 = 1;
    n_Find = 0;
    REPEAT
      IF( Stricmp( str_Insulate_type[n_Ind2], str_PanelTypes[n_Ind] ) == 0 ) THEN
      BEGIN
        r_UValuePTotal = r_UValuePTotal + ( r_PanelArea[n_Ind] * ( r_InsulateUValue[n_Ind2] /
          (r_PanelThickness[n_Ind] * 0.001 )));
        n_Find = 1;
      END
      n_Ind2 = n_Ind2 + 1;
    UNTIL( n_Ind2 > n_NumOfInsulateUValues );
    IF( n_Find == 0 & n_NumOfPanelTypes <> 0 ) THEN
    BEGIN
      Print("Virhe! Eristetyyppiä ",str_PanelTypes[n_Ind], " ei löyty tiedostosta ",
        STR_UVALUESFILE_INSULATE, "\n");
      Stop;
    END
    n_Ind = n_Ind + 1;
  UNTIL(n_Ind > n_NumOfPanelTypes);
  CalculateUValue = ( (r_TotalArea - ( r_GlassTotalArea + r_PanelTotalArea )) * r_Profile_UValue +
  r_UValueGTotal + r_UValuePTotal + r_GlassPerimeter * r_GlassCond) / r_TotalArea;
  Print("\n Kokonais pinta-ala on: ", r_TotalArea);
  Print("\n Lasin kokonais pinta-ala on: ", r_GlassTotalArea);
  Print("\n Umpiosan kokonais pinta-ala on: ", r_PanelTotalArea);
  Print("\n Lasipiiri on: ", r_GlassPerimeter);
END

```

Kuvio 9. Ohjelmakoodin osa, jossa lasketaan keskimääräinen lämmönläpäisykerroin (K-Kieli).

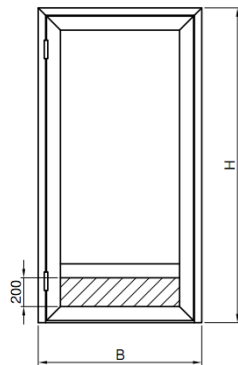


Kuvio 10. Ohjelmalla laskettu lämmönläpäisykerroin (CADS Planner).



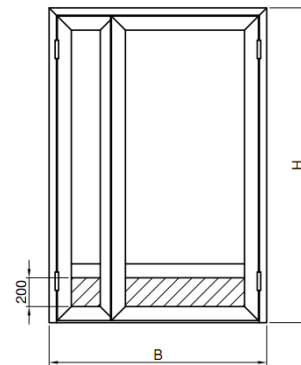
LK 78H 21.1

LK78H OVIEN U_D -ARVO ESIMERKKEJÄ



1-lehtinen ovi

Eristyslasin välilista		Alumiini 0,3			RST 0,18			TPS		
Lasin Ug arvo (W/m²K)		0,5	0,6	0,7	0,5	0,6	0,7	0,5	0,6	0,7
B	H	Oven U_D -arvo (W/m²K)								
900	2200	1,1	1,2	1,2	1,1	1,1	1,2	1,0	1,1	1,1
1000	2200	1,1	1,2	1,2	1,0	1,1	1,1	0,97	1,0	1,1
1100	2200	1,1	1,1	1,2	0,99	1,0	1,1	0,95	1,0	1,1
1200	2200	1,0	1,1	1,2	0,97	1,0	1,1	0,93	0,98	1,0



2-lehtinen ovi

Eristyslasin välilista		Alumiini 0,3			RST 0,18			TPS		
Lasin Ug arvo (W/m²K)		0,5	0,6	0,7	0,5	0,6	0,7	0,5	0,6	0,7
B	H	Oven U_D -arvo (W/m²K)								
1400	2200	1,2	1,2	1,3	1,1	1,1	1,2	1,0	1,1	1,1
1600	2200	1,1	1,2	1,2	1,0	1,1	1,1	0,99	1,0	1,1
1800	2200	1,1	1,2	1,2	1,0	1,1	1,1	0,95	1,0	1,1
2000	2200	1,1	1,1	1,2	0,97	1,0	1,1	0,93	0,98	1,0
2200	2200	1,0	1,1	1,2	0,95	1,0	1,1	0,90	0,96	1,0
2400	2200	1,0	1,1	1,1	0,93	1,0	1,1	0,89	0,95	1,0

Kuvio 11. Ovien lämmönläpäisykertoimia Purson esitteestä (Purso LK78H)

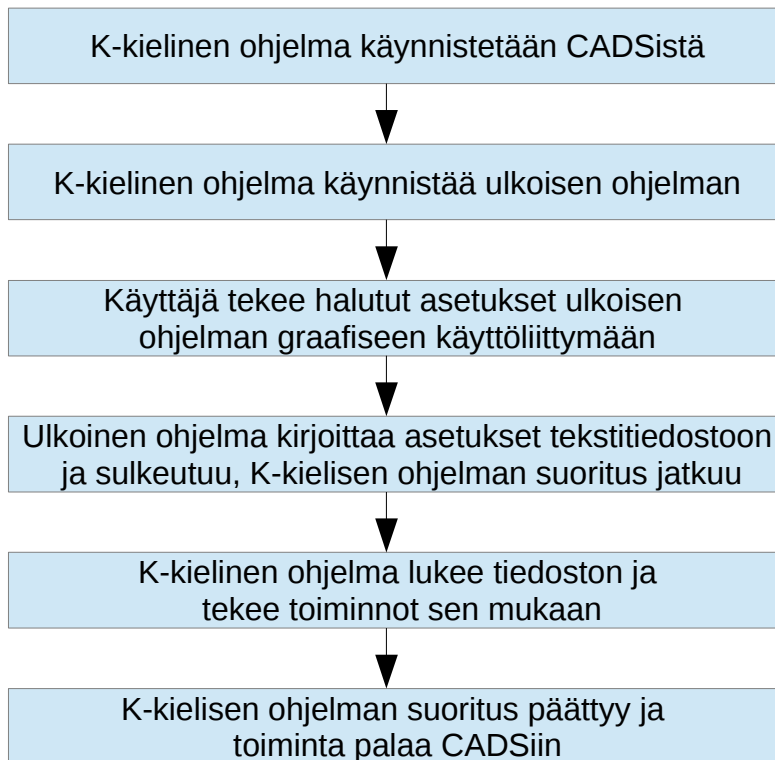
5 IDEAT JOITA EI TOTEUTETTU

5.1 Käyttöliittymän toteutus omaan ohjelmaan Cads Plannerissa

Työn aikana syntyi kaksi ideaa, joita ei kuitenkaan lähdetty toteuttamaan. Ensimmäinen niistä koskee käyttöliittymän toteuttamista CADs Plannerin K-kielillä toteutettuun ohjelmaan. Kuten edellä on todettu, CADs Planner ei anna suoraan mahdollisuutta tehdä graafista käyttöliittymää käyttäjän omille ohjelmille. CADs Planner kuitenkin sisältää mahdollisuuden suorittaa ulkoisen ohjelman komentoriviltä komennolla "suorita". Tätä kautta ulkoinen ohjelma on siten mahdollista suorittaa myös K-kielisestä ohjelmasta. Esimerkiksi Windowsin Muistion voi K-kielisestä ohjelmasta käynnistää komennolla:

```
Cads("SUORITA e e notepad^#");
```

Tämä ulkoinen ohjelma voisi olla myös käyttäjän itse ohjelmoima ohjelma, jonka käyttöliittymän kautta voisi tehdä halutut asetukset. Kun ulkoisesta ohjelmasta poistutaan, kirjoittaa se asetukset tekstimuotoiseen tiedostoon, joka sitten luetaan CADs Plannerin apuohjelmasta ja halutut toiminnot suoritetaan sen mukaan. Toiminnon syntaksi olisi siis muotoa:



Kuvio 12. Apuohjelman syntaksi.

Tällä tavoin olisi mahdollista luoda monipuolisempia apuohjelmia CADS Planneriin. Ulkoisessa ohjelmassa olisi määrittelyt oven profiilisarjalle, ulkomitoille, heloituksille ja muille tarpeellisille tiedoille. Ohjelma kirjoittaisi profiilien solmupisteiden koordinaatit ja muut tarvittavat tiedot tiedostoon, jonka perusteella K-kielinen ohjelma piirtäisi CADS Planneriin kuvan. Ongelmaksi jäisi vielä se, että tämänkaltainen automatisointi toimii vain yhteen suuntaan. Kun kuva on kerran luotu CADS Planneriin, sitä ei enää voi muokata ulkoisen ohjelman kautta, vaan olisi jatkettava piirtämistä perinteisellä tavalla. Ohjelmasta tulisi myös tavattoman monimutkainen. Ottaen huomioon K-kielen rajoitukset, tehtävä voisi osoittautua liian haastavaksi.

5.2 Profiilisovellus ObjectArx-ohjelmointikielellä

Toinen esille tullut vaihtoehto olisi toteuttaa ohjelma AutoCadin päälle esimerkiksi ObjectARX-ohjelmointikielellä. Ohjelman syntaksi olisi yksinkertaisempi kuin edellä kuvatussa CADS Plannerin apuohjelmassa, koska tiedon siirtoa kahden apuohjelman välillä ei tarvittaisi. Myös kaksisuuntaisuus onnistuisi, eli luotua kuvaa pääsisi muokkaamaan apuohjelman kautta. Kaksisuuntaisuudessa voisi hyödyntää Auto-

Cadin uusia ominaisuuksia, kuten parametreja ja dynaamisia blokkeja. Dynaaminen blokki tarkoittaa blokkia, jonka ominaisuuksia voi muuttaa blokin luomisen jälkeen. Blokin mittoja voi esimerkiksi muuttaa tekemällä siihen rajoitteet, joiden arvoja käyttäjä voi muuttaa. Parametrit puolestaan ovat eräänlaisia muistipaikkoja, joihin käyttäjä voi tallettaa tietoa. Kun rajoitteen arvoksi asetetaan jokin parametri, voi käyttäjä muuttaa parametrin arvoa ja samalla dynaaminen blokki päivittyy sen mukaan. Tässä tapauksessa ohjelma muuttaisi parametrien arvoja ja kuva päivittyisi sen mukaan.

Ongelmaksi tässäkin tapauksessa muodostuu se, että ohjelmasta tulisi erittäin laaja ja monimutkainen kokonaisuus. Koska AutoCadissa ei suoraan ole profiilien piirtoon apuvälineitä, kaikki tarvittavat ominaisuudet täytyisi koodata alusta lähtien itse. Myös ObjectARX-ohjelmointikielen opettelu vaatisi aikaa. Kielestä on kyllä hyvin saatavissa esimerkkiohjelmia ja ohjeita, mutta kieli itsessään on monimutkainen ja laaja. Kielen oppimiseksi pitää olla ennestään kokemusta olio-ohjelmoinnista ja graafisesta ohjelmoinnista, esimerkiksi MFC:n tai vastaavan luokkakirjaston kanssa.

6 YHTEENVETO

Opinnäytetyön tavoitteena oli kartoittaa ja tutkia vaihtoehtoisia suunnittelujärjestelmiä HTR-Steelin tarpeisiin. HTR-Steelin nykyinen suunnittelujärjestelmä on CADS Planner, jolla tehdään lähes kaikki suunnittelutyö. Koska CADS Planner on perinteinen 2D Cad -ohjelma, on sen käyttö koettu vanhanaikaiseksi ja työlääksi. Monet samankaltaisina toistuvat rutiinit on sillä pakko tehdä aina erikseen piirtämällä ja samoja tietoja on liitettävä kuviin moneen kertaan. Jos nämä aikaa vievät rutiinit voisi automatisoida, se säästäisi suunnittelijoiden aikaa ja vähentäisi virheitä, sekä tekisi työstä mielekkäämpää.

Selvitystyön aikana todettiin, että käyttökelpoisia vaihtoehtoja ei ole tarjolla kovin montaa. CADS Plannerissa on kuitenkin monia selkeitä vahvuuksia. Yksi etu on se, että siihen on kehitetty valmis profiilikirjasto ja tarvittavat perusohjelmat HTR-Steelin käyttämille profiilijärjestelmille. Vaikka itse piirtäminen sillä onkin jonkin verran työlästä, aikaa ei kuitenkaan mene itse perusohjelmiston ylläpitoon ja kehitykseen. Profiilijärjestelmien muutokset päivittyvät ohjelmaan päivityksinä eikä suunnittelijan tarvitse kuin ladata päivitys ohjelman kotisivulta ja asentaa se. Mikään toinen ohjelmisto ei tällä hetkellä tietävästi tarjoa tätä etua. CADS Planner on myös täysin suomenkielinen ja sillä on suomenkielinen tuotetuki. AutoCadin kaltaisena sen käyttö on helppo omaksua, jos vain suunnittelijalla on ennestään kokemusta AutoCadista. CADS Planneriin voi myös vapaasti lisätä omia profiileja.

Muista vaihtoehdoista Tekla oli selvästi lupaavin. Se tarjoaa monia etuja CADSiin verrattuna. Merkittävin niistä on mahdollisuus automatisoida aikaa vieviä piirustus-rutiineja. Haittapuolena voi nähdä henkilöstön suuren koulutustarpeen. 3D-ohjelmistoihin tottumattoman voi olla vaikeaa sisäistää mallinnusohjelman työtapoja, jotka eroavat melko paljon perinteisestä 2D-piirtämisestä. Merkittävä haittapuoli on myös ohjelmiston hinta, joka huomattavasti suurempi CADS Planneriin verrattuna.

Alun perin opinnäytetyössä oli tarkoitus vain kartoittaa muita tarjolla olevia ohjelmia, mutta työn aikana alettiin myös pohtia CADS Plannerin tarjoamia mahdollisuuksia piirtämistekniikan kehittämiseen. Tämän tuloksena CADS Planneriin kehitettiin käyttökelpoisia apuvälineitä, joilla joitakin piirustusrutiineja voidaan helpottaa. Samalla jouduttiin toteamaan myös, että mitään monimutkaisia laajennuksia

CADS Planneriin ei voi omatoimisesti tehdä. Kyndata Oy:n toiminta-ajatukseen kuuluu se, että yritys kehittää ja markkinoi lisäsovellukset itse, joten siltä näkökannalta tämä on ymmärrettävää.

Tässä yhteydessä on myös syytä ottaa yrityksen johdon näkökulma huomioon. Vaikka yrityksen henkilöstössä olisikin intoa ja taitoa kehittää sovelluksia yrityksen omaan käyttöön, on kuitenkin vaarallista, jos yritys tulee niistä liian riippuvaiseksi. Kolmannen osapuolen tarjoaman ohjelmiston saatavuus ja tuotetuki on melko turvallista. Mutta jos yrityksen suunnittelujärjestelmä on liikaa yrityksen työntekijän erikoistaitojen varassa, voi koitua ongelmaksi, jos tämä ammattitaito syystä tai toisesta yrityksestä katoaa.

Joka tapauksessa on hieman outo tilanne, jos yritys joutuu muokkaamaan ja tekemään itse suunnittelujärjestelmiä tämänkaltaiselle alalle, joka ei kuitenkaan ole mikään pienimuotoinen erikoisala. Tämän opinnäytetyön aikana vahvistui käsitys siitä, että lasirakentamisen suunnittelujärjestelmät ovat vanhanaikaisia verrattuna esimerkiksi koneenrakennuksen tai teräsrakentamisen suunnitteluohjelmistoihin. Ilmeisesti markkinat ovat kuitenkin niin pienet, että ohjelmistotalot eivät ole halunneet tehdä merkittävämpää panostusta. Voisi kuitenkin ajatella, että edulliselle 3D-mallinnusohjelmalle, joka Teklan tapaan käyttää Top-Down-mallinnuslogiikkaa, voisi olla kysyntää. Julkisivurakentamisen lisäksi tällaista ohjelmaa voisivat hyödyntää esimerkiksi puuelementtitehtaat.

LÄHTEET

Cads Planner. Kuvaruudunkaappaus ohjelmasta. Rauli Paananen.

Excel laskuri. Kuvaruudunkaappaus ohjelmasta. Rauli Paananen.

K-Kieli. Kuvaruudunkaappaus ohjelmakoodista. Rauli Paananen.

Kymdata Oy. 2013. [Verkkajulkaisu]. Tuotteet. [Viitattu 19.4.2013]. Saatavana: <http://www.cads.fi/fi/Tuotteet/>

Lasifakta. 2012. Energiansäästölasit. Lahti: Pilkington Oy.

LK78H. 2013. Purso LK78 esite. Siuro: Purso Oy.

Purso Oy. 2013. [Verkkajulkaisu]. Suunnittelumateriaali. [Viitattu 19.4.2013]. Saatavana: http://www.purso.fi/fi/tuotteet_ja_palvelut/rakennusjarjestelmat/suunnittelumateriaali.html

Purso. Kuvaruudunkaappaus ohjelmasta. Rauli Paananen.

Pursocal. Kuvaruudunkaappaus ohjelmasta. Rauli Paananen.

Suhonen, V. 2012. AutoCad- ja CADS Planner- suunnitteluohjelmien automatisoinnin selvittäminen. Kuopio: Savonia Ammattikorkeakoulu.

Tekla Oyj. 2013. [Verkkajulkaisu]. Historia. [Viitattu 22.4.2013]. Saatavana: <http://www.tekla.com/fi/about-us/history/Pages/Default.aspx>

Tekla Oyj. 2013. [Verkkajulkaisu]. Tekla BIM. [Viitattu 23.4.2013]. Saatavana: <http://www.tekla.com/fi/products/tekla-structures/Pages/Default.aspx>

Tekla Structures. Kuvaruudunkaappaus ohjelmakoodista. Rauli Paananen.

Wikipedia. 2013. [Verkkajulkaisu]. AutoCad. [Viitattu 16.4.2013]. Saatavana: <http://fi.wikipedia.org/wiki/AutoCAD>

LIITE 1. Ohjelmakoodi lasip.k

```
/*
!
=====
!
! Ohjelma : lasip.k
! Tekijä : Rauli Paananen
! Päiväys : 07.02.2013
! Kuvaus : Lisää laseihin painotiedon.
! Versio : 1.0
!
!
=====
*/
#define ARRAY_HEIGHT1 23
#define ARRAY_HEIGHT2 5
#define STR_FILE "lasip.ini"
#include "ddb.h"
#include "kdefs.h"
PROGRAM lasip;
INT n_Ok, n_Pos, n_Elem, n_Type, n_Last, n_NumOfSheets, n_GlassThicness, n_NumOfGlass;
STRING*500 str_SymName;
STRING*100 str_Settings_Name[100], str_Settings_Value[100];
REAL r_GlassLDX[500], r_GlassLDY[500], r_GlassRUX[500], r_GlassRUY[500],
r_GlassWidth[500], r_GlassHeight[500];
REAL r_SheetLDX[500], r_SheetLDY[500], r_SheetRUX[500], r_SheetRUY[500], r_SheetWidth[100],
r_SheetHeight[500], r_SheetScale[500];
STRING*100 str_GlassType[500] ;

/*****
Kysymykset
*****/

FUNCTION Questions(): INT;
BEGIN
  REPEAT
    Input( "Anna lasin yhteenlaskettu paksuus? ", n_GlassThicness );
    IF( n_GlassThicness == 0 ) THEN Print("Valinta ei kelpaa\n" );
  UNTIL( n_GlassThicness <> 0);
/* REPEAT
  Input( "Symbolin mittakaava? ", n_Scale );
  IF( n_Scale == 0 ) THEN Print("Valinta ei kelpaa\n" );
  UNTIL( n_Scale <> 0);*/

  Questions = n_GlassThicness;
END

/*****
Asetustiedoston avaus
*****/

FUNCTION OpenSettingsFile(): INT;
STRING*200 str_Path, str_DrgName;
INT n_File;

BEGIN
  Getvar( "CADSHAKEMISTO", str_Path );
  Strset( str_Path, Strlen(str_Path)+1, "\\");
  Strset( str_Path, Strlen(str_Path)+1, "Omat_Komennot\\");
  Strset( str_Path, Strlen(str_Path)+1, STR_FILE );
  // Avataan tiedosto, jos se on olemassa
  IF( Exists( str_Path )) THEN
```

```

BEGIN
  n_File = Open ( str_Path, "r" );
  IF ( Iostat() ) THEN // Tarkistetaan mahdollinen avausvirhe
  BEGIN
    Print ( "Virhe tiedoston avauksessa!\n" );
    Close( n_File);
    OpenSettingsFile = -1;
  END
  ELSE
    OpenSettingsFile = n_File;
  END
  ELSE
  BEGIN
    Print ( "\nTiedostoa ", STR_FILE, " ei löydy sijainnista ", str_Path, "!\n" );
    OpenSettingsFile = -1;
  END
  END

/*****
Luetaan asetukset tiedostosta
*****/

FUNCTION ReadSettingsFile(INT n_File): INT;
STRING*100 str_ReadLine;
INT n_End, n_Ind;
BEGIN
  n_Ind=0;
  Readln( n_File, str_ReadLine ); //Luetaan 1. rivi
  IF ( Iostat() <> 0 ) THEN
  BEGIN
    Print("Asetustiedoston lukemisessa tapahtui virhe!");
    Close( n_File );
    ReadSettingsFile = -1;
  END
  REPEAT
    n_Ind = n_Ind + 1;
    n_End = strstr( str_ReadLine, " " );
    Strmid( str_Settings_Name[n_Ind], str_ReadLine, 1, n_End-1 );
    Strmid( str_Settings_Value[n_Ind], str_ReadLine, n_End+1, strlen(str_ReadLine ));
    Readln( n_File, str_ReadLine ); //Luetaan seuraava rivi
  UNTIL( Iostat() );
  ReadSettingsFile = n_Ind;

END

/*****

Projektitulukon tiedot:

1. Projektinnumero
2. Kohteen nimi
3. Kohteen osoite 1. rivi
4. Kohteen osoite 2. rivi
5. Piirtäjä

*****/

/*****
Etsitään piirustusarkit
*****/

FUNCTION FindSheets(): INT;

```

```

INT n_Attrib, n_Start, n_End, n_Counter, n_NumOfElements;
REAL r_PointX, r_PointY;
STRING*100 str_AttName, str_AttValue, str_Value;
BEGIN
  n_NumOfSheets=0;
  BEGIN
    n_Counter = 0;
    n_Pos = FIRST;
    n_Last = LAST;
    REPEAT
      n_Elem = Ename (n_Type, n_Pos, E_SYMBOL);
      IF( n_Elem == -1) THEN GOTO GotoPassBy;
      n_Ok = Eget (n_Elem, str_SymName);
      IF( Stricmp( str_SymName, str_Settings_Value[1] ) == 0) THEN
        BEGIN
          n_Attrib = Aname( n_Elem, FIRST );
          REPEAT
            IF( n_Attrib == NO_ATTRS ) THEN
              BEGIN
                GOTO GotoPassBy;
              END
            strfill(str_AttValue,"0");
            n_Ok = Aget( n_Elem, n_Attrib, str_AttName, str_AttValue );
            IF( n_Ok <> 0) THEN
              BEGIN
                //Kerätään tiedot
                IF( strcmp( str_AttName, str_Settings_Value[4] ) == 0 ) THEN
                  BEGIN
                    n_Counter = n_Counter + 1;
                    n_Ok = getobject( n_Elem, OD_IP, r_SheetLDX[n_Counter], r_SheetLDY[n_Counter]);
                    //Arkin leveys
                    n_Start = strstr( str_AttValue, ":" );
                    n_End = strstr( str_AttValue, "," );
                    Strmid( str_Value, str_AttValue, n_Start+1, n_End-n_Start-1 );
                    Strset( str_AttValue, n_Start, "*" );
                    Strset( str_AttValue, n_End, "*" );
                    r_SheetWidth[n_Counter] = Strval( str_Value );

                    //Arkin korkeus
                    n_Start = strstr( str_AttValue, ":" );
                    n_End = strstr( str_AttValue, "," );
                    Strmid( str_Value, str_AttValue, n_Start+1, n_End-n_Start-1 );
                    Strset( str_AttValue, n_Start, "*" );
                    Strset( str_AttValue, n_End, "*" );
                    r_SheetHeight[n_Counter] = Strval( str_Value );

                    //Mittakaava
                    n_Start = strstr( str_AttValue, ":" );
                    n_End = strlen( str_AttValue );
                    Strmid( str_Value, str_AttValue, n_Start+1, n_End - n_Start );
                    r_SheetScale[n_Counter] = Strval( str_Value );

                    //Piirustus pohjan oikean ylänurkan koordinaatit
                    r_SheetRUX[n_Counter] = r_SheetLDX[n_Counter] + r_SheetWidth[n_Counter] *
                      r_SheetScale[n_Counter] - Strval( str_Settings_Value[4] ) * r_SheetScale[n_Counter];
                    r_SheetRUY[n_Counter] = r_SheetLDY[n_Counter] + r_SheetHeight[n_Counter] *
                      r_SheetScale[n_Counter] - Strval( str_Settings_Value[4] ) * r_SheetScale[n_Counter];
                    //Print( r_LDX[n_Counter], " ", r_LDY[n_Counter], " ", r_RUX[n_Counter], " ",
                      r_RUY[n_Counter], " ", r_Scale[n_Counter], "\n");
                  END
                END
            ELSE Print( "Tietojen lukeminen ei onnistunut");
            n_Attrib = Aname( n_Elem, NEXT );
          END
        END
      REPEAT
    END
  END

```

```

    UNTIL( n_Attrib == NO_MORE_ATTRS );
END
GotoPassBy;;
n_Pos = NEXT;
UNTIL( n_Elem == -1 );
n_NumOfSheets = n_Counter;
END
FindSheets = n_NumOfSheets;
END

```

```

/*****
Etsitään lasit
*****/

```

```

FUNCTION FindSGlass(): INT;
INT n_Attrib, n_Start, n_End, n_Counter, n_NumOfElements;
REAL r_PointX, r_PointY;
STRING*100 str_AttName, str_AttValue, str_Value;
BEGIN
  Cads("SEULA SY");
  n_NumOfElements = Select("Valitse lasit" );
  //Print("Elementtejä löytyi ", n_NumOfElements);
  Cads ("SEULA EI");
  IF( n_NumOfElements <> 0) THEN
  BEGIN
    n_Counter = 0;
    n_Pos = FIRST;
    n_Last = LAST;
    REPEAT
      n_Type = Sget( n_Pos, n_Elem);
      IF( n_Elem == -1) THEN GOTO GotoPassBy;
      n_Type = Eget( n_Elem, str_SymName );
      IF( Stricmp( str_SymName, str_Settings_Value[2] ) == 0) THEN
      BEGIN
        n_Counter = n_Counter + 1;
        n_Attrib = Aname( n_Elem, FIRST );
        REPEAT
          IF( n_Attrib == NO_ATTRS ) THEN
          BEGIN
            GOTO GotoPassBy;
          END
          strfill(str_AttValue, "\0");
          n_Ok = Aget( n_Elem, n_Attrib, str_AttName, str_AttValue );
          IF( n_Ok <> 0) THEN
          BEGIN
            //Kerätään tiedot
            IF( stricmp( str_AttName, str_Settings_Value[3] ) == 0 ) THEN
            BEGIN
              n_Ok = getobject( n_Elem, OD_IP, r_GlassLDX[n_Counter], r_GlassLDY[n_Counter]);
              //Lasin leveys
              n_Start = 1;
              n_End = strstr( str_AttValue, "*" );
              Strmid( str_Value, str_AttValue, n_Start, n_End-n_Start+1 );
              r_GlassWidth[n_Counter] = Strval( str_Value );
              //Print("Korkeus on ", r_GlassWidth[n_Counter]);

              //Lasin korkeus
              Strmid( str_Value, str_AttValue, n_End+1, Strlen(str_AttValue) );
              Strset( str_AttValue, n_Start, "*" );
              Strset( str_AttValue, n_End, "*" );
              r_GlassHeight[n_Counter] = Strval( str_Value );
              //Print("Leveys on ", r_GlassHeight[n_Counter]);
            END
          END
        END
      END
    END
  END

```

```

        END
        IF( strcmp( str_AttName, str_Settings_Value[8]) == 0 ) THEN
        BEGIN
            str_GlassType[n_Counter]=str_AttValue;
            //Print(str_GlassType[n_Counter],"\n");
        END
    END
    ELSE Print( "Tietojen lukeminen ei onnistunut");
    n_Attrib = Aname( n_Elem, NEXT );
    UNTIL( n_Attrib == NO_MORE_ATTRS );
END
GotoPassBy;;
n_Pos = NEXT;
UNTIL( n_Elem == -1 );
n_NumOfGlass = n_Counter;
IF( n_NumOfGlass == 0 )THEN
BEGIN
    Print("Yhtään lasia ei löytynyt\n");
    STOP;
END
END
FindSGlass = n_NumOfGlass;
END

/*****
Järjestetään lasit tyypeittäin
*****/

PROCEDURE ArrangeGlasses();
INT n_Ind, n_Ind2;
REAL r_Help;
STRING*100 str_Help;
BEGIN
    n_Ind = 0;
    n_Ind2 = 0;
    REPEAT
        n_Ind = n_Ind+1;
        //Print(n_Ind,"\n");
        n_Ind2 = 1;
        REPEAT
            IF( Strcmp(str_GlassType[n_Ind],str_GlassType[n_Ind2]) < 0 ) THEN
            BEGIN
                str_Help = str_GlassType[n_Ind];
                str_GlassType[n_Ind] = str_GlassType[n_Ind2];
                str_GlassType[n_Ind2] = str_Help;

                r_Help = r_GlassLDX[n_Ind];
                r_GlassLDX[n_Ind] = r_GlassLDX[n_Ind2];
                r_GlassLDX[n_Ind2] = r_Help;

                r_Help = r_GlassLDY[n_Ind];
                r_GlassLDY[n_Ind] = r_GlassLDY[n_Ind2];
                r_GlassLDY[n_Ind2] = r_Help;

                r_Help = r_GlassRUX[n_Ind];
                r_GlassRUX[n_Ind] = r_GlassRUX[n_Ind2];
                r_GlassRUX[n_Ind2] = r_Help;

                r_Help = r_GlassRUY[n_Ind];
                r_GlassRUY[n_Ind] = r_GlassRUY[n_Ind2];
                r_GlassRUY[n_Ind2] = r_Help;

                r_Help = r_GlassWidth[n_Ind];

```

```

    r_GlassWidth[n_Ind] = r_GlassWidth[n_Ind2];
    r_GlassWidth[n_Ind2] = r_Help;

    r_Help = r_GlassHeight[n_Ind];
    r_GlassHeight[n_Ind] = r_GlassHeight[n_Ind2];
    r_GlassHeight[n_Ind2] = r_Help;

    END
    //Print(n_Ind2, "\n");
    n_Ind2 = n_Ind2 + 1;

    UNTIL( n_Ind2 == n_NumOfGlass+1 );

    UNTIL( n_Ind == n_NumOfGlass );
END

/*****
Etsitään oikea mittakaava
*****/

FUNCTION FindScale(REAL r_GlassPointX, REAL r_GlassPointY): INT;
INT n_Find, n_Ind, n_Scale;
REAL r_Scale;
STRING*100 str_AttName, str_AttValue, str_Help, str_Help2;
BEGIN
    n_Ind = 0;
    n_Find = 0;
    n_Scale = 0;
    REPEAT
        n_Ind = n_Ind + 1;
        IF((r_GlassPointX>r_SheetLDX[n_Ind]) & (r_GlassPointY>r_SheetLDY[n_Ind]) &
            (r_GlassPointX<r_SheetRUX[n_Ind]) & (r_GlassPointY<r_SheetRUY[n_Ind]) ) THEN
            BEGIN
                n_Scale = r_SheetScale[n_Ind] ;
                n_Ind = n_NumOfSheets;
                n_Find = 1;
            END
        UNTIL( n_Ind > n_NumOfSheets-1 );
        IF( n_Find == 0) THEN
            BEGIN
                Print( "Piirustusarkkia ei löytynyt, käytetään kuvan mittakaavaa\n");
                Getvar( "MITTAKAAVA", r_Scale);
                n_Scale = r_Scale;
            END
            FindScale = n_Scale;
        END
END

/*****
Tutkitaan onko lasipainosymbolin tasoa olemassa
*****/

FUNCTION FindLayer(): INT;
INT n_Pos, n_Lock, n_On, n_Find;
STRING*100 str_Layer;
BEGIN
    n_Find = 0;
    n_Pos = FIRST;
    WHILE (Searchlayer(n_Pos, str_Layer, n_On, n_Lock)) DO
        BEGIN
            IF( Strcmp( str_Layer, str_Settings_Value[9]) == 0 ) THEN
                BEGIN
                    Print("TASO: ", str_Layer, "\n");
                END
            END
        END
    END
END

```

```

    n_Find = 1;
    END
    n_Pos = NEXT;
    END
    FindLayer = n_Find;
    END

```

```

/*****
Sijoitetaan symboli kuvaan
*****/

```

```

FUNCTION AddLabel(): INT;
INT n_Ind, n_Weigth, n_Scale;
STRING*100 str_Help, str_Question, str_Layer;
BEGIN
    n_Ind = 1;
    n_Scale = 0;
    ArrangeGlasses();
    Getvar( "TASO", str_Layer);
    IF( FindLayer() == 0 ) THEN
        Cads("TASO LUO^#"str_Settings_Value[9]^# 1^#);
        Cads("TASO VAIHDA^#"str_Settings_Value[9]^#);
        str_Question = "Anna lasityypin ";
        Strset( str_Question, strlen(str_Question)+1, str_GlassType[n_Ind]);
        Strset( str_Question, strlen(str_Question)+1, " lasien yhteenlaskettu paksuus mm: ");
        Input( str_Question, n_GlassThicness );
        str_Help = str_GlassType[n_Ind] ;
        REPEAT
            IF( Strcmp( str_Help, str_GlassType[n_Ind]) <>0 ) THEN
                BEGIN
                    str_Question = "Anna lasityypin ";
                    Strset( str_Question, strlen(str_Question)+1, str_GlassType[n_Ind]);
                    Strset( str_Question, strlen(str_Question)+1, " lasien yhteenlaskettu paksuus mm: ");
                    Input( str_Question, n_GlassThicness );
                END
                //Print( str_GlassType[n_Ind], "\n" );
                n_Scale = FindScale(r_GlassLDX[n_Ind],r_GlassLDY[n_Ind]);
                IF( n_Scale == 0 ) THEN n_Scale = 20;
                n_Weigth = Round(r_GlassWidth[n_Ind] * r_GlassHeight[n_Ind] * n_GlassThicness * 2.5 *
0.000001);
                Cads("TASO VAIHDA^#"str_Settings_Value[9]^#);
                Cads("SYMBOLI HAE lasip^#"(r_GlassLDX[n_Ind]+Strval(str_Settings_Value[5])*n_Scale),
                    (r_GlassLDY[n_Ind]+Strval(str_Settings_Value[6])*n_Scale) n_Scale n_Scale 0 n_Weigth^#);
                Cads("TASO VAIHDA^#"str_Layer^#);
                str_Help = str_GlassType[n_Ind] ;
                n_Ind = n_Ind + 1;
            UNTIL(n_Ind > n_NumOfGlass);

            AddLabel = 0;
        END

```

```

/*****
Pääohjelma
*****/

```

```

INT n_File, n_Help;
BEGIN
    n_File = OpenSettingsFile();
    IF( n_File == -1 ) THEN
        STOP; //Lopetetaan ohjelma
    
```



```
n_Ok = ReadSettingsFile( n_File );  
Close( n_File);  
IF( n_Ok == -1 ) THEN  
  STOP; //Lopetetaan ohjelma  
n_Ok = FindSheets();  
Getvar( "KI9", n_GlassThicness);  
Setvar( "KI9", n_GlassThicness);  
n_Ok = FindSGlass();  
n_Ok = AddLabel();
```

END