TAMPEREEN AMMATTIKORKEAKOULU
Tietotekniikan koulutusohjelma
Ohjelmistotuotanto

Opinnäytetyö

Jarkko Hoivala

GDL-OBJEKTIEN OHJELMOINTI PK11-PERUSOBJEKTIMALLIN MUKAAN

Työn ohjaaja Erkki Hietalahti
Työn teettäjä M.A.D. Oy
Tampere 2008

TAMPEREEN AMMATTIKORKEAKOULU
Tietotekniikan koulutusohjelma
Ohjelmistotuotanto
Hoivala, Jarkko

Opinnäytetyö 76 sivua, 5 liitettä
Työn ohjaaja Erkki Hietalahti
Työn teettäjä M.A.D. Oy, valvojana toimitusjohtaja Pauli Jantunen
Toukokuu 2008
Hakusanat CAD, tietokoneavusteinen rakennussuunnittelu, GDL, objektiteknologia, ArchiCAD, BIM, IFC, objektimalli

TIIVISTELMÄ

Älykkäät ja monipuolista tietoa sisältävät virtuaaliset GDL-objektit ovat tietokoneavusteisen arkkitehti- ja rakennussuunnittelun tärkeimpiä rakennuselementtejä, joiden sisältämän informaation avulla rakennuksen ominaisuuksia voidaan analysoida monin hyödyllisin tavoin, kuten visualisoida tiloja tai suorittaa energiasimulaatioita eri ratkaisujen suhteen.

Tarkoituksena on ollut tutkia GDL-objektiteknologian taustaa, ominaisuuksia ja työtapoja sekä laatia PK11-perusobjektimallin mukaisia objekteja GDL-ohjelmointikielellä ArchiCAD-ohjelmistoa käyttävien suomalaisten suunnittelijoiden käyttöön ohjelmiston kylkiäisenä tulevan perusobjektikirjaston osana.

Objektiteknologiaa hyödyntäviä tahoja ovat pääasiassa rakennusosavalmistajat, rakennusalan suunnittelijat ja rakennusprojekteja toteuttavat osapuolet. Koska kyseessä on melko uusi menetelmä, on tarvetta tuottaa laadukkaita ja soveltuvia objekteja osapuolien tarpeisiin. Perusobjektikirjastoon sisällytettävät objektit on toteutettu maamme erityisolot huomioon ottaen.

Ohjelmistona GDL-objektien toteuttamisessa on ollut ArchiCAD 11:n opiskelijaversio, jonka GDL-ohjelmointiin tarjoamat ominaisuudet ovat vastaavat kuin kaupallisessa versiossa.

Opinnäytetyön aikana syntynyt dokumentaatio toimii ytimekkäänä yleiskatsauksena GDL-ohjelmointiin mahdollisen itseopiskelun helpottamiseksi. Työn aikana toteutetut GDL-objektit sisällytetään joko peruskirjastoon tai toimitetaan suoraan tilauksesta suunnittelijoille.

TAMPERE POLYTECHNIC

Bachelor of Engineering
Software Engineering
Hoivala, Jarkko
Programming of GDL Objects Based on PK11 Basic Object Model
Engineering Thesis 76 pages, 5 appendices
Thesis Supervisor Erkki Hietalahti
Commissioning Company M.A.D. Oy. Supervisor: CEO Pauli Jantunen
May 2008
Keywords Computer Aided Designing, CAD, GDL, Object technology, ArchiCAD, BIM, IFC, Object Model

ABSTRACT

Intelligent virtual design objects, known as GDL objects, are commonly used in computer aided architecture and building designs. Information embedded within these objects can be used for multiple types of virtual building simulations, such as energy economics and visual appearance.

Purpose of the thesis has been to study backgrounds and features of the GDL object technology and summarize suitable methods based on PK11 basic object model for producing GDL objects for object library to be shipped with Finnish ArchiCAD software package.

GDL object are mostly used by building part manufacturers, building related designers and different parties of the construction executives. While the object technology is relatively new but highly productive, there is a growing demand for suitable and high quality objects.

Objects produced during the thesis process will be either included into the basic object library or delivered directly to the designers according to the order.

This thesis can be used as an brief introduction into GDL object technology and programming multi-featured quality objects based on PK11 basic object model developed by M.A.D. Oy and Laurimark Oy.

ALKUSANAT

Ohjelmoinnin lisäksi arkkitehti- ja sisustussuunnittelu sekä 3D-mallinnus ovat kiinnostaneet itseäni henkilökohtaisesti jo pitkään ja nämä osa-alueet yhdistyvät mielestäni mielenkiintoisella ja haastavalla tavalla GDL-objektien ohjelmoinnin kautta.

Lokakuussa 2007 osallistuin M.A.D. Oy:n Roope Syvälahden vetämään ArchiCAD 11 Road Show-tilaisuuteen TAMK:ssa, missä kuulin lyhyesti GDL-ohjelmoinnista ja osaavien ohjelmoijien tarpeesta. Tammikuussa 2008 aloitin kahden kuukauden harjoittelujakson M.A.D. Oy:ssä keskittyen GDL-objektien ohjelmoinnin itseopiskeluun, minkä aikana muodostui ajatus opinnäytetyön aiheesta.

GDL-ohjelmoinnista ei ole saatavilla suomenkielistä lähdemateriaalia, eikä ohjelmointia edesauttavien objektimallien osalta löytynyt mitään lähdemateriaalia lukuunottamatta lyhyttä yhteenvetoa M.A.D. Oy:n perusobjektikirjaston dokumentaatiossa. GDL-ohjelmoinnista ei myöskään ole tehty opinnäytetyötä, tai ainakaan sellaista ei lähdeaineistoa etsittäessä löytynyt.

Edellä mainituista syistä halusin tutkia ja dokumentoida GDL-ohjelmoinnin periaatteita, sovelluskohteita ja objektimallien käyttöä sekä yleisesti että PK11-perusobjektimallin osalta.

Kahden kuukauden harjoittelun ja itsenäisen GDL-ohjelmoinnin opiskelun myötä olen törmännyt moniin ongelmiin. Pyrkimykseni on ollut alusta asti dokumentoida niitä ja löytää ratkaisuja omaa oppimistani ja opinnäytetyötä silmällä pitäen. Olen suunnitellut tutkintotyöni sisällön ja hierarkian niin, että se olisi GDL-ohjelmoinnista kiinnostuneelle lukijalle mahdollisimman selkeä, hyödyllinen ja kattava.

GDL-ohjelmoinnin lisäksi on täytynyt perehtyä tarkemmin myös rakennusalaan, sen tarpeisiin ja termistöön, joita olen tutkinut myös tämän tutkintotyön puitteissa. Lukijalle, joka todennäköisesti on joko rakennussuunnittelun tai ohjelmistotuotannon alalla, on perusteltua esittää GDL-ohjelmoinnin ympärillä

vaikuttavia asioita ja ilmiöitä. Tästä syystä asiaa on verrattain paljon, mutta kokonaiskuvan kannalta niitä ei voi jättää tässä tutkintotyössä esittämättä.

Työ tehtiin M.A.D. Oy:n tilauksesta, vaikka itse pystyinkin paljolti vaikuttamaan aiheen valintaan. Haluan kiittää M.A.D. Oy:n toimitusjohtaja Pauli Jantusta ja GDL-ohjelmoinnin ”gurua” Ville Rantasta sekä muita työntekijöitä ja tahoja, jotka olivat edesauttamassa GDL:n salojen selvittämistä ja tämän opinnäytetyön valmistumista.

Tampereella 16. toukokuuta 2008

Jarkko Hoivala

KÄYTETYT TERMIT JA LYHENTEET

2D	Kaksiulotteisesta tasosta käytetty lyhenne
3D	Kolmiulotteisesta avaruudesta käytetty lyhenne
4D	4D = 3D + aika, eli aika-aspektin linkittämistä 3D-mallin rakennusosa- ja tilaolioihin. Aika-aspekti voi kuvata esim. rakennusosien asennuksen ajankohtaa, joilloin 4D-simuloinnilla voidaan visualisoida rakentamisen etenemistä ajassa.
5D	4D + tuotannonohjaus, eli virtuaalirakentaminen
API	Ohjelmointirajapinta (Application Programming Interface) on käyttöliittymä, jonka välityksellä eri ohjelmat voivat keskustella keskenään
BIM	Rakennuksen tuotetietojen kokonaisuus. Rakennuksen ja rakennusprosessin elinkaaren aikaisten tuotetietojen kokonaisuus, Building Information Model (BIM)
CAD	Tietokoneavusteinen suunnittelu
GUIDe04	Käyttöliittymäsuunnittelumalli
IFC	Kansainvälinen rakennusalan standardi oliopohjaisen tiedon siirtämiseksi tietojärjestelmästä toiseen
ITC	Rakentamisen tietotekniikka (Information Technology in Construction)
GDL	Avoimen standardin geometrinen kuvauskieli
GDL-objekti	Geometrisellä kuvauskielellä toteutettu parametrisoitu objekti eli suunnitteluelementti ArchiCAD-ohjelmistoon, mutta siirrettävissä myös muihin CAD-ohjelmistoihin
LVI	Kiinteistön ja siihen liittyvien tilojen teknisten palveluiden, järjestelmien ja laitteiden kokonaisuus
PK11	M.A.D. Oy:n tuottaman GDL-objektikirjaston tämän hetkisen version lyhenne
Planssi	Yleisnimitys rakennusalan dokumenttipohjille, joille piirrokset ja suunnitelmat liitetään
SFS	Suomen Standardisoimisliitto
UML	Object Management Groupin (OMG) standardoima graafinen mallinnuskieli

SISÄLLYS

1 JOHDANTO

Tietokoneohjelmistojen yleistyminen kaikilla aloilla on synnyttänyt ala- ja ohjelmistokohtaisia ohjelmointikieliä, jotka on suunniteltu tuottamaan tehokkaita ratkaisuja eri alojen erityistarpeisiin.

Tässä tutkintotyössä on tutkittu rakennussuunnittelussa yleisesti käytettyä GDL-ohjelmointikieltä ja sillä tuotettujen suunnitteluobjektien ominaisuuksia ja mahdollisuuksia.

Rakennusalalla on omat erityispiirteensä alalla tarvittavan tiedonkäsittelyn suhteen ja nykyään alalla puhutaankin muun muassa rakennusten tietomallinnuksesta, rakennussimulaattoreista ja visualisoinneista. Tiedon määrä on valtava ja sen hallitsemiseksi useat tahot ovatkin asettaneet pakollisia tietomallinnukseen perustuvia vaatimuksia hankkeilleen.

GDL-objektiteknologia sitoo yhteen perinteisen rakennuspiirtämisen ja nykyaikaisen tietomallinnuksen älykkäällä tavalla. Virtuaalinen rakennustuoteosa voi tuntea omat ominaisuutensa ja rajansa, mutta olla silti muokattavissa suunnittelijan toimesta projektin tarpeita vastaavaksi. Suunnittelijan antamien objektin tuotetietojen perusteella voidaan simuloida sen vaikutusta koko rakennuksen ominaisuuksiin, kuten energiataloudellisuuteen tai ulkonäköön.

GDL-objektit ovat oleellisella tavalla osa tiedon tehokasta hyödyntämistä rakennusalalla, joten niiden tarkastelu historian, nykytilan ja tulevaisuuden tarjoamista mahdollisuuksista teknologian kehittyessä on kilpailukyvyn kannalta tarpeen niin rakennus- kuin ohjelmistoalallakin toimiville yrityksille ja oppilaitoksille.

1.1 Tiedon ja tiedonkäsittelyn periaatteita

Tietotekniikan tehokkaan hyödyntämisen tärkein käsite on tieto (data). Tiedolle täytyy aina määritellä muoto (syntaksi) ja tarkoitus (semantiikka), jotta sillä olisi jotain merkitystä käyttäjälleen. Yksinkertaisena esimerkkinä voidaan ajatella

tilannetta, jossa kaksi ihmistä keskustelee eri kielillä tietäen esittämästään asiasta kaiken, mutta kuulija ei ymmärrä vieraskielisestä puheesta mitään. Mahdollisuutena on hyödyntää sanakirjaa tai ulkopuolista tulkkia, joka kääntää esitetyt asiat ymmärrettävälle kielelle. Tilanne on periaatteessa aivan sama tietotekniikassa.

Tiedon tärkein merkitys käyttäjälleen on sen uudelleenkäytettävyydessä ja jaettavuudessa. Uuden tiedon tuottaminen perustuu lähes poikkeuksetta aikaisemman tiedon käyttämiseen niin, että erilliset tiedon osaset muodostavat uuden käsitteellisen kokonaisuuden, jota voidaan käyttää edelleen tiedon osasena. Tiedon osasta voisi ajatella abstraktina objektina, joka koostuu muista abstrakteista objekteista.

Tiedon käyttämiseen liittyy tiedon olemus. Onko tieto staattista, jolloin tieto on itseisarvo sellaisenaan, vai funktionaalista, joka mahdollistaa tiedon käyttämisen merkityksellisten tulosten tuottamiseksi käyttämällä perusteltuja argumentteja? Esimerkkinä funktionaalisesta tiedosta olkoon esimerkiksi luonnontieteelliset kaavat ja funktiot. Staattista tietoa voisi havainnollistaa toteamuksella ”hauki on kala” tai sovitulla luonnonvakiolla ”c = valonnopeus tyhjiössä [m/s]”.

Jalostettaessa tiedon olemusta käytettävyyden kannalta, sille voidaan määritellä tietty rakenne. Rakenteen eli tietomallin tulee olla tarkkaan harkittu, mahdollisesti jopa standardisoitu käyttötarkoitustaan parhaiten vastaavaksi. Tällöin tiedon hyödyntämistä voidaan toteuttaa mahdollisimman laajalti ja tehokkaasti. Vakiorakenteinen tietomalli nopeuttaa tiedonkäsittelyä olettaen, että käyttäjä on omaksunut tietomallin omaan toimintatapaansa. Esimerkkinä mainittakoon tiedonhakua ja luettavuutta parantavat dokumentoinnin tyyli- ja rakenneohjeet.

1.2 GDL-teknologian historiaa

Ensimmäiset CAD-sovellukset (Computer Aided Design) ottivat ensiaskeliaan 1960-luvulla auto- ja lentokoneteollisuudessa koneistusautomaation ohjauksessa (tuotannosta käytetään lyhennettä CAM – Computer Aided Manufacture). 1980-luvulle tultaessa yleiskäyttöiset tietokoneet yleistyivät ja ensimmäinen PC-tietokone näki päivänvalon. Tämä aloitti 2D-suunnitteluohjelmistojen kehityksen. /1/ Vuonna 1982 julkistettiin Autodeskin AutoCAD 1.0, jolla suunnittelija pystyi

piirtämään 2D-kuvia tarkkojen mittojen mukaan. Ongelmana oli vielä tuolloin kertaalleen tuotettujen suunnittelutietojen heikko uudelleen hyödyntäminen.

Arkkitehtisuunnitteluohjelmistojen edelläkävijä unkarilainen Graphisoft kehitti avoimen GDL-objektiteknologian, joka esiteltiin ArchiCAD 1.0 -versiossa (Apple Macintosh Plussalle) vuonna 1984, jolloin suunnittelijoille annettiin mahdollisuus suunnitella objekteja ja tallentaa niitä objektikirjastoon. Kirjastoon tallennettuja objekteja voitiin uudelleen käyttää ja jakaa toisten suunnittelijoiden kesken. /2/ Näin otettiin ensiaskeleita digitaalisten objektien avoimen tietomallin hyödyntämisessä.

Tietokoneavusteisen rakennussuunnittelun kehityksen kannalta on ollut merkittävää, että siirryttäessä paperilta näytölle suunnittelun käsitteet pysyvät ymmärrettävinä. Suunnittelijoiden lähtökohtana ovat erilaiset objektit: konkreettiset esineet, joilla on erityisiä piirteitä kuten ulottuvuus, tilantarve, materiaali, muoto, sijainti ja suunta. Ohjelmistojen suunnittelun perustana nykyään on oliopohjainen toteutustapa (engl. object-oriented programming). Näiden seikkojen vuoksi on johdonmukaista toteuttaa myös CAD-ohjelmiston suunnitteluelementit objekteina.

Arkkitehtuurisuunnittelun harppaus käsin piirtämisestä tietokoneavusteiseen suunnitteluun on ollut merkittävä askel tiedonkäsittelyn ja -varmennuksen kannalta. Tietokone on voinut huomauttaa suunnittelijaa yleisistä suunnittelu- ja piirtovirheistä jo niiden syntyvaiheessa, mikä on säästänyt aikaa ja kustannuksia. Mitä kauemmin aikaa kuluu virheen tekemisestä virheen toteamiseen ja korjaamiseen, sitä enemmän siitä koituu kustannuksia arkkitehtisuunnittelualan yrityksille.

GDL-teknologian ja tuotemalliajattelun yleistyminen rakennussuunnittelussa vähentää rakennustiedon ja -suunnitelmien siirtoa eri projektin osapuolien kesken ilman tietohäviöitä. Nykyinen tilanne vaatii vielä eri ohjelmistojen ja osapuolien panostusta yhteisen tietomallin hyödyntämiseksi. /3/ Tietoa häviää, kun 2D-/3D-mallinnettu rakennus tulostetaan piirustuksina paperille. Tavoitteena kaikkien edun kannalta täytyisi olla yhteinen ja kaikilla käytössä oleva rakennuksen tietomalli, johon eri osapuolet tuottaisivat oman osuutensa rakennusprojektin edetessä. Suunnitteluvaiheessa tietomallin avulla voidaan suorittaa monipuolisia simulaatioita, laskea kustannuksia (4D-simulaatio) ja suunnitella tuotannonohjausta

(5D-simulaatio) /4; 5/. Tietomallilla olisi käyttöä myös rakennuksen ylläpidossa koko sen elinkaaren ajan.

Älykkäiden GDL-objektien tarkoituksena on osaltaan vähentää suunnittelussa mahdollisten virheiden mahdollisuutta. Ohjelmallisesti voidaan tarkastaa objektin käytännöllisyys ja oikeellisuus mittojensa ja muiden ominaisuuksiensa suhteen ja estää käyttökelvottomien arvojen käyttäminen objektissa. Suunnittelija voi näin ollen keskittyä tehokkaammin oleelliseen eli luovaan suunnittelutyöhön.

Mahdollisuus parametrisoida objekteja on ollut GDL-ohjelmointikielen vahvoja etuja kehityksen ja yleistymisen kannalta verrattuna muihin geometrisiin kuvauskieliin, esimerkiksi ARX-, O2C- ja OFM-kieliin. Lisäksi GDL-ohjelmointikielellä voidaan määritellä objektin ei-geometrisiä ominaisuuksia, kuten käytettyjä materiaaleja, värejä, määrälaskentatietoja, käyttöliittymää ym., joista lisää luvussa 3.

PK11-objektimalli julkistettiin ArchiCAD 11 -ohjelmiston yhteydessä vuonna 2007. ”PK” tulee sanasta ”peruskirjasto” ja ”11” ArchiCAD-versiosta, jonka teknisiin ominaisuuksiin se on optimoitu. /6/ Objektimallit tarjoavat suunnittelijoille joukon kehittyneitä valmiuksia objektien toteuttamisen tehostamiseksi ja yhtenäistämiseksi.

GDL-objektien geometrioiden ja oheistiedon siirrettävyys eri ohjelmistojen kesken on kehittynyt vuosien varrella. Objektit voidaan viedä esimerkiksi AutoCAD-ohjelmaan erillisen lisäsovelluksen avulla. Kansainvälisen IFC-tiedonsiirtostandardin yleistyminen ja sen ominaisuuksien laajentuminen käsittämään useiden eri alojen suunnittelutarpeita vastaavaa tietoa on vasta aluillaan /7/. Häviötöntä tiedonsiirtoa eri ohjelmistojen kesken on tutkittu esimerkiksi Spadex-projektissa, tosin vain rakennuselementtien geometrioiden osalta käytettävissä olleen IFC 1.5.1 -version takia /8/.

Objekti voi kuvata suunnitelmassa niin rakennusosaa kuin työvälinettäkin, sisustuselementistä puhumattakaan. Tästä syystä esiintyy kaikilla objekteiksi kutsuttavilla elementeillä vain muutama yhteinen parametri ja valtava joukko yksilöllisiä ominaisuuksia, joita ei välttämättä ole vielä sisällytetty IFC-standardiin. Tiedonsiirtämisen lähtökohtana nykytilanteessa on keskittyä pääasiassa yleisten

tietojen siirtämiseen luotettavasti ja yksilöllisten erikoistietojen käsittely toteutetaan tapauskohtaisin ratkaisuin.

Tulevaisuus tuo tullessaan yhä laajempaa GDL- ja IFC-standardien tukea eri ohjelmistojen yhteiskäytölle, joten niillä toteutetut suunnitteluvaiheet ovat kokonaisuudessaan käytettävissä keskenään. Toisaalta taas ArchiCAD-ohjelmiston kehitys voi tuoda tullessaan uusia ominaisuuksia Graphisoftin tai kolmansien osapuolien ohjelmiin, jolloin rakennusprojektin kaikki vaiheet voitaisiin ehkä jo toteuttaakin saman ohjelmiston alla ja tiedonsiirto-ongelmat olisivat näin ratkaistu.

2 GDL-OBJEKTI

Tässä luvussa tarkastellaan GDL-objektia, sen yleistä määritelmää, käyttöä ja toteutustapoja, sekä eri toteutustapojen vaikutusta objektin käytettävyyteen.

2.1 GDL-objektin määritelmä

GDL-objekti on avoimeen standardiin perustuvalla geometrisellä kuvauskielellä toteutettu suunnitteluelementti, joka sisältää GDL-kielistä tyyppi- ja mallikohtaista tietoa suunnitteluelementin mitoista, toimintalogiikasta ja muista ominaisuuksista. Olennaisin tieto on objektin geometrinen kuvaus 2D- ja 3D-tilassa, mutta lisäksi objektiin voidaan sisällyttää lisätietoja esimerkiksi valmistajasta, standardeista, luokista ja määrälaskennasta.

2.2 Objektien käyttö

GDL-objektiteknologian käytettävyys ja hyödyllisyys perustuvat muun muassa seuraaviin ominaisuuksiin:

Soveltuvuus verkkokäyttöön – Online-tuotekuvastojen luominen, käyttäminen ja päivittäminen on nopeaa objektien älykkään luonteen ja pienen tiedostokoon ansiosta.

Joustavuus ja hallinta – Objekteja voidaan käyttää sekä Windows- että Macintosh-ympäristöissä ja ne ovat yhteensopivia myös yleisimpien CAD-formaattien, kuten DXF:n ja DWG:n, kanssa. Lisäksi objektiteknologia tukee yleistyvää IFC-teollisuusstandardia ja XML-tyyppisiä tietokantoja.

Integroitavuus rakennussuunnitelmaan – Rakennuselementtien valinta jo suunnitteluvaiheessa hyödyttää sekä suunnittelijoita että valmistajia. Suunnittelija voi käyttää suunnitelmassaan saatavilla olevia rakennus- ja sisustuselementtejä, kun taas valmistaja voi markkinoida tuotteitaan entistä aikaisemmassa rakennusprojektin vaiheessa.

Tuotehallinta – GDL-objekteihin voidaan sisällyttää monipuolisesti tietoa valmistajasta, laatuvaatimuksista, paloluokista ja muista tarpeelliseksi katsotuista ominaisuuksista. Näitä tietoja voidaan käyttää hyväksi suunnitteluvaiheesta lähtien koko rakennuksen elinkaaren ajan monien eri osapuolten hyödyksi.

Kustannussäästöt – GDL perustuu avoimeen ja ilmaiseen standardiin, jonka avulla voidaan luoda kohtuullisen nopeasti käyttötarkoitustaan vastaavia älykkäitä objekteja, jotka voidaan automaattisesti muuntaa myös eri formaatteihin valmiiden työkalujen avulla. /9/

Tässä luetelluista syistä objektiteknologia sopii moneen eri käyttötarkoitukseen, joista yleisimmät käyttökohteet on otettu tarkasteltavaksi. Näiden lisäksi objektiteknologiaa voidaan käyttää esimerkiksi erilaisten lavasteiden suunnittelussa tai vaikkapa historian tutkimuksessa.

Mainittakoon vielä, että objektiteknologian tarjoaminen laadukkaan ja ilmaisen peruskirjaston muodossa suomenkielisen ArchiCADin oheistuotteena on todettu osaltaan lisäävän ArchiCADin maahantuojan, M.A.D. Oy:n, asiakastyytyväisyyttä ja näin ollen tukee ArchiCADin jalansijaa Suomessa suosittuna arkkitehtisuunnitteluohjelmistona /32/.

2.2.1 Rakennus- ja sisustus- ja arkkitehtisuunnittelu

Yleisin käyttökohde GDL-objektiteknologialle ovat ArchiCAD- ja AutoCAD-ympäristössä toteutettavat rakennus-, sisustus- ja arkkitehtisuunnitelmat.

Arkkitehdin kannalta objektin tärkein ominaisuus on tilavarauksen määrittäminen tiettyyn käyttötarkoitukseen. Rakennussuunnittelijalle tärkein ominaisuus puolestaan on objektin tekninen rakenne ja rakenteeseen liittyvät ominaisuudet, esimerkiksi massa ja lämmöneristävyys. Sisustussuunnittelussa painopiste on itse objektin visuaalisessa toteutuksessa materiaaleineen ja tilanvarauksineen /29/. Monet valmistajat tarjoavat tuotteistaan GDL-objekteja, mistä on suunnittelijalle se hyöty, että objektin saa suoraan ja oikeanlaisena liitettyä suunnitteludokumenttiin ja valmistajan määrittelemät asiat ovat säädettävissä.

2.2.2 GDL-objektit rakennus- ja sisustustuotteiden markkinoinnissa

Rakennusmateriaalien ja -elementtien kirjo on nykyään erittäin laaja ja kilpailu alalla on yhä tiukempaa. Tietotekniikan kilpailukykyä kehittävä hyödyntäminen on haaste niin pienille kuin suurillekin rakennustarvikealan yrityksille. Koska nykyään rakennusten suunnittelu on siirtynyt paperilta tietokoneiden näytöille, on perinteiset rakennustarvikkeiden menekinedistämistavat suositeltavaa päivittää tämän päivän tarpeita vastaaviksi.

Kivialalla toteutettiin mittava 10 miljoonan euron Kivi-teknologian kehittämisohjelma vuosien 1999 - 2003 aikana, jolloin Teknologian kehittämiskeskus TEKES ja Kiviteollisuusliitto ry. ottivat päämääräkseen luoda Suomen kiviteollisuudesta maailmanlaajuisesti alan johtava tietotekniikan hyväksikäyttäjä.

Kehittämisohjelman taustalla oli Kiviteollisuusliiton toimitusjohtaja Pekka Jauhiaisen mukaan se, että kiven käytön lisääntymiselle oli kaksi merkittävää estettä: hinta ja tiedon puute. Hinnan kohdalla hän epäili olevan kysymyksen enemmän hintamielikuvasta kuin todellisesta hintatasosta, joten kiviportaalin keskeinen tavoite oli tarjota lisää tietoa.

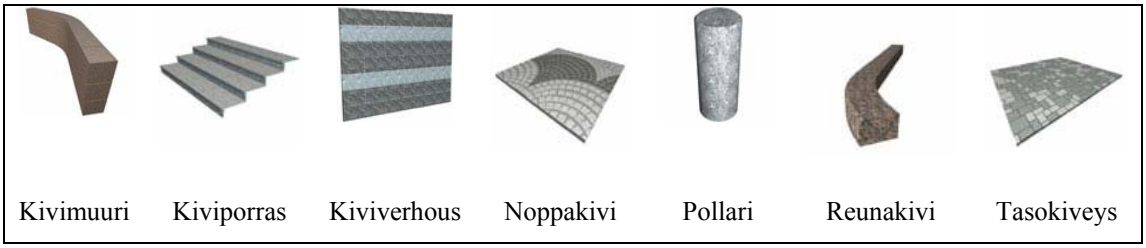
Kiven käytön kannalta suunnittelijat nähdään tärkeässä asemassa, sillä he tekevät tärkeimmät kiven menekkiin vaikuttavat päätökset. Sen vuoksi suunnittelijoille haluttiin tarjota konkreettinen työkalu kiven hyödyntämiseen. GDL-teknologia soveltui hyvin suomalaisille suunnittelijoille, joiden tietotekniset ratkaisut ja osaaminen oli jo tuolloin korkeaa tasoa.

Kiviobjektien siirtäminen portaalista suunnittelijan käyttöön toteutettiin kahdella tavalla; suunnittelija voi selata objekteja internet-selaimellaan ladattuaan käyttöönsä GDL Web Plugin -paketin tai hän voi ladata koko kirjaston omalle koneelleen kokonaisuudessaan käytettäväksi.

Kuhunkin kiviobjektiin on kytketty mittava määrä tietoa, muun muassa kiviobjektin valmistamiseen soveltuvien kivien tekstuurit ja pintakäsittelyt, joista suunnittelija voi vapaasti valita mieleisensä yhdistelmän. Materiaalien tekstuurit generoidaan objektin koodissa, eikä käyttäjän materiaalivalikkoon sisällytetä valtavaa määrää eri kivimateriaaleja. Lisäksi kiviobjekteissa on otettu huomioon ympäristöystävällisyys monipuolisten ympäristöparametrien (kuva 1) avulla. /10/

Kuva 1. Kiviobjektin ympäristöparametrit /11/

Kiviobjektikirjaston toteutuksesta ja mahdollisista päivityksistä vastaa M.A.D. Oy ja kiviportaalin teknisen toteutuksen takana on Artin Net Finland Oy.



Kuva 2. Finstone GDL-kirjaston objektit /11/

2.2.3 Online-tuotekuvasto

Graphisoft GDL Web Plugin on ilmainen lisäohjelma, joka on saatavilla Internet Explorer (Windows), Safari (Mac) ja Mozilla Firefox (Windows & Mac) -selaimiin. Sen avulla on mahdollista toteuttaa GDL-pohjaisia online-tuotekuvastoja, joissa käyttäjä voi katsella objekteja kaikista kulmista ja muokata niiden parametreja graafisesti. Löydettyään haluamansa objektit tarvittavine toimintoineen käyttäjä voi tallentaa yksittäisen objektin koneelleen.

Rakennustarvikkeidensa GDL-objekteja tarjoavalle yritykselle tämä säästää aikaa ja vaivaa objektikirjaston ylläpidossa, sillä objekteja voidaan päivittää yksi kerrallaan ja siirtää päivitykset saman tien suunnittelijoiden käyttöön ilman laajempaa kirjastopäivitystä. Kirjastopäivitykset ovat järkeviä silloin, kun voidaan tarjota useamman objektin päivitetyt versiot kerralla.

GDL-pohjaisten online-tuotekuvastojen tarjonta on kansainvälisestikin vielä melkoisen vähäistä. Suurin osa objekteista on tarjolla vain renderoidun kuvan perusteella, jolloin potentiaalinen käyttäjä ei voi itse todeta käytettävyyttä ja soveltuvuutta omaan tarpeeseensa ennen objektin lataamista. Tämän teknologian hyödyntäminen vaatii usein eri aloja edustavien yritysten yhteistyötä, kuten arkkitehtisuunnittelun ymmärtämistä, GDL-asiantuntemusta ja www-sivujen tuotantoa. Lisäksi GDL Web Plugin vaatii vielä kehittämistä, jotta se vastaisi käyttöliittymältään ja ulkoasultaan tämän päivän odotuksia. Online-tuotekuvastoa voi testata käytännössä esimerkiksi pohjoismaisen julkiskalusteita valmistavan Fora Form -yrityksen sivuilla /12/.

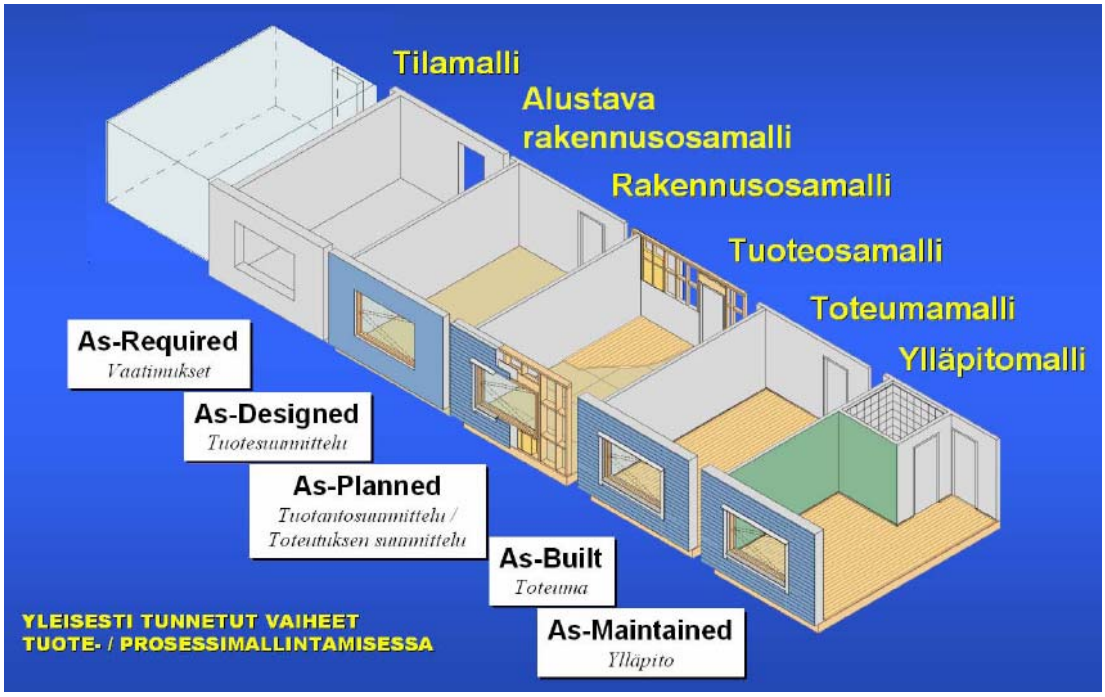
2.2.4 Tietomallinnus

Rakennuksen tietomalli, BIM (Building Information Model), sisältää ideaalitilanteessa kaiken rakennusprojektiin liittyvän tiedon yhdessä paikassa /13/. Objekteja käsiteltäessä on tarkoituksenmukaista käyttää hyväksi sovittuja mallinnussääntöjä, joihin kuuluvat muun muassa yhteiset parametrit. Tätä kutsutaan myös objektin tuotemalliksi.

Objektin tuotemalliin, esimerkiksi tiettyyn rakennuselementtiin, tehdyn muutoksen tulisi päivittyä hallitusti koko rakennussuunnitelmaan; työpiirustuksiin, määrälistoihin, kustannusarvioihin, toimitusaikatauluihin ja muihin asiakirjoihin.

Tietomallinnuksessa tulisi ottaa huomioon objekteihin liittyvät eri osapuolten vaatimukset: valmistajan, suunnittelijan ja tuotantotahon vaatimukset. Valmistajalle objekti on markkinointiväline, jolloin vaatimukset keskittyvät kolmiulotteiseen visuaalisuuteen. Suunnittelun kannalta tärkeintä on objektin käytettävyys ja toiminta yhdessä muiden objektien, ohjelmaversioiden ja ohjelmien kanssa. Tuotanto- ja rakennusprosessia varten suunnitelmasta tulee pystyä tekemään erilaisia simulaatioita, havainnekuvia ja listauksia, joissa kussakin objektin tulee käyttäytyä sille asetettujen vaatimusten mukaan.

Tietomallinnus on tärkeässä osassa suunnittelijan muokkaaman tiedon välittämisessä tuotantoprosessin käyttöön ja eri ratkaisujen vertailuissa. Isoissa rakennushankkeissa tietomallit ovat erittäin hyödyllisiä ja yhä laajenevassa määrin pakollisia. Suomessa esimerkiksi Senaatti-kiinteistöt ja valtioista muun muassa Kiina ovat vaatineet pakollista rakennusten tietomallia uudisrakennushankkeissa vuodesta 2007 lähtien /14; 15/



Kuva 3. Tuotemallintamisen vaiheistus /16/

2.2.4.1 Tietomallin käyttö käytännön rakennusprojekteissa

Senaatti-kiinteistöt on Suomen valtiovarainministeriön alainen liikelaitos, joka huolehtii valtion kiinteistövarallisuuden hallinnasta ja toimitilojen vuokrauksesta.

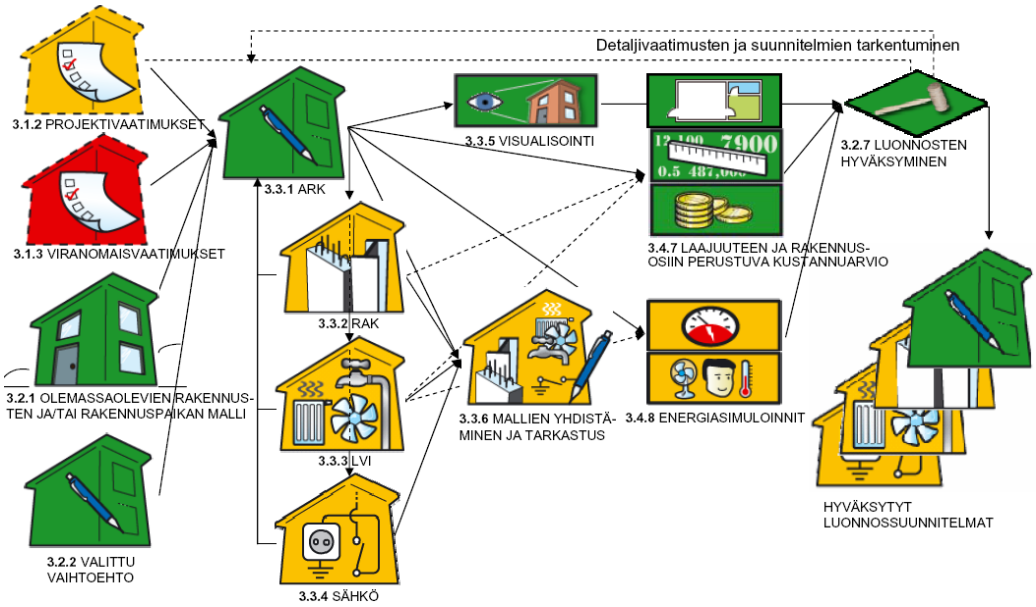
Senaatti-kiinteistöt on laatinut tarkat tietomallivaatimukset rakennusprojekteilleen. Projektin tarvekartoitus- ja hankesuunnitteluvaiheessa määritellään tietomallille projekti- ja viranomaiskohtaiset kehykset. Senaatti-kiinteistöjen tietomalliohjeistuksen mukaiset ArchiCAD-aloituspohjat ja sen valmiit ominaisuudet ovat suunnittelijan lähtökohtana tietomallin toteutukselle. /30/

Ehdotussuunnitteluvaiheessa näihin lisätään olemassa olevien rakennusten ja rakennuspaikan malli, joista saadaan joukko vaihtoehtoisia massa- ja tilamalleja rakenne- ja LVI-suunnittelijoiden yhteistyöllä. Näiden vaihtoehtojen kustannusarvioiden, energiasimulointien ja visualisointien pohjalta tehdään päätös joko edetä tietyn vaihtoehdon suhteen luonnossuunnitteluun tai jatkaa iterointia mahdollisten vaihtoehtojen kesken.

Luonnossuunnittelussa (kuva 4) suunnittelu siirtyy arkkitehdille, jonka luonnosten pohjalta rakennus-, LVI- ja sähkösuunnittelijat tuottavat omat lisänsä tietomalliin. Eriosapuolten tietomallit yhdistetään ja tarkistetaan. Projekti etenee toteutussuunnitteluun, mikäli luonnossuunnitelmat hyväksytään.

Projektiin osallistuvien välisen jatkuvan yhteistyön myötä syntyy hyväksytty toteutussuunnitelma visualisointeineen, energiasimulointeineen, määräluetteloineen ja rakennusosiin perustuvine kustannusarvioineen.

Urakkatarjousvaiheessa pääurakoitsijaehdokkaat analysoivat projektin tietomallia 4D:nä, jonka pohjalta tekevät tarjouksensa urakasta. Valittu urakoitsija toteuttaa projektin ja tuottaa omistajan ylläpidon ja huollon tarpeisiin huoltokirjan ja toteumamallin. /16/



Kuva 4. Luonnossuunnitteluvaiheen vuokaavio Senaatti-kiinteistöissä /16/

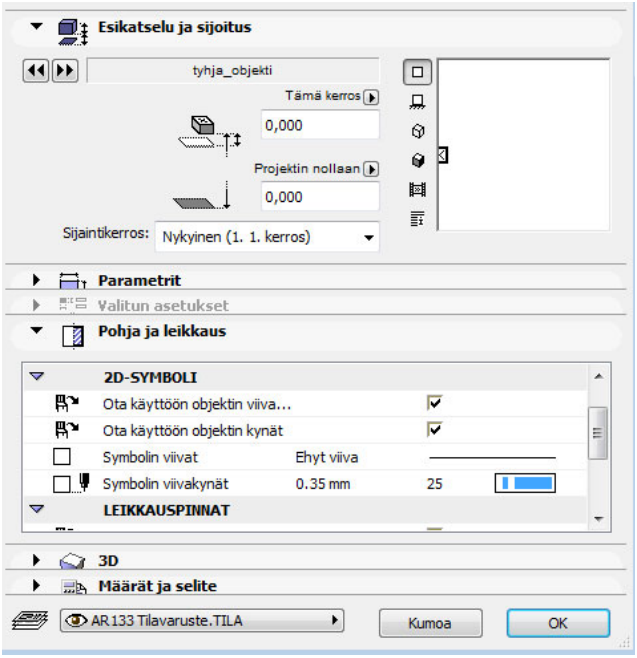
2.3 Objektien käyttöliittymät

Objekteilla on usein eri käyttöliittymiä käytettävyyden tehostamiseksi. Jokaisella objektilla on vähintään käyttöliittymäikkuna ja sen tarjoamat vakiotoiminnot, mutta sen sisälle voidaan toteuttaa parametrien käyttöliittymä joko parametrilistauksen tai erillisen dialogisivun avulla. Lisäksi objektille voidaan toteuttaa graafiset käyttöliittymät 2D- ja 3D-suunnittelutilassa.

2.3.1 Objektin käyttöliittymäikkuna

ArchiCAD 11 tarjoaa joukon valmiita asetustoimintoja objektin käyttöliittymädialogissa. Sen kautta voidaan selata levyllä tai verkossa sijaitsevia objektikirjastoja ja valitaan haluttu objekti esikatselukuvan ja nimikkeen perusteella.

Objektin oletusasetusten käyttöliittymäikkuna (kuva 5) sisältää esikatselutoiminnot 2D- ja 3D-tiloissa, sijoituspisteen, -korkeuden ja -kerroksen asetukset sekä useita välilehtiä, jotka on jaoteltu alla esiteltyjen sisältöjen mukaan.



Kuva 5. Objektin parametrien käyttöliittymäikkuna

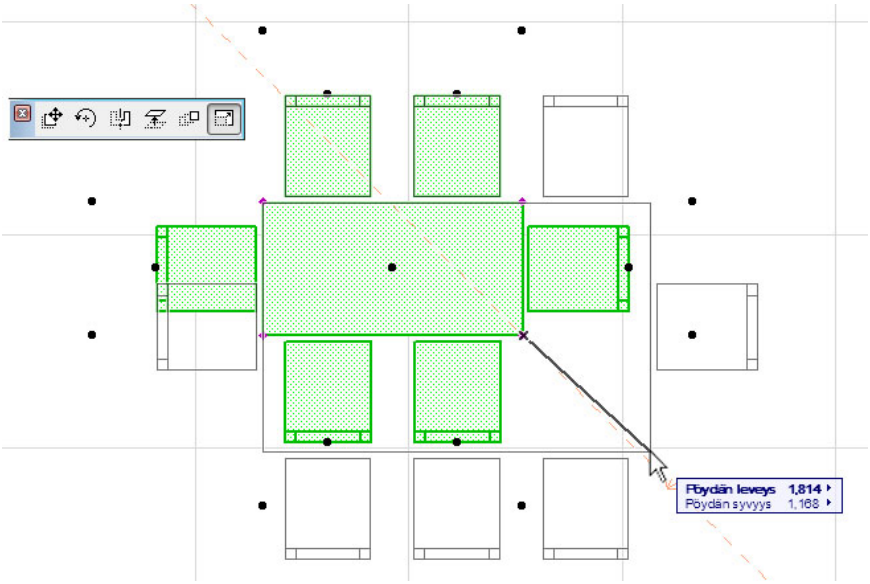
- ***Esikatselu ja sijoitus*** -valikossa objektille voidaan määritellä haluttaessa oletussijoituksesta poikkeava sijoituskorkeus ja -piste.
- ***Parametrit***-valikossa näkyvät käyttäjälle julkiset parametrit ja niiden oletusarvot, joita voidaan myös muuttaa.
- ***Pohja ja leikkaus*** -valikossa objektille määritellään 2D-piirrosten viivat, kynät ja leikkausmateriaalit.
- ***3D***-valikossa voidaan vaikuttaa 3D-näkymän materiaaleihin.
- ***Määrät ja selite*** -valikossa käyttäjä voi muokata määrätietueiden ja selitetekstien asetuksia sekä objektin yksilöllistä tunnistetta.
- ***Taso***-valikosta valitaan objektin sijoitustaso projektissa.

2.3.2 Graafinen 2D- ja 3D-käyttöliittymä

Suunnittelijan hyväksyttyä objektin asetukset käyttöliittymäikkunasta objekti on valmis sijoitettavaksi pohjaan 2D- tai 3D-suunnittelutilassa.

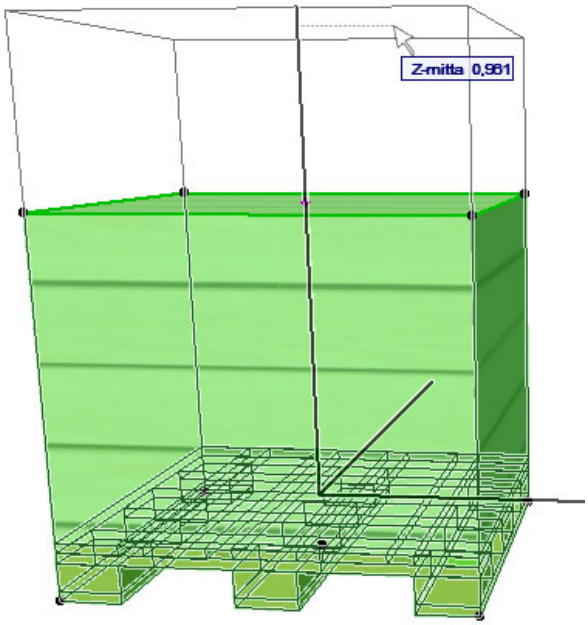
Objektin käyttöliittymäikkunan lisäksi objektin parametrien arvoja voidaan muokata graafisessa tilassa tartuntapisteiden avulla. Objektia voidaan kätevästi siirrellä, kiertää, peilata ja kopioida ja tietyissä tapauksissa myös skaalata. Tämän lisäksi, jos objektiin on ohjelmoitu dynaamisia tartuntapisteitä (venytettäviä

tartuntapisteitä), voidaan mitä tahansa objektin parametrin arvoa muuttaa tarttumalla hiirellä dynaamiseen tartuntapisteeseen ja siirtää sitä haluttuun suuntaan, jolloin parametrin arvo muuttuu GDL-ohjelmassa määritellyn logiikan mukaisesti.



Kuva 6. Pöytäryhmä-objektin 2D-käyttöliittymä

Kuvassa 6 käyttäjä on sijoittanut pöytäryhmän projektiin ja haluaa säätää pöytäryhmän leveyttä ja syvyyttä graafisesti 2D-tilassa. ArchiCADissa on valmiina useita eri toimintoja objektien muokkaamiselle, joista tässä tapauksessa on valittuna venytystoiminto. Käyttäjä tarttuu johonkin neljästä tartuntapisteestä ja siirtää tätä hiiren vasen nappi pohjassa haluamaansa suuntaan säätäen näin pöydän kokoa. Tuolit sijoitetaan ryhmään pöydän koon perusteella niin, että leveyden salliessa lisätään tuolien määrää.



Kuva 7. Kuormalava-objektin 3D-käyttöliittymä

Käyttäjä haluaa muuttaa luvussa 6 toteutetun kuormalava-objektin kuorman korkeutta kuvassa 7. Periaate on sama kuin 2D-tilassa, mutta nyt käytössä ovat myös z-akselin suuntaiset venytettävät tartuntapisteet.

2.4 Objektien toteutustavat

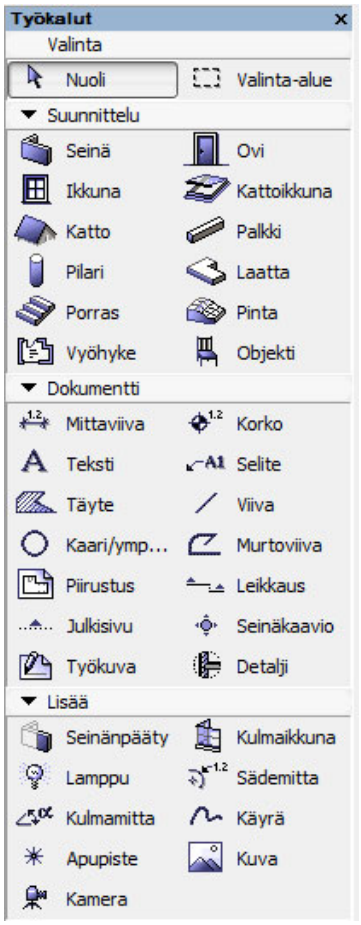
ArchiCAD mahdollistaa kaksi eri tapaa ja näiden yhdistelmän objektien toteuttamiseksi. Objektien geometriat voidaan toteuttaa mallintamalla tai ohjelmoimalla, mutta parametrisoidut ominaisuudet, ts. objektin älykkyys, tulee aina toteuttaa ohjelmoimalla.

2.4.1 Mallintaminen

Joissakin tapauksissa on perusteltua toteuttaa objekti mallintamalla eli piirtämällä ArchiCADin omilla tai lisäosina asennetuilla työkaluilla. Mallintamalla voidaan säästää aikaa objektin geometrioiden toteuttamisessa /27; 28/. Tämä sopii hyvin projektikohtaisesti käytettäville objekteille, joita ei tarvitse muokata tai uudelleen käyttää muissa projekteissa.

Mallintamisessa käytettävät geometriatyökalut löytyvät oletuksena ArchiCADin Työkalut-valikosta ruudun vasemmasta reunasta.

Mallintamisessa voidaan käyttää apuna myös taustakuvaa, pdf-dokumenttia tai 3D-mallista luotua leikkauskuvaa. Näin saadaan toteutettua suunnitelmaan tarkasti mitoitettuja objekteja ja rakenteita.



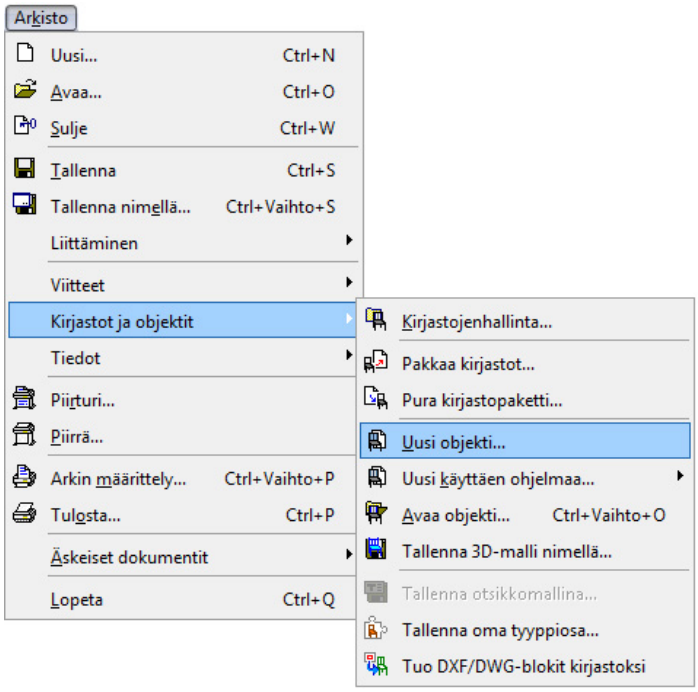
Kuva 8. Työkalut-valikko

Objektin sisältö on yleensä yksittäinen suunnitteluelementti, joka kokonaisuutena vastaa jotain reaalimaailman tuotetta tai rakennetta. Objekti voi olla myös vaikkapa kokonainen rakennus huonekaluineen ja ympäristöineen, mutta harvoin tällaiselle on perusteltua tarvetta.

2.4.2 Ohjelmointi

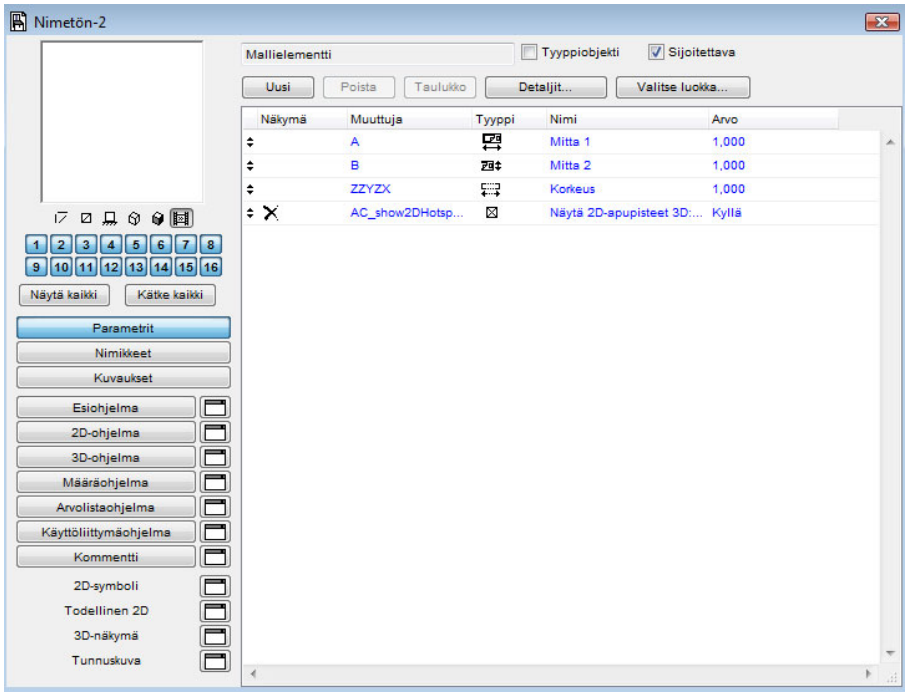
Toistuvaan käyttöön tulevat objektit kannattaa toteuttaa ohjelmoimalla, sillä näin objektille voidaan toteuttaa haluttu määrä älykkäitä ominaisuuksia. Ohjelmointi tehdään ArchiCADin omalla GDL-editorilla.

Uusi objekti luodaan avaamalla GDL-editori Arkisto → Kirjastot ja objektit → Uusi objekti – valikosta (kuva 9).



Kuva 9. Valikkopolku uuden objektin luomiseksi

Uuden objektin ohjelmointiympäristö, eli GDL-editori ja parametrit-näkymä avautuvat näkyviin (kuva 10).



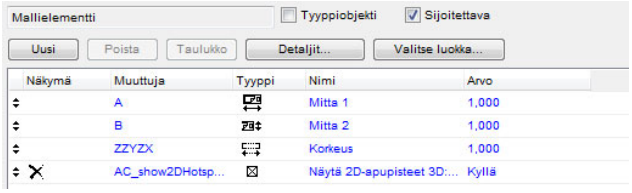
Kuva 10. ArchiCAD 11 GDL-editorin Parametrit-näkymä

2.4.2.1 GDL-editorin ominaisuudet

GDL-editori sisältää monta erillistä editoria ja toimintoa, jotka käsittelevät samaa tekstipohjaista gsm-päätteistä objektitiedostoa. Objektitiedosto sisältää objektin ohjelmalistaukset, parametrit ja niiden oletusarvot sekä mahdollista binääritietoa kuten esikatselu- ja muut kuvat.

GDL-editorin tärkeimmät toiminnot on esitetty lyhyesti taulukossa 1.

Taulukko 1. GDL-editorin tärkeimmät toiminnot

Toiminto	**Selite**
Näkymät 	Objektin 2D-symboli, 2D- ja 3D-näkymät ja tunnuskuva
2D-tasot 	Objektin 2D-näkymään määriteltyjen tasojen näkyvyyksien valinnat (GDL-ohjaus *fragment2*-funktiolla)
Parametrit 	Parametrien luominen, poistaminen, määrittäminen ja muokkaaminen
Nimikkeet	Määrälaskennassa käytettävien tietojen valinnat ja asetukset
Kuvaukset	Määrälaskentatietojen kuvaukset
Esiohjelma	2D- ja 3D-ohjelmia varten suoritettavat parametrien muokkaukset ja alustukset
2D-ohjelma	2D-geometriat ja toiminnot
3D-ohjelma	3D-geometriat ja toiminnot
Määräohjelma	Määrälaskennan toteutus

Arvolistaohjelma	Parametrien arvojen arvolistat, rajoitukset ja ehdot
Käyttöliittymäohjelma	Käyttöliittymäikkunan dialogisivun toteutus
Kommentti	Kommentteja esim. versiohistoriasta ja toteuttajista
2D-symboli	2D-symbolin esitys ja vaihtoehtoisesti valituilel tasoille piirtämällä toteutetun 2D-symbolin muokkausikkuna
Todellinen 2D	Symbolin ulkoasu sellaisena kuin se näkyy 2D-suunnittelutilassa
3D-näkymä	3D-geometrian, objektin globaalin origon ja ohjelmallisesti käsiteltävän paikallisen origon esitys
Tunnuskuva	128x128 pikselin kokoinen objektin tunnuskuva tiedostoselausta varten

2.4.3 Objektien vertailua eri toteutustapojen välillä

GDL-objektin toteutustavalla on merkitystä tiedostokokoon, käytettävyyteen ja ohjelmakoodin selkeyteen. Liitteen 2 taulukossa on vertailtu kahden toisiaan vastaavan yhden geometriakuvauksen sisältävän objektin ohjelmalistausta kahden eri toteutustavan välillä. Tiedostokoko mallinnetulla kuution muotoisella objektilla on 3,15 kt kun se vastaavalla ohjelmoidulla objektilla on vain 1,28 kt. Mallinnetun objektin tiedostokoon voidaan todeta olevan noin 2,5 kertaa suurempi kuin vastaavan ohjelmoidun objektin. Geometrioiden monipuolistuessa kasvaa näiden välinen tiedostokokojen ero entisestään ohjelmoidun objektin eduksi. Tiedostokoko ei kerro kuitenkaan objektin älykkyydestä vaan koodirivien määrästä. Objektin suoritusaika ArchiCADissä on riippuvainen myös ohjelman sisällä tehtävästä laskennasta ja toistorakenteista, jotka liittyvät pääasiassa ohjelmoituihin objekteihin. Mallinnetut objektit ovat ohjelmalliselta toteutukseltaan suoraviivaisia

ja niiden tiedostokoon ja suoritusajan välillä on selvempi yhteys kuin ohjelmoiduilla objekteilla.

2.5 Toteutustavan vaikutus käytettävyyteen

On tärkeää ymmärtää, miten toteutustapa vaikuttaa objektin käyttöominaisuuksiin, jotta osaa valita sopivan tavan kullekin objektille tai ainakin ymmärtää valintojensa merkityksen. Lisäksi on hyvä tietää että molemmille toteutustavalle on perustelunsa ja soveltuvat käyttötarkoituksensa.

2.5.1 Mallinnetut objektit

Yksinomaan mallintamalla toteutettujen objektien muunneltavuus rajoittuu ulottuvuuksien ja materiaalien yksinkertaiseen muokkaamiseen. Tämä voi riittää hyvin projektikohtaisten objektien toteutukseen, mikäli objekteja ei kopioida runsain määrin projektiin. Mallinnetut objektit sisältävät paljon toiminnan kannalta ylimääräistä tietoa, jonka prosessointiin kuluu aikaa. Mitä monimutkaisempi muoto on, sitä tehottomampaa mallintamalla tuotettu GDL-koodi on ja sitä kauemmin sen suorittaminen kestää. Objektin mitoilla ei ole merkitystä suoritusnopeuteen, vaan sen yksityiskohtien määrällä.

ArchiCADin peruskäyttäjälle mallintaminen on helpoin tapa toteuttaa tarvitsemansa objektit, joiden käyttötarve on akuutti ja projektikohtainen. Mallinnettaessa objektin toteutuksen tärkeimmät aspektit ovat geometria ja visuaalisuus.

Mallinnettua objektia voidaan parannella hybriditekniikalla, jolloin geometrian perusmuodot piirretään ArchiCADin suunnittelupohjassa origoa nollapisteenä käyttäen ja tallennuksen jälkeen muokataan 2D- ja 3D-ohjelmia niin, että eri mitoille, täytteille ja materiaaleille annetaan parametriarvot. Tämä vaatii kuitenkin ymmärrystä ohjelmien toiminnasta. Tätä tapaa voisi suositella vain monimutkaisten muotojen toteuttamiseen, sillä ohjelman muokkaaminen vie joka tapauksessa oman aikansa.

2.5.2 Ohjelmoidut objektit

GDL-teknologian olennaisin hyöty saavutetaan nimenomaan siinä, että objekteista voidaan tehdä älykkäitä, parametrisoituja ja helppokäyttöisiä. Kun objektin luominen toteutetaan ohjelmoimalla, on suunnittelijalla suora kontrolli sen jokaiseen ominaisuuteen ja siis mahdollisuus tuottaa tehokas ja älykäs objekti.

Toteuttamisen lähtökohtana on se, miten objektia halutaan käyttää ja muokata. Suunnittelussa voidaan käyttää esimerkiksi UML-kaavioita, jolloin tilaaja ja toteuttaja voivat yhdessä varmistaa objektin eri käyttötapaukset, tietosisällöt ja toimintatavat. Tilaajayrityksen liiketoiminta voi perustua tilattavan objektin älykkyyteen ja sen mahdollistamiin aika- ja kustannussäästöihin. Objektiteknologia voi tuoda todellista lisäarvoa tilaajayritykselle monilla tavoilla.

Projekteissa tarvitaan toisinaan objekteja, jotka toistuvat erittäin usein samoissa tai eri projekteissa. Tällöin objektin tehokkaalla koodilla on suurikin merkitys. Lisäksi tämä mahdollistaa sen, että useiden objektien parametreja muutetaan yhdellä kertaa. Objekteille voidaan määritellä myös parametreja, jotka säilyttävät joka tapauksessa oman yksilöllisen arvonsa, vaikka muut arvot muutettaisiin vastaavien objektien kanssa yhteneväisiksi.

Ohjelmoimalla objekteista voidaan tehdä mitoiltaan tarkkoja ja rakenteeltaan monipuolisempia kuin mallintamalla. Kappaleiden säännönmukaisuudet ja osien toistuvuudet voidaan toteuttaa lyhyilläkin toisto- ja ehtolauseilla.

Objektien renderointi on nopeampaa, sillä ohjelmoitu GDL-koodi on selkeämpää ja tehokkaampaa prosessoida.

Käytettävyyden kannalta suurin merkitys ohjelmoidun objektin toteutuksessa on se, että toteutuksen lähtökohtana voidaan käyttää objektimallia. Objektimalli voi sisältää esimerkiksi monipuolisen ja hyväksi todetun käyttöliittymäratkaisun. Objektimalleista on kerrottu tarkemmin luvussa 5.

3 OBJEKTIEN OHJELMOINTI

GDL-ohjelmointi perustuu neljän eri osa-alueen ymmärtämiseen. Osa-alueet on esitetty taulukossa 2.

Taulukko 2. GDL-ohjelmoinnin neljä osa-aluetta /17/

Geometriset funktiot GDL-kielen tarjoamat geometriset perusfunktiot, joita käyttämällä voidaan muodostaa erilaisia muotoja, joiden yhdistelmistä objektit koostuvat	**Objektien 3-ulotteisuus** Reaalimaailman kolmiulotteisten objektien analyyttinen tutkiminen ja GDL-toteutukseen tarvittavien muotojen ja parametrien määrittely
Ohjelmointi Ohjelmointikielen lauserakenteen ja syntaksin hallinta	**Käyttöliittymä** Objektin ominaisuuksien ja käyttötavan toteutus ArchiCADin, projektin ja käyttäjän tarpeita vastaavasti

Lisäksi ohjelmoitujen objektien toteutuksissa tulee kiinnittää huomiota tiettyihin perusperiaatteisiin, oli tarkastelijana sitten toteuttaja tai tilaaja:

- Objektin tyyppi tulisi määrittää oikein ja käyttää mahdollisuuksien mukaan standardisoituja parametreja
- Ohjelmakoodin rakenteen tulisi olla selkeää, jotta sitä voidaan yleisesti tarkastella, korjata, päivittää ja laajentaa
- Objektilla täytyisi olla kommentoitu versiohistoria päivämäärineen ja tekijöineen
- Renderointiajan optimointi tulee toteuttaa mm. polygonien määrää hillitsemällä ja valitsemalla käytännöllisimmät geometriafunktiot
- Automaattinen esitystarkkuuden säätö mittakaavan mukaan
- 3D-rautalankamallin tulisi olla sisällytettynä objektin toteutukseen piirturitulostusta varten
- Mittojen venytys tulisi mahdollistaa, jos sellaiset sallitaan
- Sallitut ja kielletyt arvot tulisi määrittää objektin toimivuuden varmistamisen kannalta kaikissa tilanteissa
- Tartuntapisteiden sijainnit ja looginen toiminta tulisi toteuttaa objektin asettelun ja säätämisen helpottamiseksi
- Parametri-ikkunan parametrit tulisi sijoitella hierarkisesti omiin kategorioihinsa

- Tilannekohtaisesti tulisi mahdollisesti toteuttaa objektin räätälöity käyttöliittymädialogi
- Objektille tulisi määritellä 128x128 pikselin esikatselukuva tiedostovalintaa varten
- 2D-symbolin toteutus tulisi toteuttaa itsenäisenä ohjelmana tai tasopiirroksena ohjelman resurssien säästämiseksi
- 2D-kynien, -viivojen ja -täytteeiden tulisi vastata standardisoituja väri- ja kuviokoodeja
- Materiaalien määrittäminen tulisi toteuttaa niin, että objektien siirrettävyys eri versioiden ja maiden välillä olisi mutkatonta
- Objektien tulisi toimia eri mittajärjestelmissä
- Objektikirjastot tulisi organisoida objektikategorioidensa mukaan käyttöönoton helpottamiseksi
- Objektin sijoitusorigoksi tulisi valita piste, jonka suhteen mittoja venytetään. /17/

Näiden toimintaperiaatteiden noudattaminen on lähtökohtana yleisesti hyväksyttyjen ja käyttökelpoisten objektien toteutukselle.

3.1 GDL ohjelmointikielenä

GDL-ohjelma vastaa rakenteeltaan ja syntaksiltaan BASIC-kieltä, jolloin perusasioiden omaksuminen on helppoa verrattuna esimerkiksi AutoCADin AutoLISP-kieleen, joka muistuttaa enemmän C++-kieltä. GDL-kielen opetusta järjestetään erittäin vähäisissä määrin Suomen rakennus- tai tietotekniikka-alan oppilaitoksissa. M.A.D. Oy järjestää GDL-kursseja pääasiassa ArchiMAD-kerholaisille, mutta myös kaikille halukkaille korvausta vastaan. GDL-kielen itseopiskelua varten on olemassa englanninkielisiä oppaita, kuten lähteissä mainitut GDL Cookbook 4 /17/ ja GDL Reference Manual /18/, mutta objektimallien luomisesta ja käytöstä ei ole löytynyt julkista kirjallisuutta tai dokumentaatiota.

Lähes kaikissa ohjelmointikielissä on oma syntaksinsa kommenttien sisällyttämiseksi ohjelmakoodiin. Kommentointi on tärkeää dokumentaatiota koodin tarkastelun ja seuraamisen helpottamiseksi. GDL-ohjelmakoodia kommentoitaessa käytetään koodirivillä huutomerkkiä ennen varsinaista kommenttia.

3.2 Parametrit

Parametrit eli objektin muuttujat ovat tärkeässä asemassa objektin älykkäässä toteutuksessa. Niiden avulla voidaan hallita objektin ominaisuuksia ja tehdä siitä näin tehokas suunnittelun apuväline.

Parametrien arvoja ja ominaisuuksia ohjaa ensisijaisesti arvolistaohjelma, josta on kerrottu tarkemmin kappaleessa 3.8. Seuraavana hierarkiassa ovat käyttäjän antamat parametrien arvot. Tämän jälkeen parametreja käsitellään vielä esiohjelmassa (luku 3.3), jonka jälkeen niiden arvot ovat määritelty 2D-, 3D- ja määrälaskentaohjelmaa varten.

3.2.1 Globaalit parametrit

ArchiCAD-projektin yhteiset parametrit, joita voidaan käyttää objektin toiminnan ja ominaisuuksien ohjaamiseen, tarjoavat tietoa muun muassa seinästä, johon ikkuna-/oviobjekti on liitetty tai käytetystä 2D-suunnittelutilan mittakaavasta.

Globaalien parametrien tarjoamista käyttömahdollisuuksista voisi ottaa esimerkiksi käyttäjän sijaintiin perustuvat objektin tilan muutokset. Näin voidaan luoda vaikkapa itsestään aukeavia ovia, kun kamera-ajo (katsontapisteen koordinaatti) lähestyy ovea animaatiossa.

3.2.2 Vakioparametrit

Normaalisti jokaisella objektilla on vähintään X-,Y- ja Z-mitat, joille on annettu parametrinimitykset A, B ja ZZYZX. Yleinen hämmennyksen aihe on se, miksi Z-mitalle on annettu näin kummallinen nimi, mutta sille löytyy inhimillinen selitys. Jo olemassa olleen Z-nimisen parametrin vuoksi täytyi Graphisoftin antaa vaihtoehtoinen parametrinimi, ja erään työntekijän ehdotuksesta nimi valittiin Kaliforniassa sijaitsevan paikkakunnan, Zzyzxin, mukaan. Lisäksi objekteilla on vielä yksi vakioparametri, joka kertoo, näytetäänkö 2D:ssä määritellyt tartuntapisteet 3D-tilassa.

Erikoistapauksessa objekti voi olla valonlähde, jolloin sen parametrit sisältävät tietoa väristä ja intensiteetistä.

3.2.3 Objektikohtaiset parametrit

Objektin tyypin ja olemuksen mukaan sillä voi olla lukematon määrä erilaisia parametreja. Osa näistä parametreista voi olla käyttäjän itse luomia, mutta monesti suurin osa on valmiiksi määritelty objektimallin tai objektityypin kautta.

3.3 Esiohjelma

Arvolistaohjelman ja käyttäjän antamien arvojen, sekä globaalien muuttujien perusteella voidaan eri parametrien ja niiden arvojen välille luoda laajojakin riippuvuuksia ja toimintalogiikoita.

Esiohjelma suoritetaan ennen 2D- ja 3D-ohjelmia, joten esiohjelman jälkeen molemmilla ohjelmilla on käytettävissään samat parametriarvot. Tämä takaa sen, että 2D- ja 3D-ohjelmat eivät ole riippuvaisia toisistaan, mutta voivat soveltaa samoja loogisia ominaisuuksia, kuten ehto- ja toistolauseita.

Esimerkki. *Parametrilla ”mitat” on arvonaan ”100 x 150”, missä lukuarvo tarkoittaa millimetrejä, ja halutaan eritellä arvot parametreihin A ja B. Alla on esitetty esiohjelmassa toteutettu ratkaisu esimerkkiin.*

```
nn = split(mitat, "%n x %n",a_arvo,x_merkki,b_arvo)
if nn = 3 then
            parameters a = a_arvo/1000
            parameters b = b_arvo/1000
endif
```

3.4 3D-ohjelman sisältö

Objektin materiaalit, geometriat ja toimintalogiikka 3D-tilassa toteutetaan 3D-ohjelmassa. 3D-ohjelman parametrit määritellään arvolistaohjelman, parametriarvojen ja esiohjelman kautta mainitussa järjestyksessä.

Suositeltavaa on aloittaa objektin toteutus 3D-ohjelmassa ja muokata valmiin koodin pohjalta 2D-geometriat poistamalla z-ulottuvuus.

3.4.1 Koordinaatisto

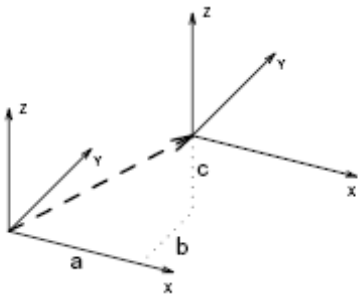
Objektin sijoitus, mitat, eri osien suhteelliset sijainnit ja ääriarvot sisältävät aina tiedon koordinaatiston sijainnista, akselien suunnasta ja mittakaavasta. Koordinaatiston periaatteellinen hallinta on tärkein edellytys objektien toteutuksille.

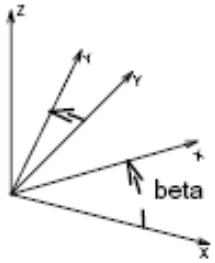
Koordinaatistoa voidaan käsitellä neljällä eri tavalla: lisäämällä arvoa eri akselien suuntaisesti, kääntämällä koordinaatistoa halutun akselin ympäri, muuttamalla akselien mittakaavojen kertoimia ja käsittelemällä koordinaatiston muokkauspinoa.

Kulloinenkin sijainti ja suunta määrittyvät origosta alkavien kumulatiivisin askelten perusteella. Jokainen koordinaatiston muokkauskäsky lisätään pinoon aina edellisen muokkauskäskyn edelle.

Taulukossa 3 on esitelty koordinaatiston muokkauksen tärkeimmät komennot.

Taulukko 3. 3D-koordinaatiston muokkauskomennot /19/

3D-koordinaatistofunktio(t)	**Selite**
ADDX-, ADDY- ja ADDZ-arvo tai ADD x,y,z 	Siirtymä akseli(e)n suuntaisesti mittakaavan mukaan.

ROTX-, ROTY- ja ROTZ-kulma 	Koordinaatiston kierto valitun akselin ympäri. Esim. *ROTZ beta* kiertää koordinaatistoa ylhäältä katsottuna vastapäivään *beta* astetta.
MULX-,MULY- ja MULZ-arvo tai MUL x,y,z	Muuttaa mittakaavan kerrointa. Esim. *MULX -1* peilaa x-arvot x-akselin vastakkaiselle puolelle.
DEL askelmäärä **(DEL TOP siirtyy origoon)**	Siirtyy askelmäärän verran takaisinpäin aikaisempaan sijaintiin. Esim. Rivi_1: *ADDX A* ! 1. askel Rivi_2: *ROTZ 15* ! 2. askel Rivi_3: *DEL 2* ! Siirrytään takaisin kahden askeleen verran.

3.4.2 Geometriset 3D-funktiot

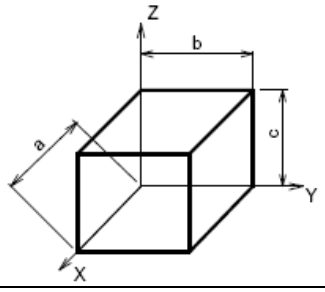
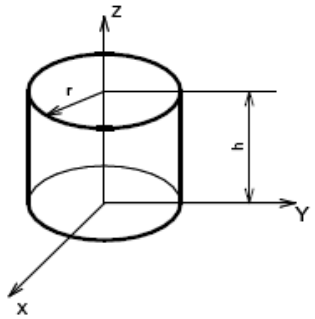
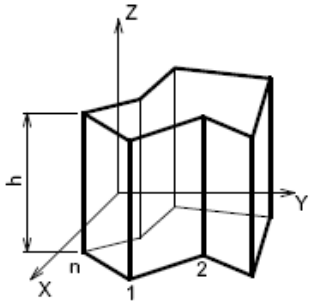
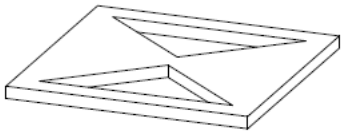
Objektin voidaan kuvitella koostuvan rakennuspalikoista, joilla on omat muokattavat perusmuotonsa. Näitä rakennuspalikoita yhdistelemällä oikealla tavalla voidaan koota monimutkaisiakin objekteja samaan tapaan kuin toiset leikkivät Lego-palikoilla. Kun käytössä on rajattu määrä erilaisia rakennuspalikoita eli geometrisiä funktioita, on oleellista tuntea niiden ominaispiirteet ja parametrit siten, että osaa koota näistä toivotun mukaisen objektin.

Käytettäessä eri palikoille eri materiaaleja, ne täytyy määritellä ennen graafista funktiota, ellei funktio itsessään sisällä materiaalimäärityksiä. Materiaalit kannattaa parametrisoida, tai kannattaa käyttää vakioituja materiaaliparametreja kuten *gs_frame_mat* -parametria. Materiaali määritellään ohjelmassa seuraavasti:

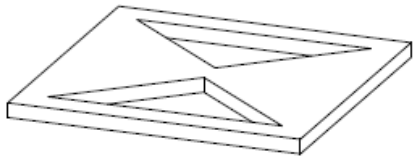
MATERIAL gs_frame_mat

Erilaisista geometrisistä funktioissa on tarkat kuvaukset GDL Reference Guidesta, joista on esitetty muutamia perusmuotoja kuvaavat funktiot ja niiden parametrit taulukossa 4.

Taulukko 4. 3D-geometriafunktioita /19/

3D-geometriafunktio	Selite
BLOCK x, y, z 	Kuutio, jolle annetaan mitat xyz-akseleiden suhteen.
CYLIND h, r 	Sylinteri, jolle annetaan korkeus *h* ja säde *r*
PRISM n, h, x1, y1, ...xn, yn 	Prisma, jolle määritellään kulmien määrä *n*, korkeus *h* ja kulmien xy-koordinaatit. *n >= 3*
CPRISM_ top_mat, bot_mat, side_mat, n, h, x1, y1, s1,… xn, yn, sn 	Prisma, jolle määritellään materiaalit eri pinnoilla, kulmien määrä n, korkeus h, kulmapisteden xy-koordinaatit ja pisteiden statusarvot *s* (Liite 3). Kts. kuva 11.

Piirrettävään monikulmioon voidaan leikata reikiä piirtämällä ensin peruskappaleen muodot soveltuvalla prismafunktiolla, jonka alkupisteeseen palaavassa loppupisteessä annetaan statukseksi *s = -1*, jolloin seuraava määritelty piste aloittaa uuden muodon piirtämisen, kunnes jälleen määritellään statukseksi *s = -1*. Niiltä osin kuin uusi muoto leikkaa aikaisempaa muotoa, suoritetaan boolean toiminto, joka leikkaa päällekkäisiin kohtiin reiän. Kuvassa 11 on tästä havainnollinen esimerkki.



```
CPRISM_ "Iron", 0, T_,              !"Iron" is a predefined material.
                                    !0 is a general material.
                                    !T_ is a global variable
                                    !(a material index)
    13, 0.2,
    0, 0, 15,
    2, 0, 15,
    2, 2, 15,
    0, 2, 15,
    0, 0, -1,                       !end of the contour
    0.2, 0.2, 15,
    1.8, 0.2, 15,
    1.0, 0.9, 15,
    0.2, 0.2, -1,                   !end of first hole
    0.2, 1.8, 15,
    1.8, 1.8, 15,
    1.0, 1.1, 15,
    0.2, 1.8, -1                      !end of second hole
```

Kuva 11. Cprism-funktio ja status-arvon käyttäminen reikien määrittelemiseksi /19/

Piirrettäessä objektia, joka koostuu useasta geometrisestä peruspalikasta, täytyy aina siirtyä koordinaatistossa oikeisiin palikoiden sijoituskohtiin. Tätä varten on suositeltavaa suunnitella etukäteen toisiinsa liittyviä geometriakokonaisuuksia, jotka voidaan sijoittaa vaikkapa samaan aliohjelmaan. Aliohjelmassa suoritetaan geometriset kuvaukset ja koordinaatiston siirrot. Palattaessa aliohjelmasta palautetaan myös koordinaatisto takaisin kutsua edeltäneeseen pisteeseen. Tähän kannattaa käyttää muuttujaa, jossa lasketaan aliohjelmassa tapahtuneet koordinaatiston siirrot, jolloin virheiden mahdollisuus vähenee esimerkiksi silmukkarakenteiden yhteydessä.

```
if piirra_palikat then gosub 1000
...
1000: ! palikanpiirto-aliohjelma
     palautus = 0 ! alustetaan laskurimuuttuja
     for i = 1 to 12
          block a/2, b/2. zzyzx/2
          ! kierretään koordinaatistoa ja lisätään siirrosmäärää
          ! laskurimuuttujaan 'palautus'
          rotz 30 : palautus = palautus + 1
          ! siirrytään x-akselilla eteenpäin a:n mitan ja lisätään
          ! siirrosmäärää
          addx a : palautus = palautus + 1
     next i
     del palautus ! siirrytään takaisin sijaintiin, jossa oltiin ennen
                       ! aliohjelmaa
    palautus = 0
return
```

3.4.3 Staattiset tartuntapisteet

Objektille täytyy määritellä staattiset tartuntapisteet, jotka auttavat objektin sijoittamisessa haluttuun kohtaan koordinaatistoa ja suhteessa toisiin objekteihin. Tartuntapisteet sijoitetaan pääasiassa objektin äärikulmiin, jolloin vältytään sijoittamasta objektia vahingossa osin sisäkkäin toisen objektien kanssa.

Tartuntapisteitä käytetään myös objektin referenssipisteinä, kun objektia esimerkiksi siirretään, kopioidaan, kierretään tai peilataan. A-, B- ja ZZYZX-mittojen ja akselien 0-kohtien leikkauspisteisiin toteutetaan useimmiten tartuntapisteet, jotka mahdollistavat automaattisesti näiden parametrien graafisen muokkaamisen ellei sitä ole estetty.

Staattisen tartuntapisteen määrittäminen on yksinkertaista ja toteutetaan seuraavalla 3D-ohjelmassa funktiolla:

HOTSPOT x, y, z [, unID : unID = unID + 1]

Objektin vakioparametrin *AC_show2DHotspotsIn3D* arvon ollessa tosi, luodaan automaattisesti 2D-ohjelmassa toteutetut tartuntapisteet 3D-tilassa objektin z-akselin 0-tasolla.

3.4.4 Dynaamiset (tai venytettävät) tartuntapisteet

Periaatteessa kaikkien parametrien muokkaaminen 3D-suunnittelutilassa on mahdollista venytettävien tartuntapisteiden avulla. Tämän on tarkoitus helpottaa ja nopeuttaa objektin muokkaamista käyttäjän tarpeita vastaavaksi.

Yhteen pituus-tyyppiseen parametriin vaikuttavan venytettävän tartuntapisteen määrittäminen vaatii kolmen erityyppisen tartuntapisteen määrittämistä muokattavalle parametrille valitulla akselilla:

- parametrin arvon 0-piste, tyyppi 1 (+128 piilottaa pisteen, vrt. binääriarvo 10000001b)
- parametrin referenssipiste 0-pisteen negatiivisella puolella, tyyppi 3 (aina piilotettu)
- venytettävä tartuntapiste, tyyppi 2

Alla on kuvattu X- ja Z-akselin suuntaisten venytettävien tartuntapisteiden määrittäminen.

Taulukko 5. X-akselin suuntaisen venytettävän 3D-tartuntapisteen määrittäminen ja parametrit

HOTSPOT	***x,***	***y,***	***z,***	***tunniste,***	***muuttuja,***	***tyyppi : tunniste++***
HOTSPOT	**x_0,**	y,	z,	unID,	**x_muuttuja,**	1+128: unID=unID+1
HOTSPOT	**x_{ref},**	y,	z,	unID,	**x_muuttuja,**	3 : unID=unID+1
HOTSPOT	**x,**	y,	z,	unID,	**x_muuttuja,**	2 : unID=unID+1

```
! Korkeuden säätö, z-akseli
HOTSPOT a, b, 0,     unID, zzyzx, 1+128  : unID = unID + 1          !BASE
HOTSPOT a, b,-1,     unID, zzyzx, 3      : unID = unID + 1          !REFERENCE
HOTSPOT a, b, zzyzx, unID, zzyzx, 2      : unID = unID + 1          !MOVEABLE
```

Yhden kulma-tyyppisen venytettävän tartuntapisteen määrittely vaatii neljän pisteen määrittelyä:

- kulman keskipiste, CENTER (x_{ac}, y_{ac}, z_{ac})
- kulmaparametrin 0-piste, BASE (x_0, y_0, z_0)
- kulmaparametrin referenssipiste, REFERENCE (x_{ref}, y_{ref}, z_{ref}), joka keskipisteen kanssa määrittää kierrettävän tason akselin
- kulman venytettävä tartuntapiste, MOVEABLE (x,y,z)

Taulukko 6. Kulma-tyyppisen venytettävän 3D-tartuntapisteen määrittäminen ja parametrit

HOTSPOT	***x,***	***y,***	***z,***	***tunniste,***	***muuttuja,***	***tyyppi : tunniste++***
HOTSPOT	**x_{ac},**	**y_{ac},**	**z_{ac},**	unID,	**kulma,**	6 : unID=unID+1
HOTSPOT	**x_0,**	**y_0,**	**z_0,**	unID,	**kulma,**	4+128: unID=unID+1
HOTSPOT	**x_{ref},**	**y_{ref},**	**z_{ref},**	unID,	**kulma,**	7 : unID=unID+1
HOTSPOT	**x,**	y,	z,	unID,	**kulma,**	5 : unID=unID+1

```
! objektin z-akselin kierto origon ja x-akselin pisteen mukaan
hotspot 0, 0, 0,     unID, kulma, 6      : unID = unID + 1          !CENTER
hotspot a, 0, 0,     unID, kulma, 4+128  : unID = unID + 1          !BASE
hotspot 0, 0, zzyzx, unID, kulma, 7      : unID = unID + 1          !REFERENCE
rotz kulma                                                          !rotation
hotspot a, 0, 0,     unID, kulma, 5      : unID = unID + 1          !MOVEABLE
```

3.5 2D-ohjelman sisältö

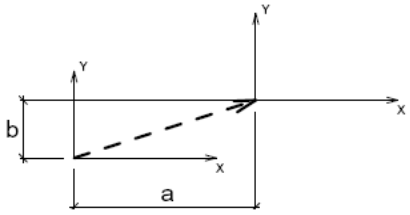
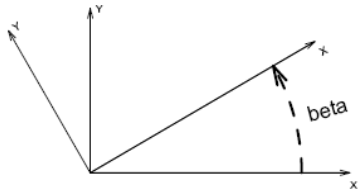
Objektin piirrossymbolin geometriat, piirtokynät, täytteet ja toimintalogiikka 2D-tilassa toteutetaan 2D-ohjelmassa. 3D-ohjelman parametrit määritellään arvolistaohjelman, parametriarvojen ja esiohjelman kautta mainitussa järjestyksessä.

Kuten kappaleessa 3.4 todettiin, 2D-ohjelma on järkevää toteuttaa valmiin 3D-kuvauksen pohjalta poistaen siitä z-ulottuvuus ja korvaamalla 3D-funktiot vastaavilla 2D-funktioilla.

3.5.1 Koordinaatisto

Koordinaatisto toimii 2D-tilassa kuten 3D-tilassakin, lukuun ottamatta z-ulottuvuutta. Koordinaatiston muokkausfunktiot ovat hieman erimuotoisia, mutta toimivat samalla periaatteella.

Taulukko 7. 2D-koordinaatiston muokkauskomennot /19/

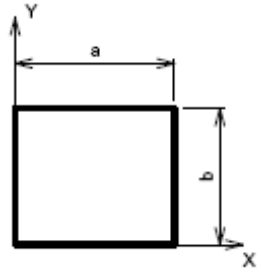
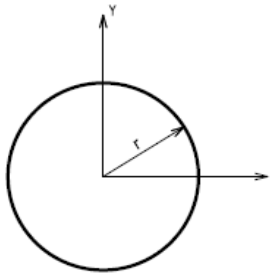
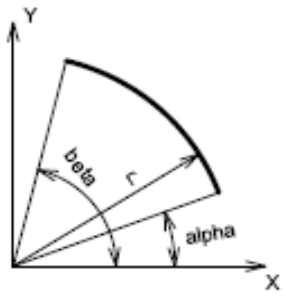
2D-koordinaatistofunktio(t)	**Selite**
ADD2 a, b 	Koordinaatiston xy-siirtymä mittakaavasta riippuen.
ROT2 kulma 	Koordinaatiston xy-akselien kierto. Esim. *ROT2 beta* kiertää xy-koordinaatistoa ylhäältä katsottuna vastapäivään *beta* astetta.
MUL2 x,y	Muuttaa mittakaavan kerrointa. Esim. *MUL2 -1,0* peilaa x-arvot x-akselin vastakkaiselle puolelle.
DEL askelmäärä **(DEL TOP siirtyy origoon)**	Siirtyy askelmäärän verran takaisinpäin aikaisempaan sijaintiin. Esim. Rivi_1: *ADD2 A,B* ! 1. askel Rivi_2: *ROT2 15* ! 2. askel Rivi_3: *DEL 2* ! Siirrytään takaisin kahden askeleen verran.

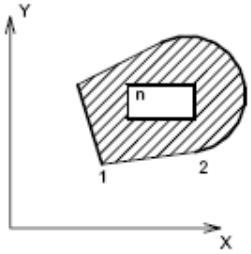
3.5.2 Geometriset 2D-funktiot

GDL-kielestä löytyy monipuoliset funktiot 2D-piirtämiseen ja usein funktioilla on vastineensa 3D-puolella. 2D-symbolin kuvaus perustuu 3D-projektioon objektin yläpuolelta katsottuna. Projektion luomiseksi on *project2*-funktio, jota ei kuitenkaan suositella käytettävän ohjelman toiminnan hidastumisen vuoksi.

2D-symboli on pohjapiirustukseen näkyviin tuleva kuva, jonka täytyy olla yhtenäisen käytännön mukainen ja selkeä. Tästä syystä 2D-symboli toteutetaan erillisenä, mutta soveltuvuuden rajoissa voidaan käyttää pohjana 3D-ohjelman funktioita, jotka korvataan vastaavilla 2D-funktioilla ja poistetaan geometrioiden päällekkäisyydet. Tapauksesta riippuen saattaa olla tarvetta piirtää symboli halkileikkauskuvaksi, josta näkyy rakenteen sisusta ylhäältä katsottuna.

Taulukko 8. 2D-geometriafunktioita /19/

2D-geometriafunktio	**Selite**
RECT a,b 	Suorakulmio, jonka sivujen pituudet ovat *a* ja *b*. Vrt. 3D: *BLOCK a, b, zzyzx*
CIRCLE r 	Ympyrä, jonka säde on *r*. Vrt. 3D: *CYLIND h, r*
ARC r, alpha, beta 	Kaari, jonka säde on *r*, alkukulma *alpha* ja loppukulma *beta*.

POLY2_ n, frame_fill, x1, y1, s1, ... xn, yn, sn  	Monimuotoinen kuvio, missä *n* on kulmapisteiden määrä, *frame_fill* täytteen attribuutit, *xn* ja *yn* kulmapisteiden koordinaatteja ja *sn* kulmapisteen statusarvo. *n>=2*. Esim. *frame_fill = 1+2+4* on täytetty ja suljettu kuvio. Vrt. 3D: *PRISM_ n, h, x1, y1,s1,...xn, yn, sn*
POLY2_B{2} n, frame_fill, fill_pen, fill_bg_pen, x1, y1, s1, … xn, yn, sn	Monimuotoinen kuvio, kuten edellä, mutta sisältää lisäksi täytteen kynän (*fill_pen*) ja täytteen taustakynän (*fill_bg_pen*) määrittelyt. Vrt. 3D: *CPRISM_ top_mat, bot_mat, side_mat, n, h, x1, y1, s1, ... xn, yn, sn*

3.5.3 Staattiset tartuntapisteet

Staattinen tartuntapiste mahdollistaa objektin ulkoiset funktiot – siirtämisen, peilaamisen, kiertämisen ja kopioimisen. Staattisia tartuntapisteitä määritellään vain niihin kohtiin, jotka ovat tarkoituksenmukaisia objektin ulkoisten ominaisuuksien muuttamisessa, jotka liittyvät objektin sijaintiin ja suuntaan.

Taulukko 9. Hotspot2-funktiolla luodut staattiset tartuntapisteet

HOTSPOT2 x , y, unID	x = x koordinaatti origosta, y = y koordinaatti origosta
HOTSPOT2 0 , 0, unID	koordinaatiston origo (0,0)
HOTSPOT2 A , B, unID	koordinaatiston origon vastainen kulma etäisyydellä (origo + A, origo + B)
HOTSPOT2 A/2, B/2, unID	piste koordinaatiston origosta, joka on (origo+A/2, origo+B/2) etäisyydellä

3.5.4 Dynaamiset (tai venytettävät) tartuntapisteet

Muuttujien arvojen graafisen muokkaamisen mahdollistavien venytettävien tartuntapisteiden määrittäminen vaatii useiden erilaisten pisteiden määrittelyitä – kolme jokaista 2D-tilan kulma- tai pituusmuuttujaa kohden. 2D-tilan kulma-tyypin muuttuja ei tarvitse kierrettävän tason ja sen akselin määrittävää pistettä, kuten 3D-tilassa.

Muuttujat xVar ja yVar ovat objektin parametrilistassa määriteltyjä, tässä tapauksessa pituus-tyyppisiä parametreja. Muuttujat toimivat niin, että kun

dynaamista tartuntapistettä siirretään graafisesti, tallennetaan muuttujiin niiden uusi sijainti x_0,y_0 pisteeseen verrattuna.

Alla esitellään pituus-tyyppisen dynaamisen tartuntapisteen määrittäminen kahdelle muuttujalle 2D tilassa. Muuttujan *xVar*-arvo muuttuu sen perusteella, kuinka paljon ja mihin suuntaan tartuntapistettä siirretään x-akselilla. Samoin muuttujan *yVar* arvo riippuu siitä, kuinka paljon ja mihin suuntaan tartuntapistettä siirretään y-akselilla.

Määritellään tarvittavat kuusi pistettä, jotka muodostavat kaksi venytettävää tartuntapistettä (taulukko 10).

Taulukko 10. Venytettävien 2D-tartuntapisteiden määrittäminen ja parametrit

HOTSPOT2	**[x]**	**[y]**	**[tunniste]**	**[Muuttuja]**	**[tyyppi] : [tunnisteNro++]**	**[kommentti]**
HOTSPOT2	x,	y_0,	unID,	yVar,	1 + 128 : unID=unID+1	!BASE (I)*
HOTSPOT2	x,	y_{ref},	unID,	yVar,	3 : unID=unID+1	!REF (I)
HOTSPOT2	x,	y,	unID,	yVar,	2 : unID=unID+1	!MOVE (I)
HOTSPOT2	x_0,	y,	unID,	xVar,	1 + 128 : unID=unID+1	!BASE (II)*
HOTSPOT2	x_{ref},	y,	unID,	xVar,	3 : unID=unID+1	!REF (II)
HOTSPOT2	x,	y,	unID,	xVar,	2 : unID=unID+1	!MOVE (II)

*Tyyppi 1: [1] tekee BASE- eli 0-pisteen ja [+128] piilottaa sen näkyvistä

Taulukko 11. Hotspot2-koordinaattimerkintöjen selitteet.

x	Sijainnin x-koordinaatti
y	Sijainnin y-koordinaatti
x_0	X-akselin muuttujan **xVar** 0-kohta (A – parametrille normaalisti = 0)
y_0	Y-akselin muuttujan **yVar** 0-kohta (B – parametrille normaalisti = 0)
x_{ref}	Referenssipiste, jonka täytyy olla muuttujan nollapisteen negatiivisella puolella (oletus -1)
y_{ref}	Referenssipiste, jonka täytyy olla muuttujan nollapisteen negatiivisella puolella (oletus -1)
xVar	X-akselin muuttuja
yVar	Y-akselin muuttuja

3.6 Määrälaskentaohjelma

Objektin parametreja voidaan hyödyntää myös projektin määrälaskennassa toteuttamalla objektille määrälaskentaohjelma, jossa voidaan muun muassa laskea määriä tarvittavilla kaavoilla ja lisätä objektin symboli, tunnus ja muita tietoja määrälaskentaraporttiin.

Taulukko 12. Määrälaskentaohjelman yleisimmät käskyt

Komento	**Parametrit**	**Selite**
descriptor	nimi [, koodi, avain]	Nimike
component	nimi, määrä, yksikkö	Lukumäärä ja laskentayksikkö
surface3d()		Funktio, joka laskee objektin pintojen pinta-alan
volyme3d()		Funktio, joka laskee objektin tilavuuden
drawing		Lisää kuvan objektin symbolista määrälaskentaan
database_set	nimi	Tietokannan määrittely
binaryprop		Määrittelee käytettäväksi objektin nimikkeistön ja kuvaukset, jotka on sisällytetty objektiin

3.7 Käyttöliittymäohjelma

Käyttöliittymäohjelmassa toteutetaan käyttäjäystävällinen parametrien muokkaustapa. Käyttöliittymään voidaan luoda dialogisivu, johon puolestaan voidaan sisällyttää useita välilehtiä. Tärkeimmät komennot ja niiden oleellisimmat parametrit käyttöliittymän ohjelmoimiseksi on esitelty alla olevassa taulukossa.

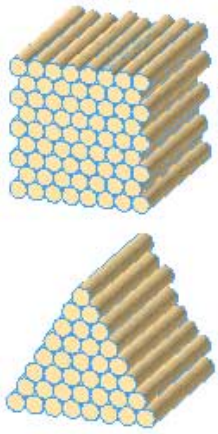
Taulukko 13. Käyttöliittymäohjelman yleisimmät komennot

Komento	**Parametrit**	**Selite**
ui_dialog	otsikko [,x_koko, y_koko]	Dialogisivun määrittely
ui_page	sivun numero	Sivunumeron määrittäminen
ui_current_page	sivun indeksi	Aktiivisen sivun numero
ui_button	tyyppi, selite, x,y,leveys,korkeus	Toimintopainike
ui_style	fonttikoko (0..2), tyyli (0..16)	Tekstin koko ja tyyli
ui_outfield	seliteteksti, x,y,x_koko,y_koko	Selitetekstikenttä, sisältö ja sijainti
ui_infield	”parametri”, x, y, x_koko, y_koko	Syöttökenttä valitulle parametrille ja sijainti
ui_infield	”parametri”, x, y, x_koko, y_koko, metodi (1..4), ”kuvan_nimi”, kuvien_lkm, rivien_lkm, solu_x, soly_y, kuva_x, kuva_y, kuva_indeksi_1, ”seliteteksti 1”, kuva_indeksi_2, ”seliteteksti 2”..	Graafinen valintakenttä, jossa käyttäjä voi valita objektin ulkomuodon visuaalisesti. Kyseiselle valinnalle tehdään toteutus, joka antaa objektille valintaa vastaavat parametrien arvot.
ui_pict	”kuvan_nimi”, x, y tai indeksi, x, y	Havainnollistavan kuvan liittäminen käyttöliittymään. Kuva tulee tallentaa joko objektikirjastoon tai suoraan objektiin binäärimuotoisena (indeksiviittaus).

Graafiset valintalistat edustavat edistyneempää tapaa toteuttaa objektin parametrien asetusten säätäminen. Näiden toteuttaminen on työlästä, mutta lisää huomattavasti parametrien valintojen havainnollisuutta. Objektin käyttöliittymään täytyy toteuttaa kuvatiedosto, jossa objektin rakenteen vaihtoehdot on esitettyinä /20/. Taulukossa 14 on esitetty tästä toteutustavasta yksinkertainen esimerkki.

Taulukko 14. Esimerkkikoodi graafisesta valikkolistasta ja valikon kuvalähde

```
ui_dialog "Puupinon asetukset"

ui_style 1,1

ui_outfield "Puupinon malli",5,20, 100,20

ui_infield "pinomodel", 5,40,138,128,
                                1, "puupinot.jpg",
                                2, 2, 128, 128, 128, 128,
                                1, "Suora pino",
                                2, "Vino pino"

if pinomodel = "Suora pino" then parameters pinomalli = 1
if pinomodel = "Vino pino" then parameters pinomalli = 0

end
```



”puupinot.jpg”
128x256

3.8 Arvolistaohjelma

Objektin parametreille voidaan määritellä ehtoja, rajoituksia, arvolistoja ja resoluutioita niin, että objekti tuntee omat rajansa. Näin voidaan määritellä esimerkiksi standardimittavaihtoehdot, ääriarvot tai muita objektin käyttöä helpottavia asetuksia.

Taulukko 15. Arvolistaohjelman yleisimmät komennot /19/

Komento	**Parametrit**	**Selite**
values	”parametri” `arvo_1`,...,`arvo_n`	Arvolistan määritys
values	”parametri” *range*[alkuarvo, loppuarvo]	Ääriarvojen määritys
values	”parametri” *step* nollapiste, askellus	Resoluution määritys
lock	”parametri”	Parametrin asetuksen lukinta käyttäjältä
hideparameter	”parametri”	Parametrin piilottaminen parametrilistasta

4 OBJEKTIEN SUUNNITTELU

Suunnittelun lähtökohtana on objektin tuleva käyttötarkoitus. Lisäksi on syytä muistaa luvussa 3. esitellyt objektien hyvän tavan mukaiset toteutusperiaatteet. Tässä luvussa perehdytään tarkemmin siihen, mitkä eri suunnittelumenetelmät sopivat parhaiten GDL-objektien suunnitteluun.

4.1 Suunnitteluprosessi

Yksittäisten objektien toteutus vie useimmiten aikaa vain tunneista muutamaan päivään ja objektikirjastojen toteutus objektien määrän ja vaativuuden mukaan päivistä kuukausiin /20/. Nopeatempoinen toteutusprosessi asettaa omat vaatimuksensa suunnittelun osuudelle. Hyvin toteutettu suunnittelu voi viedä joskus turhankin paljon aikaa, joten käytännön toteutus on usein prosessi, jossa suunnittelu kulkee käsi kädessä toteutuksen kanssa. Toisaalta hiemankin monimutkaisemman ja älykkäämmän objektin toteutuksessa suunnittelu on hyvin tärkeää tehdä pitkälle valmiiksi etukäteen, jotta toteutuksessa ei tule seinä vastaan toivottujen ominaisuuksien osalta.

Suunnittelun lähtökohtana ovat tilaajan toiveet, joiden pohjalta pyritään tekemään määrittelyt eri ominaisuuksien osalta. Tilaajan tehdessä toimeksiannon kokonaisen objektikirjaston toteutuksesta tulee kirjastolle määrittää yleiset ja objektikohtaiset määrittelyt.

Toteuttajan tulee ymmärtää objektin käyttötarkoitus jopa paremmin kuin tilaajan silloin kun tilaajalla ei ole riittävää näkemystä GDL:n mahdollisuuksista ja rakennussuunnittelijan tarpeista. Tämä asettaa haasteita määrittely- ja suunnitteluprosessille, joita tarkastellaan lähemmin seuraavissa kappaleissa.

4.1.1 Objektien käyttötarkoitukset

Objektit voidaan jakaa kuuteen eri luokkaan käyttötarkoituksen mukaan:

- 2D-symboli
- 3D-kappale
- makro
- tyyppiobjekti
- tieto-objekti
- 2D-sovellus

2D-symboli sisältää objektin piirrosmerkin toteutuksen, jolloin suunnittelun kannalta olennaisinta on selvittää sovellettavat piirrosmerkkistandardit, objektin toivotut parametrit ja looginen käyttäytyminen käyttäjän ja ohjelmiston toimesta.

3D-kappale on reaalimaailman esine, jonka osien geometriat, ulottuvuudet ja materiaalit sisällytetään objektin toteutukseen. Suunnittelu perustuu geometrioiden analyyttiseen tutkimiseen ja loogisen toiminnan määrittämiseen. Lisäksi selvitetään tilaajan visuaaliset vaatimukset, joiden perusteella tehdään päätöksiä materiaalien tekstuureista. Materiaalivaatimukset vaikuttavat myös geometriafunktioiden valintaan.

Makro, eli kutsuttava objekti, on muiden objektien käytettävissä oleva yleiskäyttöinen toteutus 2D-, 3D- tai määrälaskentaohjelmasta. Makro-objektin peruslähtökohtana on jonkin perusratkaisun jakaminen useiden objektien käyttöön, jolloin yksittäisistä objekteista tulee selkeämpiä ja kooltaan pienempiä. Makro suunnitellaan pääasiassa toteutuksensa käyttötarkoitustyypin mukaisesti, mutta suunnitteluvaiheessa on huomioitava, että makro täytyy sisällyttää aina sitä käyttävään objektikirjastoon. Makroa kutsuvaa objektia ei voi tarjota yksittäisenä objektina esimerkiksi online-tuotekuvaston kautta, sillä osa sen toteutuksesta jää puuttumaan, mikäli käyttäjä ei lataa myös kutsuttua makro-objektia.

Tyyppiobjektin suunnittelun lähtökohtana on esiin tullut tarve toteuttaa useita samat perusparametrit sisältäviä objekteja. Tyyppiobjekti ei sellaisenaan sisällä itse

ohjelmakoodia. Sitä käytetään vain parametrien yhdenmukaistamiseen samankaltaisten objektien kesken.

Tieto-objektin käyttötarkoituksia voi olla useita, mutta suunnittelun lähtökohtana on tarve yhdistää ja tutkia projektissa esiintyvää tietoa tähän tehtävään ohjelmoidun objektin kautta. Tietoa voidaan esittää 2D-tilassa, josta se voidaan siirtää esimerkiksi plansseille tai kootusti määrälaskentataulukkoon. Tieto-objektiin voi toteuttaa myös materiaali-, viiva, täyte tai muita vastaavia määrityksiä.

2D-sovellus on tieto-objektin erikoistus, jolloin ArchiCADin 2D-tilaan voidaan toteuttaa projektin sisältämää tietoa havainnollistava sovellus. Sovelluksen graafinen käyttöliittymä on 2D-geometrian ja objektin asetusikkunan yhdistelmä. Suunnittelumenetelmäksi soveltuu esimerkiksi käyttäjätapausten määrittely.

4.1.2 Ketterä ohjelmistokehitys

Yhä nopeammin muuttuvan yritysmaailman tarpeisiin tarvitaan muutoksiin nopeasti mukautuvia toimintatapoja. Ketterien menetelmien soveltaminen ohjelmistoprojekteissa on todettu lisäävän tehokkuutta ja vähentävän kehityskustannuksia jopa 70 %. Tämä parantaa yrityksen kilpailukykyä vastata markkinoiden kysyntään, hintakilpailuun ja muuttuviin haasteisiin. /21/

Ketterään toimintaan liittyvät karakterisoinnit /22/:

- kontekstiin liittyvät: todellisuus voittaa teorian, menetelmien valinta tilanteen mukaan
- ihmisiin liittyvät: Yksilöllinen osaaminen, dynaamiset roolit, vuorovaikutus
- tekemisen kohteeseen liittyvät: asiakaslähtöisyys, kohteen yhteinen omistajuus
- toimintatyyliin liittyvät: tavoitehakuisuus, luovuus, suunnittelu asioiden edetessä

M.A.D. Oy:n kokemuksen mukaan GDL-objektiprojektit ovat luonteeltaan nopeatempoisia ja ketterät menetelmät ovat luonnollisin vaihtoehto käytettäväksi

projektimalliksi. GDL-ohjelmoijien osaamistasoissa on eroja, joten dynaamiset työnjaot ja roolit huomioidaan objektin toteutuksen vaativuuden perusteella.

Objekteille ei ole yhtä ainoaa toteutustapaa vaan toteutukset vaihtelevat projektien kesken. Tavoitteena on valita kuhunkin projektiin sopivin toteutus tilanteen mukaan. Eri yrityksissä ja eri maissa on toisistaan poikkeavia vakiintuneita käytäntöjä, jotka tulee ottaa huomioon projekteissa asiakaslähtöisyyden periaatteiden toteutumiseksi.

Tilaajan ja toteuttajan välillä voi toisinaan olla näkemyseroja, joiden selvittäminen ei onnistu aina määrittelyvaiheessa. Joskus objektien toteutus ja suunnittelu kulkevat käsi kädessä siten, että tilaajalle toimitetaan useita välivaiheiden objekteja kommentoitavaksi, jolloin määrittelyt täsmentyvät asteittain.

4.1.3 Asiakaslähtöisyys

Mielekkään ja kannattavan GDL-objektiprojektin päämääränä on aina tarjota käyttöominaisuuksiltaan tilaajan tarpeeseen sopiva objektituote. GDL-objektin tilaaja ei useinkaan ole objektin loppukäyttäjä eli suunnittelija vaan mahdollisesti projektista yleisesti vastaava henkilö.

Käyttöliittymäsuunnittelija Karri-Pekka Laakson (Reactor Innovations Oy) mukaan tilaajalle sovelluksen, tässä yhteydessä objektin lisäarvo on nimenomaan siinä, kuinka tehokkaasti ja tuottavasti objektia voidaan hyödyntää suunnittelussa. Asiakkaan tarpeet ja toiveet, objektin käyttötarkoitus ja ominaisuudet kannattaa määritellä yhdessä asiakkaan kanssa, jolloin objektin tavoitetila on selvillä jo ohjelmoinnin alkaessa. /23/

4.1.3.1 GUIDe – mallin mukainen käyttöliittymäsuunnittelu

Tilaajan ollessa suunnittelun ammattilainen ja mahdollinen loppukäyttäjä voidaan käyttää esimerkiksi GUIDe04-mallin mukaista käyttöliittymäsuunnittelumallia. Vaatimuksena GUIDe-mallin soveltamiselle on se, että tilaaja pystyy esittämään tarkat toiveet ja esitiedot toteutettavalle objektille. /23/

Määrittelyn tehtävänä on selvittää tilaajalta seuraavat asiat:

- mitä parametreja ja tietoja objekti käyttäjälleen näyttää
- miten objektin parametrit ja tietosisältö organisoidaan ja visualisoidaan käyttöliittymässä
- mitä toimintoja objekti tarjoaa käyttäjälleen eri käyttöliittymien kautta
- miten objekti reagoi kun käyttäjä muokkaa parametreja.

GUIDe-malli täydentää hyvin ketterien menetelmien määrittelyvaiheen aukkoja:

- käyttötilanteet kuvaavat käyttäjän ongelman eivätkä lainkaan ratkaisua
- yksikäsitteinen ja suoraviivainen ratkaisu em. ongelmiin
- ratkaisu esitetään kuvina ja kuvasarjoina, joita lukijan on helppo ymmärtää
- käyttöliittymäratkaisun laatii suunnittelija, ei asiakas
- käyttötilanteet selvitetään ensisijaisesti järjestelmän tulevilta käyttäjiltä, ja asiakkaan rooli on ohjata suunnittelijat oikeiden käyttäjien luo tutkimaan heidän työtään
- ratkaisun oikeellisuutta ja kattavuutta voidaan testata jo ennen toteutusta. /24/

GUIDe-malli on näennäisesti ristiriidassa ketterien menetelmien manifeston kanssa. Ohjelmoinnin toteutustyön pitäisi periaatteessa alkaa heti projektin ensimmäisistä hetkistä lähtien, kun GUIDe-mallin periaatteen mukaisesti ensin suunnitellaan käyttöliittymä ja sitten vasta siirrytään toteutukseen. Ketterien menetelmien tarkoituksena on kuitenkin minimoida tarpeetonta dokumentaatiota ja tehostaa työn tuottavuutta. Käyttöliittymän määrittely ja suunnittelu ohjelmoimalla on huomattavasti kalliimpaa kuin kynällä paperille piirretyt käyttöliittymäkuvat, ja lisäksi se tuottaa helposti huonoja ratkaisuita /23; 24/.

4.1.3.2 Objektien suunnittelu käytännössä

GUIDe-mallin mukainen käyttöliittymäsuunnittelu ei käytännössä aina sovellu sellaisenaan GDL-objektiprojekteihin. Tämä johtuu siitä, että monessa tilanteessa objektin toteutuksen tilaaja ei ole sen loppukäyttäjä. Tilaajan ollessa esimerkiksi markkinointipuolesta vastaava henkilö, hän ei välttämättä tunne tilaamansa objektin rakennetta, erityisominaisuuksia tai käyttötapaa niin tarkkaan, että siitä

olisi apua objektin määrittelyssä. Tilaaja voi esimerkiksi olla halukas tarjoamaan suunnittelijoille tuotteistaan online-tuotekatalogin ja markkinoida tuotteitaan jo suunnitteluvaiheessa eri rakennusprojekteihin. Tästä johtuen objektien suunnittelu ja toteutus voi olla joskus käytännössä erittäin haasteellista ja vaatii sopivia lähestymistapoja. Lopputuloksen täytyy kuitenkin tyydyttää sekä asiakasta että loppukäyttäjää. Käyttöliittymän suunnittelussa suureksi avuksi on PK11-perusobjektin käyttöliittymä, jota muokkaamalla ja täydentämällä saadaan aikaiseksi suomalaiselle käyttäjälle jo valmiiksi tutun oloinen käyttöliittymä. Kyseistä käyttöliittymää sovellettaessa tärkein asia on selvittää, mitä osia ja muokattavia parametreja objektin tulee sisältää sekä varmistaa objektin looginen toiminta hierarkioineen.

Suunnitteluvaiheessa kannattaa epävarmat tilanteet ratkaista parametrisoimalla ne ja antamalla jokin oletusarvo, joka voidaan tarvittaessa jättää vakioksi. Tämä ei juurikaan lisää toteutukseen kuluvaa aikaa, mutta voi säästää sitä, jos tilaaja toteaakin tarvitsevansa määrittelyvaiheessa avoimeksi jääneitä parametreja. Näin myös objektien päivittäminen pitkällä aikavälillä helpottuu, jos huomataankin tarve muokata parametrein jotain aikaisemmin vakiolta vaikuttanutta ominaisuutta.

4.2 Toimeksianto ja vaatimukset

Objektien toteutus M.A.D. Oy:ssä perustuu joko peruskirjaston objektien toteuttamiseen ArchiCAD-ohjelman käyttäjien toiveiden perusteella tai erillisten asiakkaiden toimeksiantoon.

Toteutettavan objektin osalta tärkeimmät vaatimukset määräävät, mitkä mitat ja ominaisuudet parametrisoidaan käyttäjän tarpeisiin muokkautuviksi. PK11-perusobjektimallissa on valmiiksi määriteltynä käyttöliittymän rakenne, jota pyritään noudattamaan toteutettaessa uutta objektia. Lisäksi vaatimukset voivat määrittää objektissa käytetyt pintamateriaalit, tekstuurit ja oletussijoitusorigon x-, y- ja z-suunnissa.

4.3 Standardit

Rakennusalan standardit ovat erittäin moninaisia riippuen osa-alueesta. Sähkö-, LVI- ja arkkitehtisuunnittelussa on omansa, kuten myös valmistajapuolella esimerkiksi rakennemitat ja menettelytavat. Suunnittelijan täytyy selvittää objektia koskevat standardit ja yleiset käytännöt, mikäli niillä on merkitystä toteutuksen ja käytön kannalta. Tietoa eri standardeista löytyy internetistä, kirjastoista, rakennusosavalmistajilta ja –maahantuojilta tai mm. Suomen Standardisoimisliitosta. Osa objekteista perustuu malli- ja yrityskohtaisiin ratkaisuihin, jolloin tälle täytyy toteuttaa suunnittelijalle mahdollisuus antaa vapaat arvot standardien lisäksi.

4.4 Käyttölogiikka

Objektin älykkyyden ja käyttäjäystävällisen toiminnan kannalta on olennaista selvittää, miten objektin tulisi reagoida ympäristöönsä tai käyttäjän määrittelemien parametrien eri arvoihin.

Käytön ja toteutuksen kannalta on kiinnitettävä huomiota objektin sijoittumisesta koordinaatistoon. Objektin paikalliseksi origoksi on suositeltavaa valita piste, jonka suhteen mittoja pääasiassa muutetaan ja jonka on tarkoitus sijaita pohjakuvassa paikallaan. Muussa tapauksessa voidaan joutua toteuttamaan monimutkaisia koordinaatiston muokkauksia, jolloin objekti pysyy manuaalisesti ja graafisesti mittoja venytettäessä tarkoituksenmukaisella paikallaan.

4.4.1 Käyttötapaukset

Monipuolisen käyttöliittymän toteuttamiseksi voidaan aluksi määritellä käyttötapaukset, joihin objektin täytyisi kyetä muuntautumaan.

Käyttötapaukset voidaan jaotella seitsemään ryhmään:

- Älykkäät käyttörajoitukset eli objektin toimintarajat
- Mittakaavaan mukautuminen
- Valinnaiset rakennevaihtoehdot
- Mittavaihtoehdot
- Graafinen muokkaus
- Manuaalinen muokkaus
- Ympäristön vaikutus

Älykkäät käyttörajoitukset antavat objektille rajat, joiden puitteissa objekti on käytettävissä. Objekti siis opetetaan ymmärtämään omat rajansa. Näin objektia käyttävän suunnittelijan virheiden mahdollisuus vähenee.

Mittavaihtoehdot määrittelevät eri osien mahdolliset mitat ja mitta-askeleet käyttörajoitusten puitteissa. Mitta-askel voi perustua esimerkiksi objektin toteutuksessa käytettävien perusosasten vakiomittoihin. Näin käyttäjä voi automaattisesti varmistua siitä, että objekti voidaan toteuttaa valituilla perusosilla rikkomatta niiden vakiomittoja.

Valinnaiset rakennevaihtoehdot antavat käyttäjälle mahdollisuuden valita, mitä varusteita tai rakenteita objektissa käytetään. Rakennevaihtoehdot vaikuttavat usein myös käyttörajoituksiin ja mittavaihtoehtoihin, jolloin ne muodostavat tapauskohtaisen hierarkian. Näiden väliset riippuvuudet tulee selvittää ja toteuttaa käyttäjäystävällisesti.

Mittakaava on tärkeä osa rakennussuunnittelua ja objektin esitystarkkuuden tulee myös mukautua mittakaavan muutoksiin niin, että käyttäjän ei välttämättä tarvitse tätä erikseen valita.

Käyttäjän ensimmäinen kosketus objektiin on sen muokkaaminen käyttöliittymädialogin kautta manuaalisesti. Tämän tulee olla selkeä ja nopea käyttää, mutta vaihtoehtoja toteutukseen on useita. On määriteltävä toteutetaanko objektille kaikki parametrien muokkaukset parametrit-ikkunassa vai tehdäänkö käyttöliittymään oma dialogisivu objektin parametreille mahdollisine ohjekuvineen ja valintalistoineen.

Graafinen muokkaus on objektin jatkomuokkaamista käyttöliittymäikkunassa määriteltyjen arvojen jälkeen. On määriteltävä, mitä ominaisuuksia käyttäjä haluaa säätää tartuntapisteiden avulla ja mitkä tartuntapisteet ovat olennaisia objektin sijoittelun kannalta.

4.4.2 Algoritmien suunnittelu

Algoritmien avulla määritellään objektin rakentumislogiikka eri osien summana. Objekti jaetaan erillisiin osiin, jotka kasataan tiettyjen sääntöjen mukaisesti. Nämä säännöt täytyy selvittää ja suunnitella paras toteutus niitä noudattavalle objektin rakenteelle.

Objektin eri osaset sijoittuvat toisiinsa nähden sääntöjen mukaisille paikoille, ja jokaiselle osaselle muodostetaan sijainti joko origon tai toisten osasten kiintopisteiden suhteen. Suositeltavaa on valita selkeitä ja varmasti tiedossa olevia kiintopisteitä, jotka eivät mittoja skaalatessa vääristä osasten sijaintia. Tästä syystä objektin origo on paras kiintopiste. Jos voidaan olla varmoja osasten välisestä sijoittumisesta, on mahdollista käyttää näiden yhteisiä kiintopisteitä.

Objektin käyttäytymiselle parametrien suhteen määritellään tarvittaessa omat sääntönsä, joiden pohjalta voidaan suunnitella varsinainen toteutuslogiikka GDL-kielellä. Mitä dynaamisempia objekteja toteutetaan, sitä monimutkaisempia algoritmeista tulee.

Eri osaset kannattaa jaotella omiksi aliohjelmikseen, joita voidaan kutsua esimerkiksi toistorakenteiden sisällä määriteltyjen ehtojen mukaisesti. Dynaamisen rakenteen toteuttamiseksi voidaan käyttää taulukkotyyppisiä muuttujia, joiden indekseinä voidaan käyttää esimerkiksi toistolausekkeen laskureita.

4.4.3 Käyttöliittymä

Objektin parametrien määrittäminen tapauskohtaisesti täytyy toteuttaa käyttäjäystävällisesti, jolloin objektin käyttö on sujuvaa ja tehokasta. Objektilla on enintään kolme eri käyttöliittymää, jotka on esitelty kappaleessa 2.3.

Käyttöliittymäikkunassa ohjelmoija voi vaikuttaa parametrilistan muotoiluihin. Parametrit tulee jakaa omiin hierarkisiin kategorioihinsa, jotta käyttäjä tietää mitä asiaa hän kulloinkin on säätämässä. Parametrit, joiden muokkaus halutaan estää, tulee piilottaa näkyvistä.

Käyttöliittymäikkunaan voidaan toteuttaa oma dialogisivu, johon voidaan rakentaa vapaamuotoinen käyttöliittymä parametrien säätämiseksi havainnollisine kuvineen ja graafisine valintalistoineen. Tälle dialogisivulle voidaan lisätä myös www-linkkejä vaikkapa valmistajien sivuille. Yksilöidyn dialogisivun toteutus ilman valmista mallipohjaa vie suhteellisen paljon aikaa, mutta hyvin toteutettuna se parantaa objektin käytettävyyttä. Suositeltavaa olisi käyttää valmista objektimallipohjaa, jonka käyttöliittymään on toteutettu oma dialogisivu ja kytketty sen kentät olemassa olevan objektimallin parametreihin.

2D- ja 3D-käyttöliittymät perustuvat staattisiin ja venytettäviin tartuntapisteisiin. Staattiset tartuntapisteet sijoitetaan pääasiassa objektin A-, B- ja ZZYZX -mittojen kulmapisteisiin ja objektin sijoitusasettelun kannalta hyödyllisiin pisteisiin. Venytettäville tartuntapisteille määritetään aina kolme pistettä parametria kohden, jolloin järjestelmä tietää parametrin 0-kohdan, muutettavan asetusarvokohdan ja negatiivisen suunnan 0-kohtaan nähden. Venytettävien tartuntapisteiden kautta helpotetaan graafista parametrien säätämistä, mutta tilannekohtaisesti tulee päättää, mitä parametreja on perusteltua muokata graafisessa tilassa.

4.5 Parametrien määrittely

Objektin älykkyys perustuu nimenomaan sen parametreihin, joten on tärkeää määritellä mitä ominaisuuksia halutaan parametrisoida ja minkä tyyppisiä nämä parametrit ovat.

Objektin ominaisuus tulee parametrisoida, jos sitä tarvitsee joskus muuttaa, verrata tai käyttää raja-arvojen tai -ehtojen määrittämiseen.

Totuustyyppiset parametrit ovat käyttökelpoisia objektin valinnaisten osien ja rakenneratkaisuiden määrittelyssä. Totuustyypin parametrille täytyy antaa sen

toimintaa kuvaava selkeä seliteteksti parametrilistaan, jolloin käyttäjän ei tarvitse arvailla sen vaikutusta objektiin.

Pituustyyppiset parametrit määrittelevät objektin eri osien mittoja. Parametrisoidut mitat auttavat objektin eri osien koon suhteellisen sijoittelun toteutuksessa.

Kulmatyyppiset parametrit kuvaavat asteina objektin osien välisiä kulmia eri akseleiden suhteen.

Tekstityyppiset parametrit kuvaavat objektin ominaisuuksia tekstimuodossa, usein valmiiksi määriteltyjen arvolistojen avulla. Ne ovat käyttökelpoisia esimerkiksi eri standardien tai mallivaihtoehtojen valinnoissa sekä listattaessa objektille annettuja ominaisuuksia määrälaskennassa.

Parametri voi olla myös taulukkomuotoinen, jolloin indeksien avulla voidaan käsitellä parametrin arvoja esimerkiksi toistorakenteissa. Taulukkoparametrit ovat käteviä tarvittaessa paljon samantyyppisiä parametreja kuvaamaan objektin ominaisuuksia, kuten dynaamisten rakenteiden välisiä kulmia.

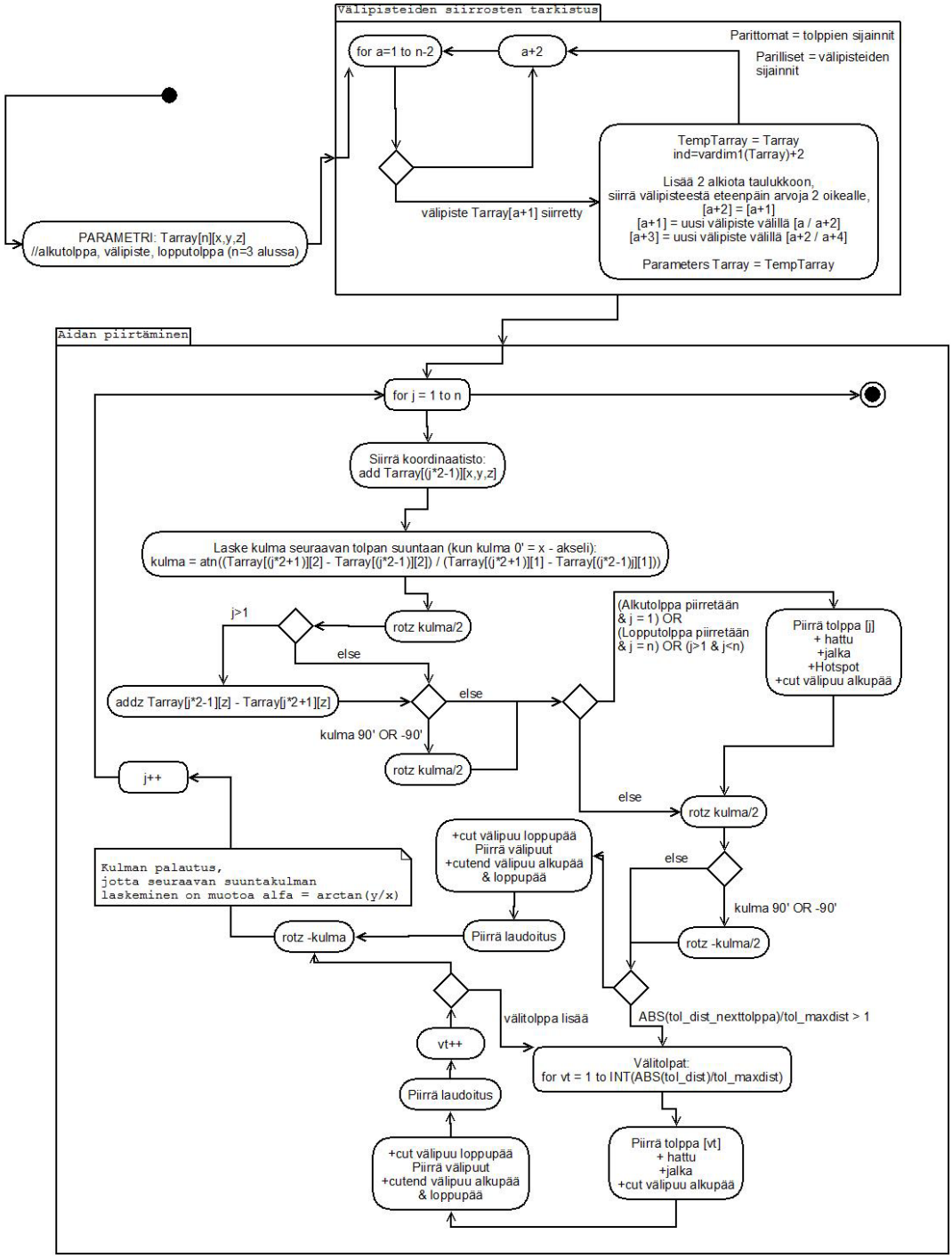
Lisäksi on joukko parametrityyppejä kuten viivoja, kyniä, täytteitä ja materiaaleja, jotka pitää useimmiten parametrisoida.

4.6 Toimintalogiikan UML-kuvaus

Objektin älykkyysvaatimusten noustessa kasvaa myös tarve suunnitella ja dokumentoida objektin toimintalogiikka etukäteen esimerkiksi UML-kaaviona.

Kuvassa 12 on esitetty dynaamisen aitaobjektin toimintalogiikka UML-kaaviona. Kyseisen objektin toiminta perustuu graafiseen käyttöliittymään, jolloin käyttäjä voi aitatolppien välissä olevaa venytettävää tartuntapistettä siirtämällä luoda uuden tolpan ja sijoittaa se haluamaansa kohtaan. Dynaamisesti luodun uuden tolpan molemmille puolille tehdään uudet venytettävät tartuntapisteet, joiden avulla voidaan tehdä jälleen lisää tolppia.

Monimutkaisten, kuten edellä mainitun dynaamisen aitaobjektin, toimintalogiikoiden toteutus vaatii onnistuakseen hyvää suunnittelua ja dokumentaatiota. Yksinkertaisten objektien suunnittelussa UML-kuvauksen laatiminen on tarpeetonta, sillä suoraviivaiset toiminnot on melko helppo toteuttaa ja ne dokumentoidaan kommentein suoraan objektin ohjelmakoodiin.



Kuva 12. Dynaamisen aitaobjektin UML-aktiviteettikaavio

5 PK11-OBJEKTIMALLI

Useissa eri ohjelmointikielissä on valmiita toteutusmalleja, jotka yksinkertaistavat usein käytettävien ominaisuuksien hyödyntämistä yleistämällä hyväksi katsottuja toteutustapoja. Näin useat käyttäjät voivat ymmärtää, omaksua ja käyttää samoja hyväksi todettuja toteutusmalleja riippumatta toisistaan.

GDL-objekteja on saatavilla maailmanlaajuisesti jo lukuisia, mutta ne ovat monesti hyvinkin erinäköisiä ja tekijänsä käsialan mukaisia. Tällöin objektien yhteiskäyttö ja päivittäminen tulevat työlääksi. Tämän ongelman ratkaisemiseksi M.A.D. Oy on vuosien aikana kehittänyt ja jalostanut perusobjektimalliaan niin, että sen tarjoamat perustoiminnot täydentävät kattavasti objektin älykkyyttä ja käyttöliittymä on kunkin peruskirjaston objekteilla yhdenmukainen. /6/

M.A.D. Oy:n suomenkieliset PK11-objektimallit ovat lähtökohtana suomalaisten ArchiCAD-pakettien mukana tulevien peruskirjastojen objektien toteutukselle. PK11-objektimalleissa on myös otettu huomioon maamme erityistarpeet. /6/

5.1 Objektimallin yleisiä määritelmiä

Vähimmäisvaatimuksena objektimalliin perustuvilla objekteilla on oltava tyypilleen ominaiset yhteiset parametrit. Tästä johtuen esimerkiksi ikkunalla on eri parametrit kuin vaikkapa ovella. ArchiCADistä löytyy valmiita objektimalleja, niin sanottuja tyyppiobjekteja, joita voi käyttää uusien objektien tyypille ominaisten parametrien luomisessa.

Tyyppiobjektilla on tapauksesta riippuen Graphisoftin määrittelemiä parametreja ja niitä täydentämässä IFC-standardin määrittelemiä parametreja. Nämä erottuvat selvästi toisistaan siten, että Graphisoftin parametrit alkavat tunnisteella ”gs_” ja IFC:n parametrit tunnisteella ”ifc_”.

Tarvittaessa objektimalliin voidaan sisällyttää parametrikohtaisia arvolistoja, rajoituksia, ehtoja ja resoluutioita, mitkä määritellään objektin arvolistaohjelmassa.

Parametrien esikäsittely 2D-, 3D- ja määrälaskentaohjelmaa varten suoritetaan esiohjelmassa, johon voidaan toteuttaa joukko objektimallille yhteisiä toimintoja.

2D-, 3D-, määrälaskenta- ja käyttöliittymäohjelmiin on mahdollista sisällyttää objektimallin tarkoitusta vastaavia yhteisiä ominaisuuksia.

Objektimallissa on hyvä olla enemmän valmiita parametreja ja ominaisuuksia, kuin mitä yksittäisen objektin toteutuksessa välttämättä tarvittaisiin. Yhteisiä parametreja ja ominaisuuksia on helpompi poistaa tai piilottaa kuin luoda uusia.

5.1.2 Objektimallien käyttämisen käytännön hyödyt

Vaikka objektimallin kehittäminen tyhjästä vie aikaa, sen käyttäminen säästää suunnitteluun ja ohjelmointiin kuluvaa aikaa, mikä on suoraan laskettavissa myös taloudelliseksi hyödyksi niin objekteja tuottaville kuin tilaavillekin yrityksille. Lisäksi objekteja on helppo päivittää seuraavia ArchiCAD-versioita varten, jolloin objektien elinkaari pitenee.

Objekteja tuottaville yrityksille yhteinen objektimalli mahdollistaa myös useiden eri ohjelmoijien yhteistyön sujuvuuden ja jatkuvuuden eri objektiprojektien kesken.

5.1.3 Jatkuvaa kehitystä

Graphisoft kehittää jatkuvasti ArchiCAD-ohjelmistoaan ja sisällyttää siihen uusia ominaisuuksia ja toimintoja. Tästä johtuen myös objektimalleja on syytä päivittää hyödyntämään uusia ominaisuuksia, jolloin uudet ominaisuudet saadaan nopeasti käyttäjien hyödynnettäväksi.

Objektimallin päivittäminen vaatii vähintään tallentamisen uuden ArchiCAD-version mukaiseksi, jotta voidaan varmistua siitä, että sitä voidaan käyttää uuden version objektien mallina. Tämä tapahtuu avaamalla objektimalli uudessa ArchiCADissä ja tallentamalla se uutena objektina, jolloin myös tiedoston versionumero päivittyy. Tämän jälkeen suoritetaan mallin testaus niin, että voidaan todeta kaikkien ominaisuuksien toimivan toivotulla tavalla. ArchiCAD-versiot ovat

laajalti yhteensopivia siirryttäessä vanhemmasta versiosta uudempaan. Tästä löytyy tarkempaa informaatiota Graphisoftin ja M.A.D. Oy:n nettisivuilta.

5.1.4 Johdanto PK11-objektimalliin

ArchiCAD 11-version uudet ominaisuudet muuttivat monien aikaisempien peruskirjastojen objektien käyttöluonnetta ja jopa tarpeellisuutta. Tietomallien yhteiskäytön tarpeen yleistyessä myös objektien tietosisällön yhdenmukaistaminen vaati parametrien nimitysten vakioimista niiltä osin, kuin oli mahdollista.

Suomenkielisen ArchiCAD 11-ohjelmiston käyttäjille suunnatut PK11-objektimallit ovat syntyneet näiden vaatimusten ja jatkuvan kehityksen myötä, pääasiassa Laurimark Oy:n Lauri Melvasalon ja M.A.D. Oy:n yhteistyön ansiosta.

PK11-objektimalli on jokaisen PK11 peruskirjaston objektin pohjana, jolloin kirjaston objektien yhteiskäyttö on mahdollisimman saumatonta ja toimivaa. PK11-objektimalli on ollut myös vapaassa käytössä myös ulkopuolisten kirjastojen objektitoteutuksille. /6/

5.2 Perusobjekti PK11

Perusobjekti PK11 on objektimalli, johon M.A.D. Oy:n peruskirjasto 11 objektit perustuvat. Perusobjektin lisäksi on olemassa myös perusikkuna-, perusovi- ja perusvalo-objektit, jotka ovat tyyppinsä mukaisia versioita perusobjektista niin, että näiden yhteiskäyttö olisi tehokasta yhtenevien parametrien ansiosta. /6; 31/
Tässä tutkintotyössä perehdytään nimenomaan perusobjektin toteutukseen ja ominaisuuksiin, sillä sen sisäistäminen auttaa muiden objektimallien ymmärtämistä periaatteellisen samankaltaisuutensa ansiosta. PK11-perusobjektin parametreista, ohjelmalistauksista ja niiden yhteyksistä kuvaava kaavio on liitteessä 4.

5.2.1 Parametrit

PK11-perusobjektilla on suuri joukko käytettävissä olevia parametreja moneen eri tarkoitukseen. Liitteestä 1 löytyy kaikki perusobjektin parametrit, niiden kategoriat ja tyypit.

Suurin osa PK11-parametreistä on jo valmiiksi sisällytetty käyttöliittymään, arvolistaohjelmaan ja määräohjelmaan. Osa parametreista liittyy objektin tietomalliin, jossa määritellään objektille yksilöllisiä ominaisuuksia, kuten esimerkiksi valmistaja, kantavuus ja syttymisluokka.

PK11-perusobjektin parametrit on esitetty liitteessä 1. Esimerkiksi *muoto*-parametri määrittelee piirretäänkö geometriat suorakulmaisen vai pyöreän perusmuodon mukaan, ja joille molemmille täytyy ohjelmoida oma toteutuksensa 2D- ja 3D-ohjelmissa.

5.2.2 Esiohjelma

Esiohjelmassa on tiettyjen PK11-parametrien perusteella tehtävät toiminnot, jotka liittyvät joko perusmuodon valintaan tai siihen, onko objektia kutsuttu makrona toisesta objektista.

Luotaessa omia parametreja ja määrittelemällä niiden arvot esimerkiksi arvolistaohjelmassa, voidaan esiohjelmassa tehdä tarkistuksia, alustuksia tai erotella tekstimuotoisesta arvosta numeeriset osat. Esiohjelmasta on täydellinen listaus liitteessä 5.

5.2.3 2D-ohjelma

2D-ohjelmassa toteutetaan graafisessa 2D-tilassa esitettävät geometriat ja tartuntapisteet sekä niiden toiminta. Ohjelmoinnin nopeuttamiseksi on toteutettu valmiita toimintoja, jotka suoritusjärjestyksessään ovat seuraavat:

1. Origon tartuntapiste ja *unID* indeksin alustus arvoon 201
2. Valinnainen origon piirtäminen toteutuksen avuksi
3. Koordinaatiston siirto origosijainti-parametrin määräämään kohtaan
4. Tartuntapisteiden sijoitus kulmiin ja keskelle A- ja B-parametrien mukaan
5. 2D-mittakaavan määrittäminen
6. Valinnainen tekstin tulostus
7. Valinnainen tilantarpeen esittäminen
8. 2D-kynien ja -täytteiden määrittäminen
9. Perusmuodon perusteella suoritettavien aliohjelmien valinta
10. Aliohjelma, johon toteutetaan perusmuotoon ja mittakaavaan sopiva geometrinen kuvaus toimintalogiikoineen.

Objektin tilavarauksen toteutus riippuu objektin geometriasta, joten sen toteutus tulee tarkastaa ja muokata asianmukaisesti. Tämä onnistuu tilanvarausfunktion koordinaatteja säätämällä.

Objektin valmistuttua turhat toiminnot voidaan joko kommentoida pois käytöstä tai poistaa kokonaan tiedostokoon pienentämiseksi ja koodin selkeyden parantamiseksi. 2D-ohjelmasta on täydellinen listaus liitteessä 5.

5.2.4 3D-ohjelma

3D-ohjelmassa toteutetaan graafisessa 3D-tilassa esitettävät geometriat ja tartuntapisteet sekä niiden toiminta. Rakenteeltaan ja toiminnaltaan se on hyvin samankaltainen 2D-ohjelman kanssa ottaen lisäksi huomioon korkeusmitan ja materiaalit. Objektimalliin toteutetut valmiit toiminnot suoritusjärjestyksessään ovat seuraavat:

1. Origon tartuntapiste ja *unID* indeksin alustus arvoon 301
2. Valinnainen origon piirtäminen toteutuksen avuksi
3. Koordinaatiston siirto origosijainti-parametrin määräämään kohtaan
4. Tartuntapisteiden sijoitus kulmiin ja keskelle A-, B- ja ZZYZX-parametrien mukaan
5. 3D-esitystarkkuuden määrittäminen
6. 3D-kynien ja -materiaalien määrittäminen

7. Perusmuodon perusteella suoritettavien aliohjelmien valinta
8. Aliohjelma, johon toteutetaan perusmuotoon ja mittakaavaan sopiva geometrinen kuvaus toimintalogiikoineen.

Objektin valmistuttua turhat toiminnot voidaan joko kommentoida pois käytöstä tai poistaa kokonaan tiedostokoon pienentämiseksi ja koodin selkeyden parantamiseksi. 3D-ohjelmasta on täydellinen listaus liitteessä 5.

5.2.5 Määrälaskentaohjelma

Objektimalliin on toteutettu valmiiksi toiminnot, jotka lisäävät objektin yleiset tiedot määrälaskentaan /31/. Useimmiten määrälaskentaohjelmaan ei tarvitse tehdä muutoksia. Jos objektin parametri ”laske” on tosi, tuotetaan objektin tiedot automaattisesti määrälaskentaan. Havainnollisuuden parantamiseksi viedään määrälaskentaan oletuksena muun muassa objektin 2D-symboli.

Lisättävät objektitiedot tuotetaan yksinkertaisesti seuraavalla listauksella:

```
descriptor ""+tyyppi, "001"
descriptor ""+tunnus, "002"
descriptor ""+nimi, "003"
descriptor ""+kuvaus, "004"

descriptor ""+str("%.0mm",A), "005"
descriptor ""+str("%.0mm",B), "006"
descriptor ""+str("%.0mm",ZZYZX), "007"
descriptor ""+katisyys, "008"
descriptor ""+tieto, "009"
descriptor ""+versio, "010"
```

Lisäksi tulostetaan rakennusosa- ja rakennustarvikeluokka, mikäli ne ovat määriteltyinä. Määrälaskentaohjelmata on täydellinen listaus liitteessä 5.

5.2.6 Käyttöliittymä

Olennainen osa PK11-objektimallin käyttöliittymää on ”Objektin asetukset” -dialogisivu käyttöliittymäikkunassa. Se sisältää kaikkien parametrien muokkaamiseksi tarvittavat kentät, selitteet ja toiminnot. Lisäksi dialogisivulla on www-linkki M.A.D. Oy:n kotisivuille. Yhdenmukaisen käyttöliittymätoteutuksen tavoitteena on parantaa objektien käytön sujuvuutta ja tuttuuden tunnetta /31/.

Käyttöliittymän tekstien ja kenttien asettelun määritys on tärkeä ymmärtää, joten tarkastellaan sijoittelun toimintalogiikkaa lähemmin.

Ensin määritellään vakiomitat riveille ja sarakkeille:

```
sarakelev=444/4 ! neljä saraketta
rivikor=20
outlev=sarakelev-5
outkor=rivikor-3
inlev=sarakelev-20
inkor=rivikor-3
```

Määriteltyjä vakiomittoja käytetään selitteiden ja kenttien asettelussa seuraavasti:

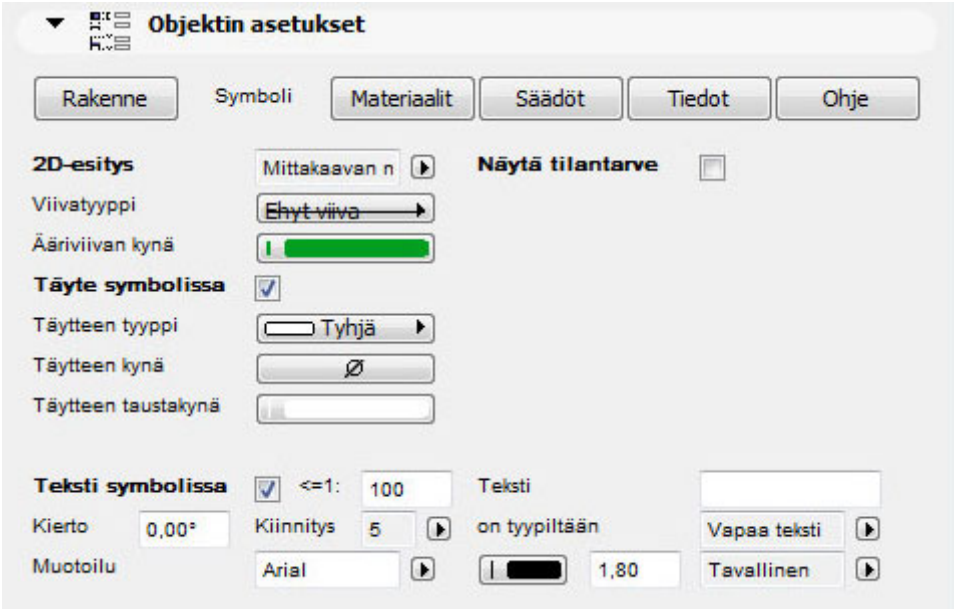
```
! Ensimmäinen rivi varattu painikkeille, selitteet ja
! kentät 2. riviltä alkaen
rivinro=2
! Seliteteksti ”VASEN” 1. sarakkeessa (x = 0)
ui_outfield "VASEN", 0, rivinro*rivikor, outlev,outkor
! Syöttökenttä ”ro20” 2. sarakkeessa (x = sarakelev)
ui_infield "ro20", sarakelev,rivinro*rivikor, inlev,inkor
! Seuraava rivi, jonka jälkeen mahdollisesti lisää selitteitä ja kenttiä
             rivinro=rivinro+1

rivinro=2
! Seliteteksti ”OIKEA” 3. sarakkeessa (x = 2*sarakelev)
ui_outfield "OIKEA", 2*sarakelev, rivinro*rivikor, outlev,outkor
! Syöttökenttä ”ro20” 4. sarakkeessa (x = 3*sarakelev)
ui_infield "ro20", 3*sarakelev,rivinro*rivikor, inlev,inkor
             rivinro=rivinro+1
```

Näiden asetteluperiaatteiden mukaan on sijoitettu kaikki painikkeet, selitteet ja kentät ”Objektin asetukset” -dialogisivun välilehdille.

Rakenne-välilehdelle sijoitetaan tärkeimpien mittojen ja rakennevaihtoehtojen selitteet ja syöttökentät ja vain tarvittaessa pohjan perusmuodon valinta. Rakennesäätöjen ja havainnekuvien käyttöön voidaan tarvittaessa ottaa myös sarakkeet 2 ja 3, jossa oletuksena on ”Tunnus”-parametrit, jotka on toteutettu myös Tiedot-välilehdellä.

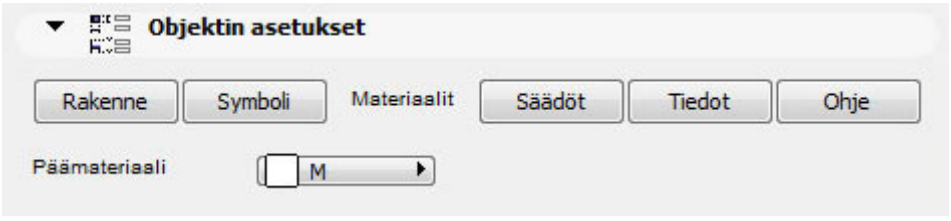
Kuva 13. Objektin asetukset -dialogisivun Rakenne-välilehti

Symboli-välilehdellä on toteutettu 2D-symbolin kynien, täytteiden, viivatyyppien, tekstin ja tilantarpeen säädöt. Näitä tarvitsee muokata vain harvoissa tapauksissa.



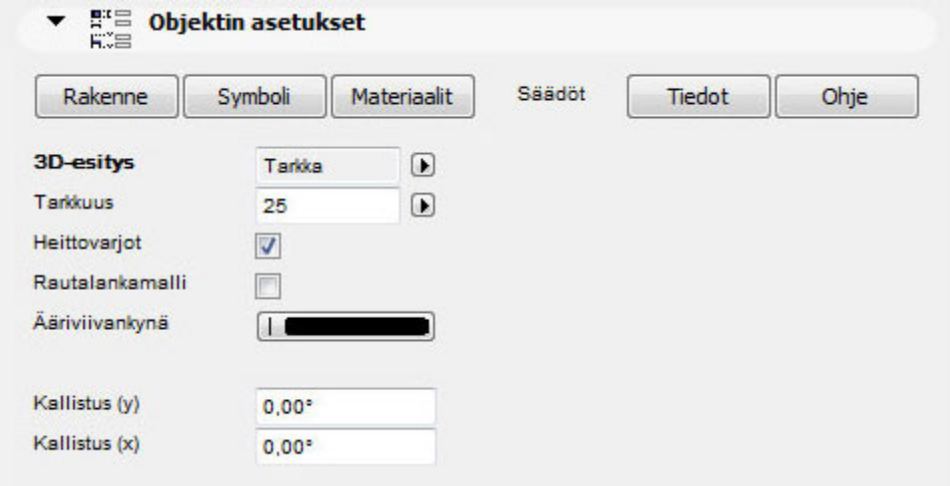
Kuva 14. Objektin asetukset -dialogisivun Symboli-välilehti

Materiaalit-välilehdellä toteutetaan kaikkien objektin 3D-tilassa käyttämät materiaalit. Materiaalivalintoja on valmiina käyttöliittymäohjelmassa ”!”-kommenteilla käytöstä piilotettuina, joten kommentit poistamalla saadaan helposti käyttöön perusobjektissa valmiiksi määriteltyjä materiaaliparametreja.



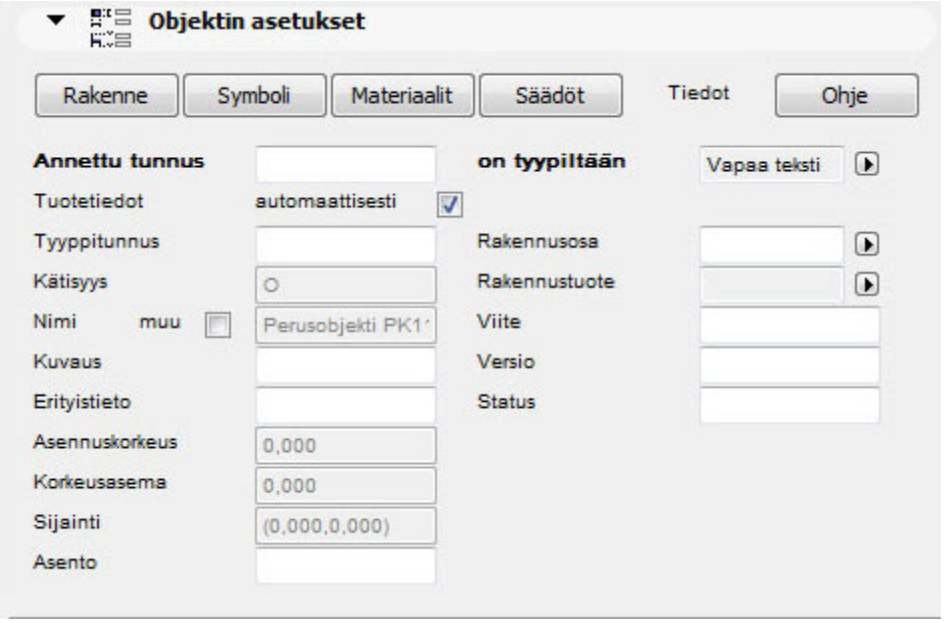
Kuva 15. Objektin asetukset -dialogisivun Materiaalit-välilehti

Säädöt-välilehdellä on objektin perussäädöt 3D-tilassa. Optiona tälle välilehdelle voidaan lisätä tarpeellisia säätöjä objektin säätöparametreille, jotka eivät mahdu Rakenne-välilehdelle.



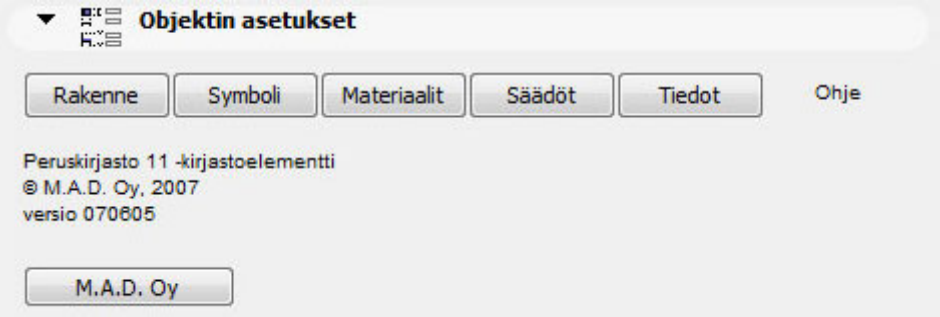
Kuva 16. Objektin asetukset -dialogisivun Säädöt-välilehti

Määrälaskentaa varten objektilla on Tiedot-välilehti, jossa objektille voidaan antaa tarkempia tuotetietoja. Tuotetietoparametreja voidaan lisätä tälle välilehdelle tarpeen vaatiessa.



Kuva 17. Objektin asetukset -dialogisivun Tiedot-välilehti

Ohje-välilehdellä esitetään objektin perustiedot, mahdolliset käyttöohjeet, valmistajan tai toteuttajan www-sivujen linkki tai muuta objektiin liittyvää ohjeistusta.



Kuva 18. Objektin asetukset -dialogisivun Ohje-välilehti

Käyttöliittymäohjelmasta on täydellinen listaus liitteessä 5.

5.2.7 Arvolistaohjelma

Arvolistaohjelmassa määritellään parametreille arvojen vaihtoehdot, joista käyttäjä voi valita. Olemassa olevia arvoja tarvitsee useimmiten muokata vain piilotettaessa objektin kannalta epäolennaisia vaihtoehtoja. Lisäyksiä arvolistaohjelmaan tehdään tarpeen mukaan ja se on jopa suositeltavaa virheellisten arvojen syöttämisen estämiseksi.

Arvolistaohjelmassa määritellään esimerkiksi mittakaava- ja tarkkuusvaihtoehdot 2D- ja 3D-esitystä varten:

```
! --- Mittakaava ja tarkkuus
values "gs_detlevel_2d" `Mittakaavan mukaan`,`1:50`,`1:100`,`1:200`,`Projisoitu`,
`Muu`
values "gs_detlevel_3d" `Tarkka`,`Yksinkertainen`,`Pois`, "Kutsuttu"
```

Arvolistaohjelmasta on täydellinen listaus liitteessä 5.

6 GDL-OBJEKTIEN TOTEUTUS KÄYTÄNNÖSSÄ

Tutkintotyötä tehtäessä on toteutettu muutamia objekteja mm. M.A.D. Oy:n peruskirjastoon lisättäväksi. Tässä luvussa esitetään yhden objektin toteutusprojekti PK11-perusobjektimalliin perustuen.

6.1 Case: Kuormalava-objekti

M.A.D. Oy saa paljon palautetta ja toiveita ArchiMAD-kerholaisilta, jotka ovat yrityksen asiakkaita ja ArchiCAD-ohjelmiston käyttäjiä. Usein objektien toteutus lähteekin asiakkaiden toivomuksista – näin on myös tässä luvussa esiteltävän varastotilojen suunnittelussa ja mitoituksessa käytettävän kuormalavaobjektin tapauksessa.

6.1.1 Toimeksianto

Kuormalavan mitat perustuvat FIN-, EUR- ja kertalavojen standardimittoihin, mutta toimeksiannon mukaisesti lavalle täytyy voida tarvittaessa määritellä myös omat mitat.

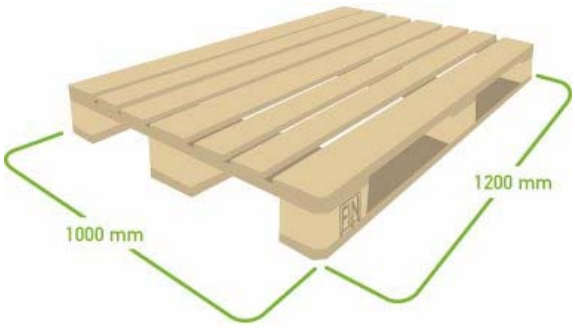
Käyttäjän täytyy voida valita, onko kuormalavalla kuorma vai ei. Kuorman korkeus pitää voida määritellä vapaasti. Lavan leveyttä ja pituutta sekä kuorman korkeutta täytyy voida säätää graafisesti suunnittelutilassa.

6.1.2 Standardit

Kuormalavat ja mitat perustuvat standardisoituihin kokoihin, vaikka erityistilauksesta ja -tarpeesta saatavilla on myös vapaita lavakokoja. Yleisimmät standardimitat sisällytetään objektin arvolistaan, josta mitat muutetaan esiohjelmassa A- ja B-mitoiksi. Tärkeimmät standardit ovat FIN- ja EUR-lavat /25/, sekä niiden ohella käytetyt monen mittaiset kertalavat.

FIN-lava

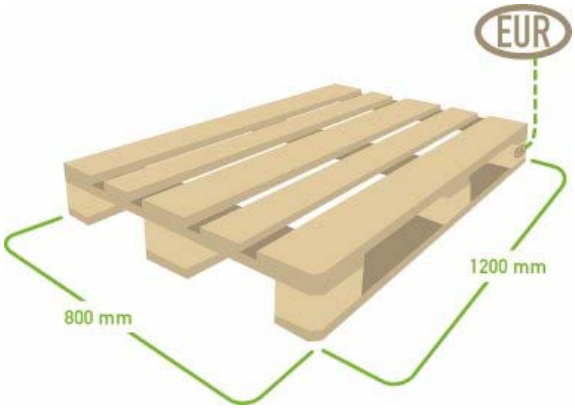
- rakenteeltaan standardiin perustuva nelitielava
- pohja-ala 1000 x 1200 mm, korkeus 144 mm
- lava yleisesti käytössä Suomen säännöllisessä tavaraliikenteessä
- lavaa ei juurikaan käytetä Suomen rajojen ulkopuolella. /25; 26/



Kuva 19. FIN-lava ja mitat (SFS EN 13698-2)

EUR-lava

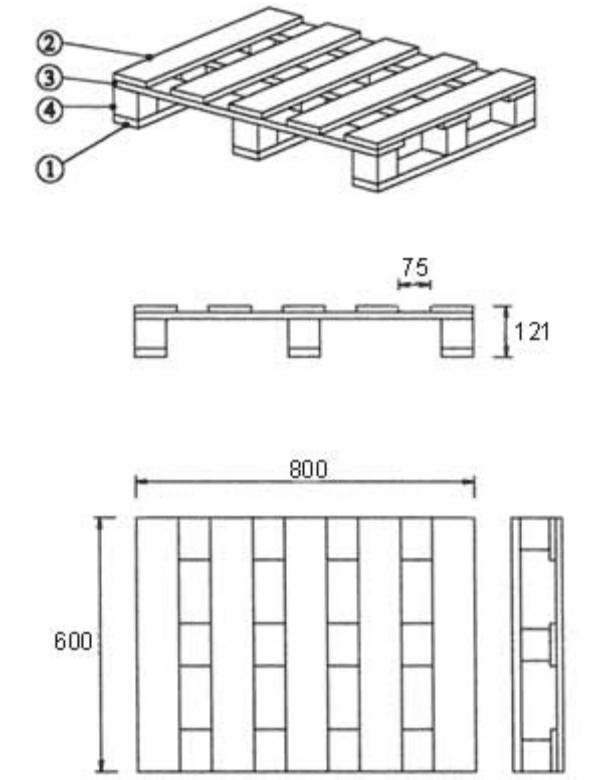
- rakenteeltaan standardiin perustuva nelitielava
- pohja-ala 800 x 1200 mm, korkeus 144 mm
- EUR-lavaa käytetään laajalti koko EUR-alueella ja EUR-merkintä helpottaa tunnistamista. /25; 26/



Kuva 20. EUR-lava ja mitat (SFS EN 13698-1)

Kertalava

Kertalavan mitoituksia on useita, mutta käytetyimmät vaihtoehdot eri valmistajilta ovat 800 x 600 mm – 1215 x 3000 mm, jotka lisätään objektin mittavaihtoehtoihin.



Kuva 21. Saarset Oy:n 800 x 600 kertalava /26/

6.1.3 Parametrit

Kuormalava-objektin toteuttamiseksi tarvitaan joukko objektin omia parametreja, joilla muokataan objektin rakennetta ja erityispiirteitä (taulukko 16).

Taulukko 16. Kuormalavan objektikohtaiset parametrit

Uudet parametrit	**Selite [tyyppi]**
kl_mitat	Kuormalavan mittavaihtoehdot [arvolista/omat mitat]*
kl_kork	Kuormalavan korkeus [pituus]
kuorma	Kuorma lavalla [totuusarvo]
kkork	Kuorman korkeus [pituus]*
lvali	Kansilautojen minimiväli [pituus]
lauta_paks	Laudan paksuus [pituus]
lautalev	Laudan leveys [pituus]
palikkakork	Välipalikan paksuus [pituus]

PK11-perusobjekti tarjoaa joukon parametreja, joita täytyy käyttää geometrioiden ja toiminnallisuuden ohjelmoinnissa, mikäli mahdollista. Kuormalava-objektissa käytettiin taulukossa 17 esitettyjä PK11-objektimallin valmiita parametreja.

Taulukko 17. Kuormalavan PK11-objektimallikohtaiset parametrit

PK11-parametrit	**Selite [tyyppi]**
A	Kuormalavan x-mitta / leveys [pituus]*
B	Kuormalavan y-mitta / pituus [pituus]*
ZZYZX	Kuormalavan z-mitta / korkeus [pituus]*
gs_frame_mat	Kuormalavan materiaali [materiaali]
gs_seat_mat	Kuorman materiaali [materiaali]

Taulukoissa 16 ja 17 tähdellä merkittyjä parametreja on voitava säätää graafisesti suunnittelutilassa.

6.1.4 Toimintalogiikka

Määrittelyjen pohjalta voidaan erotella kaksi eri toimintatilaa: kuormalava kuorma päällä tai tyhjä kuormalava. Molemmissa toimintatiloissa kuormalavan ulottuvuusparametrien A, B ja ZZYZX täytyy mukautua toimintatilaan sopiviksi.

Toimintatila ei vaikuta A- ja B-parametrien käyttäytymiseen, mutta korkeutta kuvaavan ZZYZX-parametrin on mukauduttava siihen, onko kuorma päällä vai ei. Tyhjän kuormalavan korkeus on ZZYZX, jota voidaan muuttaa graafisesti. Kuormalavan ja sen kuorman yhteiskorkeus on myös oltava ZZYZX, jolloin korkeuden muutos vaikuttaa vain kuorman korkeuteen kuormalavan korkeuden pysyessä vakiona.

Käyttäjän valitsee jonkin standardimittaisen lavan, jolloin kyseisen lavan mitat asettuvat automaattisesti A- ja B-parametrien arvoiksi. Jos kyseessä on EUR- tai FIN-lava, asettuu myös ZZYZX-parametrin arvo standardin mukaiseksi. Standardimittaisen kuormalavan täytyy olla venytettävissä graafisessa tilassa, jolloin sen mittojen oletusarvoksi muutetaan automaattisesti ”Omat mitat”.

Leveys, pituus ja korkeus täytyy olla muokattavissa A-, B- ja ZZYZX-parametrien kautta tartuntapisteillä objektin kulmapisteissä sekä keskellä.

Kuormalavan esitystarkkuuden sopeutuminen mittakaavaan automatisoidaan toteuttamalla yksinkertaistettu ja tarkka vaihtoehto. 2D-tilassa yksinkertaistetun esityksen raja-arvoksi valitaan ≤ 1:200 ja tarkemman esityksen raja-arvoksi > 1:200. 3D-tilassa esitystarkkuutta ohjaa parametri *gs_detlevel_3d* vaihtoehdoilla ”Tarkka” tai ”Yksinkertainen”.

6.1.4.1 Geometria

Kuormalava rakentuu laudoista ja välipalikoista seuraavasti:

- kerroksen nollatasoon kolme jalaslautaa, jotka sijoittuvat keskenään pituussuunnassa samansuuntaisina lavan reunoille ja keskelle
- jokaisessa jalaslaudassa on 3 välipalikkaa, keskellä ja molemmissa päissä
- jalaslaudat yhdistyvät toisiinsa keskenään vaakasuunnassa samansuuntaisilla välilaudoilla, jotka kiinnittyvät välipalikkoihin
- välilautojen päälle ladotaan päällimmäiset laudat pituussuunnassa keskenään samansuuntaisesti niin, että ne jakautuvat tasaisin välimatkoin ja että lautojen lukumäärä määräytyy lautojen pienimmän asetetun välityksen mukaan.

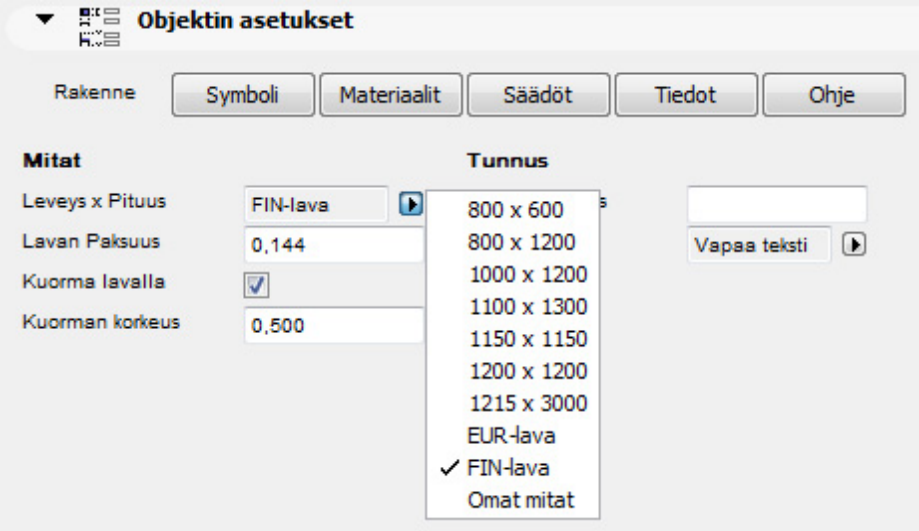
Kuormalavan kuorma mukailee lavan pituus- ja leveysmittoja, mutta käyttäjä voi muuttaa vapaasti kuorman korkeutta käyttöliittymäsivulta tai graafisessa suunnittelutilassa.

6.1.4.2 Käyttöliittymäsivu

Objektin käyttöliittymän dialogisivun Rakenne-välilehdellä voidaan määritellä lavan mitat joko perustuen valmiisiin standardimittoihin, jotka on kirjattu arvolistaohjelmassa, tai valitsemalla ”Omat mitat” vaihtoehto, jolloin käyttäjä voi antaa vapaasti omat arvot leveys- ja pituusparametreille.

Käyttäjän valitessa kuorman lavalle, näytetään kenttä jossa käyttäjä voi määritellä kuorman korkeuden.

Materiaalit-välilehdellä käyttäjä voi määritellä lavan ja kuorman materiaalit ArchiCADiin asennetuista materiaaleista. Oletuksena kuormalavan materiaali on ”Puu-mänty vaaka” ja kuorman materiaali on ”Puu-ulkovuoraus vaalea”.



Kuva 22. Kuormalavan käyttöliittymän dialogisivu ”Objektin asetukset/Rakenne”

6.1.5 Toteutus

Kuormalava-objektin pohjaksi valitaan PK11-perusobjekti, joka tallennetaan haluttuun hakemistoon nimellä ”Kuormalava PK11”. Tämän jälkeen luodaan uudet parametrit, jotka on esitelty taulukossa 16 ja annetaan näille oletusarvot. Kaikille ohjelmassa käytetyille suureille pyritään antamaan parametrisoitu arvo, jolloin myöhemmin voidaan muokata arvoja parametrien kautta. Osa parametreista on käytössä vain suunnittelu- ja toteutusvaiheessa ja niiden määrittely käyttäjän toimesta ei ole perusteltua.

6.1.5.1 Arvolistaohjelma

Arvolistaohjelmassa määritellään kl_mitat-parametrin arvot ja rajoitukset pituus- ja leveysmitoille sekä mittojen resoluutiot:

```
values "kl_mitat" `800 x 600`,`800 x 1200`,`1000 x 1200`,`1100 x 1300`,`1150 x
1150`,`1200 x 1200`,`1215 x 3000`, `EUR-lava`,`FIN-lava`, `Omat mitat`

values "a" range[.4,1.2) step 1, 0.05, range[1.2,10) step 0, 0.1
values "b" range[.4,1.2) step 1, 0.05, range[1.2,10) step 0, 0.1
```

6.1.5.2 Esiohjelma

Arvolistaohjelman jälkeen luodaan esiohjelmaan toiminnallisuus, joka mukauttaa parametrien arvot objektin toimintatilan mukaan sekä asettaa EUR- ja FIN-lavojen leveys- ja pituusmitat ”mm x mm” -muotoon ja korkeuden 144 mm:iin:

```
! --- EUR ja FIN - lavojen mittojen asetus, jos ei ole standardit tiedossa
              if kl_mitat = "EUR-lava" then
                            kl_mitat = "800 x 1200" ! ohjelmansisäinen arvo
                            parameters kl_kork = 0.144 ! standardikorkeus
                            lock "kl_kork" ! estetään standardikorkeuden muutos
                            ! välipalikan korkeuden asettaminen
                            parameters palikkakork = kl_kork - 3*lauta_paks
              endif
              if kl_mitat = "FIN-lava" then
                            kl_mitat = "1000 x 1200" ! ohjelmansisäinen arvo
                            parameters kl_kork = 0.144 ! standardikorkeus
                            lock "kl_kork" ! estetään standardikorkeuden muutos
                            ! välipalikan korkeuden asettaminen
                            parameters palikkakork = kl_kork - 3*lauta_paks
              endif

! Parametrit toimintatilan mukaan
eps = 0.00001 ! vakio: nolla-arvojen esto funktioissa
if kuorma = 0 then
              if k_ch = 0 then ! Kuormaa ei ole ollut vielä
                            if zzyzx < 3*lauta_paks + eps then
                                          zzyzx = 3*lauta_paks + eps
                            else
                                          kl_kork = zzyzx
                            endif
                            palikkakork = zzyzx - 3*lauta_paks
              else
                            parameters zzyzx = kl_kork
                            parameters k_ch = 0 ! Palataan alkutilaan
              endif
else
              if k_ch = 0 then ! Kuorma valittu päälle
              ! Siirretään arvot zzyzx-parametriin
              parameters zzyzx = kl_kork + kkork
              ! Kuorman valinta muuttanut Z-parametria
              parameters k_ch = 1
              else
              ! Kuorman korkeus riippuu Z-parametrista, jota voidaan esim.
              graafisesti muuttaa
              kkork = zzyzx - kl_kork

              endif
              palikkakork = zzyzx - 3*lauta_paks - kkork
endif
```

Seuraavaksi lisätään toiminto, joka tulkitsee arvolistan ”mm x mm” -arvot haluttuihin parametreihin (A = leveys, B = pituus) metreinä ja määritetään välipalikan korkeus annetun kuormalavan korkeuden perusteella:

```
! --- Mittojen sijoittaminen arvolistasta muuttujiin

if ed_kl_mitat <> kl_mitat then
              if kl_mitat = "Omat mitat" then
                            lev = a
                            pit = b
                            parameters ed_kl_mitat = kl_mitat
              else
              ! luetaan kl_mitat
                            nn = split(kl_mitat, "%n x %n",lev,xva,pit)
              ! jos saadaan tulos, sijoitetaan arvot ja muutetaan metreiksi
                            if nn = 3 then
                                          lev = lev/1000
```

```
                                        pit = pit/1000
                                        parameters ed_kl_mitat = kl_mitat
                            else
                                        lev = a
                                        pit = b
                                        parameters kl_mitat = "Omat mitat"
                                        parameters ed_kl_mitat = kl_mitat
                            endif
                endif
else
                ! luetaan kl_mitat
                            nn = split(kl_mitat, "%n x %n",lev,xva,pit)
                ! jos saadaan tulos, sijoitetaan arvot ja muutetaan metreiksi
                if nn = 3 then
                                        lev = lev/1000
                                        pit = pit/1000
                                        if a <> lev AND b <> lev then

                                        lev = a
                                        pit = b
                                        parameters kl_mitat = "Omat mitat"
                                        parameters ed_kl_mitat = kl_mitat
                                        else
                                                    lev = a
                                                    pit = b
                                        endif
                            else
                                        lev = a
                                        pit = b
                            endif
endif

! Tallennetaan lev ja pit parametreihin A ja B
parameters a = lev
parameters b = pit

palikkakork = kl_kork - 3*lauta_paks
```

6.1.5.3 3D-ohjelma

Esiohjelmassa tulkittujen arvolistan arvojen jälkeen parametreja voidaan käyttää ja niiden perusteella piirtää jalaslaudat ja niihin kiinnittyvät palikat 3D-ohjelmassa:

```
! alkusijainti pisteessä 0,0,0
!! LAVAN PIIRTO ========================================
3000:
 !materiaali
 set material gs_frame_mat

3100: !
! Jalkalaudat
block lautalev,b,lauta_paks ! jalaslauta
 gosub 3500                                ! välipalikka
 addx a/2-lautalev/2                       ! siirrytään oikealle x-suunnassa
block lautalev,b,lauta_paks ! jalaslauta
 gosub 3500 ! palikka                      ! välipalikka
 addx a/2-lautalev/2
block lautalev,b,lauta_paks
 gosub 3500 ! palikka
 addz palikkakork+lauta_paks
 addx -2*(lev/2-lautalev/2)
 gosub 3400                                ! Välilaudat (ristiin)
if (kuorma) gosub 4000                     ! kuorma
 gosub 3600                                ! Pintalaudat
return
```

Välipalikoiden piirtäminen aliohjelmassa:

```
3500: !Välipalikat
              addz lauta_paks
              block lautalev,lautalev,palikkakork
              addy b/2-lautalev/2
              block lautalev,lautalev,palikkakork
              addy b/2-lautalev/2
              block lautalev,lautalev,palikkakork
              del 3
return
```

Välipalikoiden päälle piirretään välilaudat, jotka yhdistävät jalaslaudat toisiinsa:

```
3400: ! välilaudat
              block  a, lautalev, lauta_paks          ! välilauta
                           addy b/2-lautalev/2        ! koordinaatiston siirto
              block  a, lautalev, lauta_paks
                           addy b/2-lautalev/2
              block  a, lautalev, lauta_paks
                           del 2
return
```

Ylimmät laudat piirretään välilautojen päälle ristiin:

```
3600: ! kansilaudat
              !materiaali
              set material gs_frame_mat
              addz lauta_paks
              laskuri = 0
              dellask = 0
                           ! 1. lauta
                           block lautalev,b,lauta_paks
                           addx a-lautalev
                           ! viim. lauta
                           block lautalev,b,lauta_paks
                           del 1
                           addx lautalev
                           ! reunimmaisten lautojen väliin jäävä tila
                           llev = a - 2*lautalev
              ! ed. väliin mahtuu max lkpl määrä kansilautoja väleineen
                           lkpl = int(llev / (lautalev+lvali))
                           ! lautojen korjattu väli
                           vali = (llev-lautalev*lkpl)/(lkpl+1)
              repeat
                           addx vali
                           block lautalev,b,lauta_paks
                           addx lautalev
                           dellask = dellask +1
                           laskuri = laskuri + 1
              until laskuri >= lkpl+1
              del dellask
return
```

Käyttäjän valinnasta riippuen piirretään lopuksi kuorma:

```
4000: !kuorma
!materiaali
set material gs_seat_mat
addz lauta_paks*2
block a,b,kkork
del 1
return
```

3D-ohjelmassa toteutetaan myös yksinkertainen 3D-esitys, jossa kaikki mahdolliset geometriat pyritään esittämään yleisellä tasolla niin, että se voidaan vielä tunnistaa, mutta yksityiskohdat on karsittu pois. Kuormalava-objektia voidaan yksinkertaistaa niin, että jalaslaudat ja välipalikat piirretään yhtenäisenä block-funktiolla ja näiden päälle taso, jonka paksuus on kaksi kertaa laudan paksuus. Kuorman toteutus pysyy samanlaisena.

```
!! LAVAN PIIRTO yksinkertainen
5000:
! Jalkalaudat
vali_paks = lauta_paks
lauta_paks = lauta_paks*2 + palikkakork
block lautalev,b,lauta_paks! 1. lauta
              addx a/2-lautalev/2
block lautalev,b,lauta_paks! 2. lauta
              addx a/2-lautalev/2
block lautalev,b,lauta_paks! 3. lauta
del 2
addz lauta_paks
              !materiaali
              set material gs_frame_mat
              block a,b,vali_paks        ! kansilevy
del 1
addz vali_paks+palikkakork
lauta_paks = vali_paks
if (kuorma) gosub 4000     ! kuorma
return
```

Graafisen säätämisen mahdollistamiseksi objektille määritellään tartuntapisteitä. Tartuntapisteet määritellään objektin kulmiin ja keskelle. Nämä ovat valmiina perusobjektimallin 2D- ja 3D-ohjelmissa osaksi kommenttimerkein käytöstä poistettuina, joten tarvitsee vain aktivoida kommentoidut tartuntapisteet käyttöön.

```
!! --- Tartuntapisteet
hotspot 0,0,0, unID : unID=unID+1
hotspot a,0,0, unID : unID=unID+1
hotspot a,b,0, unID : unID=unID+1
hotspot 0,b,0, unID : unID=unID+1

hotspot 0,0,zzyzx, unID : unID=unID+1
hotspot a,0,zzyzx, unID : unID=unID+1
hotspot a,b,zzyzx, unID : unID=unID+1
hotspot 0,b,zzyzx, unID : unID=unID+1

hotspot a/2,b/2,0, unID : unID=unID+1
hotspot a/2,b/2,zzyzx, unID : unID=unID+1
```

6.1.5.4 2D-ohjelma

Objektin 2D-symbolin kuvaus perustuu 3D-ohjelmassa käytettyihin x- ja y-mittoihin. Vaikka 2D-symboli voidaan generoida 3D-kuvauksen ylhäältä päin otetusta projektiosta, ei tätä tapaa suositella käytettäväksi. Projektiossa muodostetaan ensin objekti 3D-muodossa, jonka jälkeen luodaan näkymä suoraan objektin yläpuolelta. Tämä hidastaa ArchiCADia ja tuhlaa koneen resursseja.

2D-ohjelmassa piirretään ylhäältä päin näkyvät osat, eli kansilaudat ja välilaudat tai kuorma. Jos lavalla ei ole kuormaa, 3D-ohjelman aliohjelmassa ”3600” (kansilaudat) käytetty block-funktio korvataan alla olevalla 2D-funktiolla.

```
POLY2_B{2} 5, 1+2*gs_fill+4, gs_fill_pen, gs_back_pen, 0,0,0,
0,0,1,
lautalev,0,1,
lautalev,pit,1,
0,pit,1,
0,0,1
```

Tämän jälkeen koordinaatistosiirrot muutetaan 2D-muotoon *add2 x,y*.

Jos lavalla on kuorma tai mittakaava on ≤ 1:200, voidaan 2D-symboli muodostaa yksinkertaisesti piirtämällä vain kuorman tai kuormalavan reunat:

```
POLY2_B{2}    5, 1+2*gs_fill+4, gs_fill_pen, gs_back_pen, 0,0,0,
0,0,1,
lev,0,1,
lev,pit,1,
0,pit,1,
0,0,1
```

7 PÄÄTELMÄT

Objektiteknologian kehitys on ollut nopeaa, ja tulevaisuus näyttää tutkittujen asioiden pohjalta valoisalta siirryttäessä rakennuspiirtämisestä rakennusten tietomallintamiseen. Tietomallinnuksen vaatimukset yleistyvät suurista yrityksistä yhä pienempiin, mikä tehostaa rakennusprojektien täsmällisyyttä, kustannustehokkuutta sekä tavoitteiden asettamista ja täyttämistä.

Tällä hetkellä suurin ongelma objektiteknologian täysipainoisessa hyödyntämisessä ovat puutteelliset tiedonsiirtostandardit eri suunnitteluohjelmistojen välillä. Useat rakennusalan yritykset tarvitsevat enemmän tietoa tietoteknisistä ratkaisuista ja mahdollisuuksista sekä monitaitoista suunnitteluhenkilöstöä.

GDL-objektiteknologian tarjoamat potentiaaliset mahdollisuudet uusien innovaatioiden toteuttamiseksi ovat laajat, mutta Suomen koulutusjärjestelmä ei tällä hetkellä huomioi tarpeeksi rakennusalalla esiintyviä objektiteknologiaan ja tietomallintamiseen liittyviä ohjelmistotuotannollisia tarpeita. Tämä saattaa johtua siitä, että GDL-ohjelmointi ei ole selkeästi rakennusalan tai ohjelmistoalan sektorissa vaan jotain siltä väliltä. Tämän tutkintotyön yhtenä tarkoituksena on ollut osaltaan lähentää näitä kahta eri toimialasektoria ymmärtämään paremmin toisiaan.

Tutkitun PK11-perusobjektin sisältämä tietomallipohja täyttää kattavasti suomalaisten suunnittelijoiden tietomallinnuksen tarpeet ja sen tietosisältöä voidaan laajentaa tarpeen mukaan. Perusobjektin tietomalli auttaa ymmärtämään rakennusprojektien tietomallinnuksen periaatteita ja tarpeita laajemmassakin mittakaavassa.

GDL-ohjelmoinnin itseopiskelu vaatii ymmärtämystä rakennusalan suunnittelun käytännöistä ja tarpeista sekä ArchiCADin ominaisuuksien tyydyttävää hallintaa. Itse ArchiCADin käytön perusteiden opiskeluun löytyy suomenkielistä ja käyttökelpoista materiaalia, mutta objektien ohjelmoinnista ei ole saatavilla suomenkielistä materiaalia muutamia lehtiartikkeleita lukuunottamatta.

Itse GDL-ohjelmointikieli on selkeää ja ymmärrettävää, mutta objektien ohjelmoinnin voidaan todeta vaativan ohjelmoijalta monia erityisominaisuuksia

kuten kaksi- ja kolmiulotteisen avaruuden analyyttistä hallintaa, geometriafunktioiden ja tehokkaiden toimintalogiikoiden soveltamista sekä käyttäjäystävällisen käyttöliittymän ymmärtämistä. Näiden ominaisuuksien pohjalta voidaan tunnistaa potentiaalisia GDL-ohjelmoijan taitoja niin omalta kuin yritystenkin taholta.

Tutkintotyötä ei ole vielä päätelmiä tehtäessä hyödynnetty käytännössä opinto- ja lisätietomateriaalina, mutta se täyttää selkeän aukon GDL-teknologian suomenkielisessä dokumentoinnissa. Tästä syystä tutkintotyön voidaan olettaa olevan kiinnostava useiden eri tahojen tiedontarpeen kannalta.

Valmiiden mallien käyttö on kiistämättä hyödyllistä riippumatta ohjelmointikielestä, joten objektikirjastojenkin osalta on tutkittujen etujen perusteella suositeltavaa käyttää joko kirjastokohtaista objektimallia tai vaikkapa laadukkaaksi ja monipuoliseksi todettua PK11-objektimallia, mikäli kirjasto on tarkoitettu suomalaiseen käyttöön. Perusobjektista voidaan tarvittaessa toteuttaa kohtuullisessa ajassa toisenkielinenkin versio käyttöliittymänsä osalta kääntämällä parametrilistan ja dialogisivun selitteet halutulle kielelle. Lisäksi täytyisi hieman muokata arvolistaohjelmaa ja esiohjelmaa niin, että kielivalinta ei vaikuttaisi objektin toimintoihin.

GDL-objektit ovat pääasiassa yhteensopivia uudempien versioiden kanssa, mutta uusien ominaisuuksien myötä on aiheellista tarkistaa päivitystarve. ArchiCAD 12 ilmestyi virallisesti vasta opinnäytetyön lähes valmistuttua, joten objektit on tässä työssä optimoitu versiota 11 varten. Ennakkotietojen perusteella perusperusobjektimalli PK12 säilyy hyvin pitkälti samanlaisena kuin PK11, joten siltä osin työ on ajankohtainen seuraavankin version suhteen. PK11-periaatteen mukaiset objektit tulevat todennäköisesti olemaan käytössä pitkään yleiskäyttöisyytensä ansiosta, joten tässä tutkintotyössä tarkastellut perusasiat tulevat pysymään käyttökelpoisina pitkään.

Jatkotutkimusta objektiteknologian hyödyntämisestä voidaan tehdä useilla eri toimialasektoreilla ja itse näkisin hyödyllisimpänä tutkia nimenomaan usean eri alan tarpeiden yhdistämistä käytännössä objektiteknologisin ratkaisuin ja tehdä niistä yleiskäyttöisiä toimintamalleja. Lisäksi mainittakoon ArchiCAD-laajennusten API-ohjelmointi, joka mahdollistaa erittäin kehittyneiden objektiteknologioiden toteutuksen ja josta ei myöskään ole saatavilla suomenkielistä dokumentaatiota.

LÄHTEET

1 CADAZZ, the best CAD software history on the Web. [www-sivu]. [viitattu 14.5.2008] Saatavissa: http://www.cadazz.com/cad-software-history.htm

2 Martens, Bob - Peter, Herbert, ArchiCAD Best Practice: The Virtual Building Revealed. Springer Verlag. Wien 2006. 286 s.

3 M.A.D. Oy. [pdf-dokumentti]. [viitattu 14.5.2008] Saatavissa: http://www.mad.fi/mad/tiedostot/pdf/kasikirja11/FIN_TM.AC_tuotemallinnus.pdf

4 Rönkkönen, Teemu, Rakentaminen ja mallinnus. ArchiMAD 3/2005, s. 21.

5 Laine, Tuomas, Aurora II / Talotekniikan analyysit. ArchiMAD 4/2005, s. 22.

6 M.A.D. Oy. [pdf-dokumentti]. [viitattu 14.5.2008] Saatavissa: http://www.mad.fi/mad/tiedostot/pdf/kasikirja11/FIN_KIR.PK11_peruskirjasto11.pdf

7 Hietanen, Jiri, IFC, asiaa!. ArchiMAD 4/2005, s. 16.

8 YIT-yhtymä Oy. [pdf-dokumentti]. [viitattu 14.5.2008] Saatavissa: http://cic.vtt.fi/vera/Documents/SPADEX_final_report.pdf

9 Graphisoft. [www-sivu]. [viitattu 14.5.2008] Saatavissa: http://www.graphisoft.co.uk/products/object_technology/

10 Reinikainen, Jukka, Suomeen maailman laajin kivituotekirjasto. Rakennustaito 10/2002, s. 24-25.

11 Kiviteollisuusliitto ry. [www-sivu]. [viitattu 14.5.2008] Saatavissa: http://www.finstone.fi/db/gdl/index.html

12 Fora Form A.S. [www-sivu]. [viitattu 14.5.2008] Saatavissa: http://foraform.icatalog.aimit.se/

13 Wikipedia. [www-sivu]. [viitattu 14.5.2008] Saatavissa: http://en.wikipedia.org/wiki/Building_Information_Modeling

14 Senaatti Kiinteistöt. [www.sivu]. [viitattu 14.5.2008] Saatavissa: http://www.senaatti.fi/document.asp?siteID=1&docID=546

15 Rakennuslehti. [www-sivu]. [viitattu 14.5.2008] Saatavissa: http://www.rakennuslehti.fi/uutiset/lehtiarkisto/7605.html

16 VTT PRO IT 2005. [pdf-dokumentti]. [viitattu 14.5.2008] Saatavissa: http://virtual.vtt.fi/virtual/proj6/proit/julkiset_tulokset/proit_pilottiraportti_051115_vtt.pdf

17 GDL Cookbook 4. [pdf-dokumentti]. [viitattu 14.5.2008] Saatavissa: http://www.btsquarepeg.com/news/2008/03/11/download-gdl-cookbook-4-pdf/

18 Graphisoft GDL Reference Manual. Graphisoft. 2000.

19 Graphisoft GDL Reference Guide. [pdf-dokumentti]. [viitattu 14.5.2008] Saatavissa: www.idc.ch/Archicad/Download/PDF/gdl_refrence9.pdf

20 Graphisoft. [www-sivu] [viitattu 14.5.2008] Saatavissa: http://www.mad.fi/mad/tiedostot/pdf/gdltechstandards.pdf

21 VTT. [www-sivu]. [viitattu 14.5.2008] Saatavissa: http://www.vtt.fi/uutta/2006/20060323.jsp

22 Vuori, Matti, Ketterän toiminnan filosofiaa ja periaatteita. Systeemityö 4/2007 s.7-9.

23 Ohjelmistotuotteen käyttöliittymä, sen suunnittelu ja suunnittelija. [pdf-dokumentti]. [viitattu 14.5.2008] Saatavissa: http://www.plugit.fi/seminaari29300304/materiaalit/kayttoliittymasuunnittelu.pdf

24 Laakso, Karri-Pekka, Kokemuksia GUIDe-käyttöliittymäsuunnittelun ja Scrum-menetelmän yhdistämisestä. Systeemityö 4/2007 s.16-18.

25 Helsingin teknillinen korkeakoulu. [pdf-dokumentti]. [viitattu 14.5.2008] Saatavissa: http://www.tkk.fi/Yksikot/Liikenne/Opinnot/161/Luento_09.pdf

26 Saarset Oy. [www-sivu]. [viitattu 24.5.2008] Saatavissa: http://www.saarset.fi/kertalavat.html

27 Pietilä, Ville, 3D-objektien tekeminen mallintamalla ja piirtämällä. ArchiMAD 1/2004, s. 24-25.

28 Rantanen, Ville, Peitelista koristeprofiililla - GDL-objekti helposti. ArchiMAD 1/2006, s. 20-21.

29 Voutilainen, Hannu, Sisustussuunnittelua ArchiCADillä - Clarion Hotel Santa Claus. ArchiMAD 1/2008 s.6-8.

30 Aalto, Anu, ArchiCAD 11 aloituspohja Senaatille. ArchiMAD 1/2008 s.9-11.

31 Melvasalo, Lauri, Peruskirjasto 11. ArchiMAD 3/2007 s.3-6.

32 Jantunen, Pauli, Asiakastyytyväisyys. ArchiMAD 4/2007 s.19.

Vakioparametrit:

Parametri	**Tyyppi**
A	Pituus
B	Pituus
ZZYZX	Pituus
AC_show2DHotspotsIn3D	Totuusarvo

GS eli Graphisoft-parametrit:

Parametri	**Tyyppi**
gs_detlevel_2d	Teksti
gs_line_type	Viiva
gs_cont_pen	Kynä
gs_fill	Totuusarvo
gs_fill_type	Täyte
gs_fill_pen	Kynä
gs_back_pen	Kynä
gs_font_type	Teksti
gs_font_pen	Kynä
gs_font_size	Reaaliluku
gs_font_style	Teksti
gs_min_space	Totuusarvo
gs_detlevel_3d	Teksti
gs_resol	Kokonaisluku
gs_shadow	Totuusarvo
gs_view_pen	Kynä
gs_detlevel_sect	Teksti
gs_sect_pen	Kynä
gs_sect_fill_type	Täyte
gs_sect_fill_pen	Kynä
gs_sect_back_pen	Kynä
gs_frame_mat	Materiaali
gs_seat_mat	Materiaali
gs_toe_mat	Materiaali
gs_3_mat ... gs_10_mat	Materiaali
gs_frame_thk	Pituus
gs_seat_height	Pituus
gs_base_height	Pituus
gs_toe_height	Pituus
gs_ui_current_page	Kokonaisluku

PK11-parametrit:

Parametri	Tyyppi		Parametri	Tyyppi
txt_on	Totuusarvo		kantavuus	Teksti
txt_raja	Kokonaisluku		kestavyys	Teksti
txt	Teksti		pintaluokka	Teksti
opeta_txt	Teksti		syttymisluokka	Teksti
txta	Kokonaisluku		palonlevitysluokka	Teksti
txtx	Pituus		muuominaisuus	Teksti
txty	Pituus		valmistaja	Teksti
txtr	Kulma		tuotenimi	Teksti
msfront	Pituus		tuotenumero	Teksti
msright	Pituus		kayttoika	Teksti
msleft	Pituus		ymparistovaikutus	Teksti
msrear	Pituus		elinkaarikustannus	Teksti
msfilltype	Täyte		laske	Totuusarvo
msfillpen	Kynä		opeta_tieto	Totuusarvo
msfillbackpen	Kynä		opeta_tunnus	Teksti
msside	Pituus		ro20	Teksti
varaus3d	Totuusarvo		rt20	Teksti
kutsuttu	Teksti		tyyppi	Teksti
korjaa	Pituus		katisyys	Teksti
korjaakulma	Kulma		tunnus	Teksti
muoto	Teksti		nimi	Teksti
osa3 ... osa10	Pituus		muunimi	Totuusarvo
s1 ... s10	Pituus		kuvaus	Teksti
rakenne	Teksti		tieto	Teksti
heloitus	Teksti		viite	Teksti
lukitus	Teksti		versio	Teksti
liitosrakenne	Teksti		status	Teksti
varuste	Teksti		asennuskorkeus	Pituus
laite	Teksti		korkeusasema	Pituus
kynnys	Teksti		sijainti	Teksti
puite	Teksti		asento	Teksti
laatu	Teksti		guid	Teksti
ominaisuus	Teksti		liittyminen	Teksti
paloluokka	Teksti		origosijainti	Kokonaisluku
dbluku	Teksti		nayta_origo	Totuusarvo
uarvo	Teksti		tilt	Kulma
materiaali	Teksti		angle	Kulma
pinta	Teksti			

LIITE 2. TOISIAAN VASTAAVIEN GDL-OBJEKTIEN VERTAILUA ERI TOTEUTUSTAPOJEN VÄLILLÄ

Mallinnettu GDL-objekti: Kuutio

2D-ohjelmalistaus

```
mul2     A/1.000000, B/1.000000
add2     -15'-1.1962", -28'-11.3939"
pen      89
set fill "Tyhjä"
poly2_b{3}       5, 3, 0, 0,
    0, 0, 1, 0, 0, 1,
        4.602382436159, 8.823804336026,     1,
        5.602382436159, 8.823804336026,     1,
        5.602382436159, 9.823804336026,     1,
        4.602382436159, 9.823804336026,     1,
        4.602382436159, 8.823804336026,     1
hotspot2 4.602382436159, 8.823804336026
hotspot2 5.602382436159, 8.823804336026
hotspot2 5.602382436159, 9.823804336026
hotspot2 4.602382436159, 9.823804336026
hotspot2 4.602382436159, 8.823804336026
```

3D-ohjelmalistaus

```
mulx                    A/          1
muly                    B/          1
mulz                    ZZYZX/          1
add     -4.602382436159, -8.823804336026,           0
body    -1
model solid
resol       36
GLOB_SCRIPT_TYPE =      3
GLOB_CONTEXT =      3
GLOB_SCALE =    100
GLOB_NORTH_DIR =          90
GLOB_DRAWING_BGD_PEN =     19
GLOB_FRAME_NR =     -1
GLOB_EYEPOS_X = 6.335727755887
GLOB_EYEPOS_Y = 5.574522435378
GLOB_EYEPOS_Z = 1.678755704738
GLOB_TARGPOS_X = 4.708165962562
GLOB_TARGPOS_Y = 15.01942813068
GLOB_TARGPOS_Z = -1.986589108107
GLOB_SUN_AZIMUTH =          240
GLOB_SUN_ALTITUDE =          35
body    -1
body    -1
body    -1
body    -1
body    -1
body    -1
body    -1
body    -1
body    -1
body    -1
body    -1
body    -1
GLOB_HSTORY_HEIGHT =           3
pen      89
GLOB_LAYER = "AR122 Alapohja.TALO"
GLOB_ID = "Laatta 001"
GLOB_INTGUID = "10FB356A-EA4B-44E9-B6BC-96348DEE6497"
body    -1
cprism_ "DEFAULT", "DEFAULT", "DEFAULT",
          5,          1,
        4.602382436159, 8.823804336026,     15,
        5.602382436159, 8.823804336026,     15,
        5.602382436159, 9.823804336026,     15,
        4.602382436159, 9.823804336026,     15,
        4.602382436159, 8.823804336026,     -1
body    -1
```

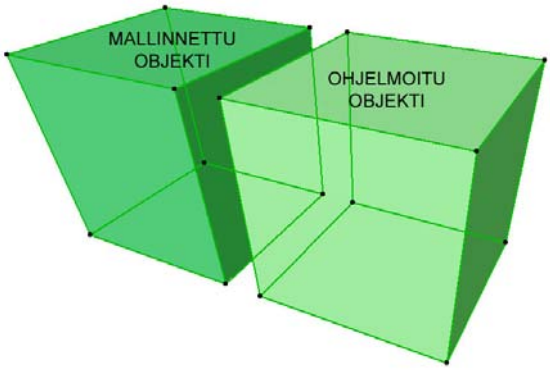
Ohjelmoitu GDL-objekti: Kuutio

2D-ohjelmalistaus

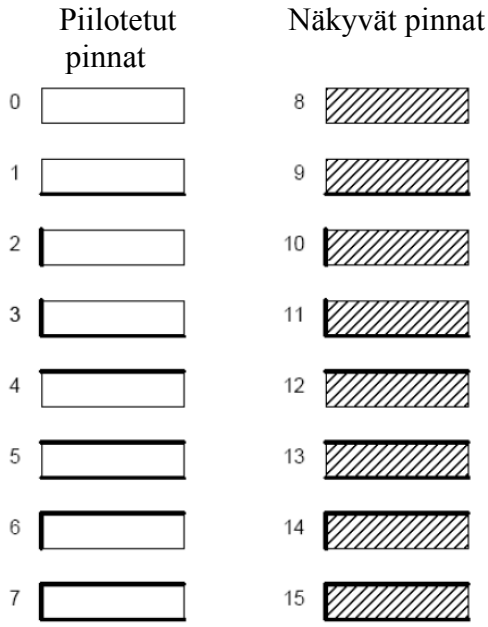
```
poly2_b{3} 5, 3, 0, 0, 0, 0, 1, 0, 0, 1,
                        0,0,1,
                        0,b,1,
                        a,b,1,
                        a,0,1,
                        0,0,1
hotspot2 0,0
hotspot2 0,b
hotspot2 a,b
hotspot2 a,0
hotspot2 a/2,b/2
```

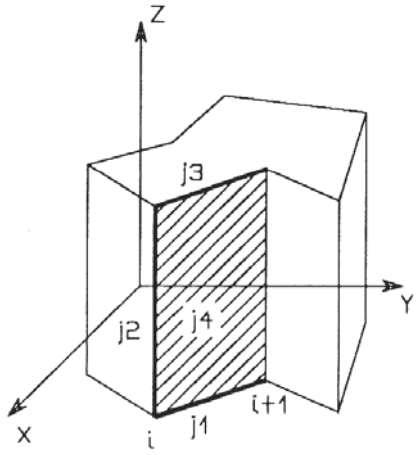
3D-ohjelmalistaus

```
cprism_ "DEFAULT", "DEFAULT", "DEFAULT",
          5,          zzyzx,
                        0,0,15,
                        a,0,15,
                        a,b,15,
                        0,b,15,
                        0,0,-1
```

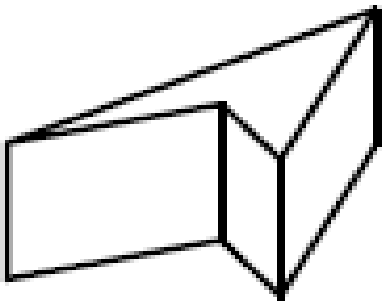


LIITE 3: S-STATUSARVOT (0-15) REUNA-, YLÄ- JA ALAMATERIAALIN TAI VIIVAN NÄYTTÄMISEKSI MÄÄRITTELYPISTEESTÄ SEURAAVAAN PISTEESEEN

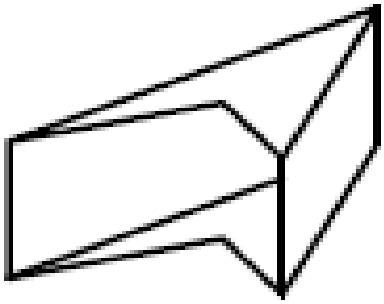




S-statusarvon määrittäminen binäärikertoimilla:
$S = j_1*1 + j_2*2 + j_3*4 + j_4*8$, missä $j_n = 0$ tai 1



```
PRISM_ 4,1,
        0,0,15,
        1,1,15,
        2,0,15,
        1,3,15
```

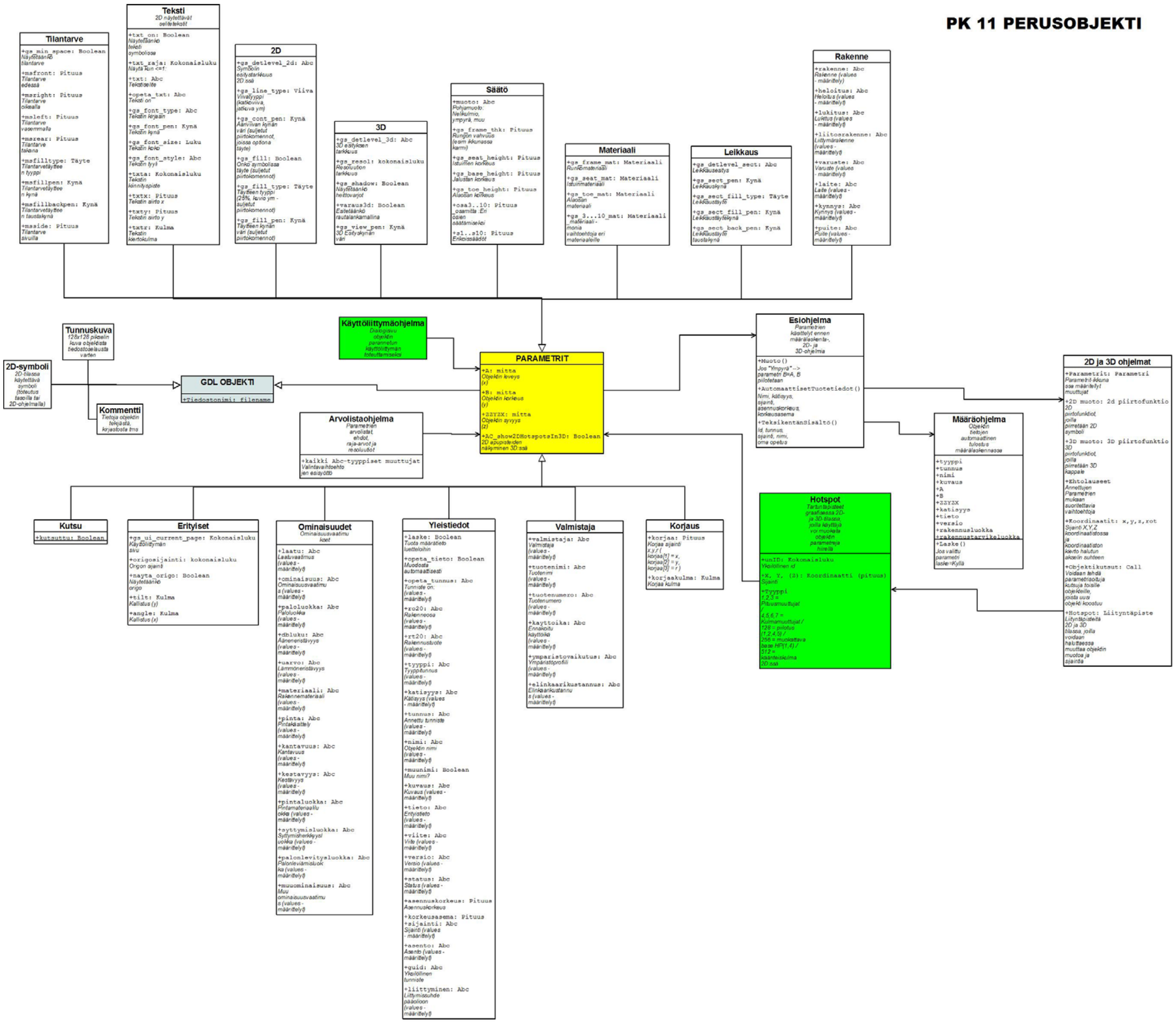


```
PRISM_ 4,1,
        0,0,7,
        1,1,5,
        2,0,15,
        1,3,15
```

> 15 STATUSARVOILLA VOIDAAN TOTEUTTAA MM. ERILAISIA PYÖRISTYKSIÄ JA KAARIPIIRTOFUNKTIOITA

(GDL Reference Guide s.139-150.)

LIITE 4: PK11-PERUSOBJEKTIN PARAMETRIT, OHJELMALISTAUKSET JA NIIDEN VÄLISET YHTEYDET



ESIOHJELMA

```
! --- Objektikohtaisesti opetetut
! parameters origosijainti=5, muoto="Nelikulmio"
! ---
! --- Jos muoto "Ympyrä"
if muoto="Ympyrä" then
              parameters b=a
              hideparameter "b"
endif
! --- Jos tuotetiedot muodostetaan automaatisesti, niin seuraavat
if opeta_tieto then
              main_name=""
              if muunimi=0 then
                            kysy=REQUEST ("Name_of_main", "", main_name)
                            kysy=REQUEST ("Name_of_macro", "", my_name)
                            if main_name="" then
                            ! jos tätä objektia ei ole kutsuttu, niin käytetään...
                                          parameters nimi=strsub(my_name,1,strstr(my_name,".")-1)
                            ! objektin omaa nimeä
                            else
                                          parameters nimi=strsub(main_name,1,strstr(my_name,".")-1)
                            ! kutsuvan objektin nimeä
                            endif
              endif
              if SYMB_MIRRORED then
                            ! kätisyys, asento
                            katisyys="V"
              else
                            katisyys="O"
              endif
              parameters katisyys=katisyys

              parameters sijainti="("+str("%.3m",SYMB_POS_X)+","+str("%.3m",SYMB_POS_Y)+")"
              parameters asennuskorkeus=GLOB_ELEVATION , korkeusasema=GLOB_HSTORY_ELEV+GLOB_ELEVATION

              ! --- opetetut tunnukset
              if opeta_tunnus="ID" then parameters tunnus=GLOB_ID
              if opeta_tunnus="Tyyppi+ID" then parameters tunnus=tyyppi+GLOB_ID
              if opeta_tunnus="Nimi" then parameters tunnus=nimi
              if opeta_tunnus="ID+kätisyys" then parameters tunnus=GLOB_ID+katisyys
              if opeta_tunnus="Tyyppi+ID+kätisyys" then parameters tunnus=tyyppi+GLOB_ID+katisyys

              lock "katisyys", "asennuskorkeus", "korkeusasema", "sijainti"
              if muunimi=0 then lock "nimi"
              if opeta_tunnus<>"Vapaa teksti" then lock "tunnus"
else
!             hideparameter "opeta_tunnus"
endif
! --- Tekstikentän sisällön opetus
              if opeta_txt="ID" then : parameters txt=GLOB_ID : lock "txt" : endif
              if opeta_txt="Tunnus" then : parameters txt=tunnus : lock "txt" : endif
              if opeta_txt="Sijainti" then : parameters txt=sijainti : lock "txt" : endif
              if opeta_txt="Nimi" then : parameters txt=nimi : lock "txt" : endif
              if opeta_txt="Oma opetus" then : parameters txt="Opetettu teksti" : lock "txt" : endif !
tässä voi monistaa lisää omia automaatteja...
```

2D-OHJELMA

```
! --- Origo
hotspot2 0,0, 201 : unID=unID+1
if nayta_origo then circle2 0,0,0.1        ! näyttää origon paikan (apulainen työvaiheessa)

! --- ja sen sijainti
goto origosijainti
1: add2    0,  -b           : goto 10      ! takana vasemmalla
2: add2 -a/2,  -b           : goto 10      ! takana keskellä
4: add2    0,-b/2           : goto 10      ! keskellä vasemmalla
5: add2 -a/2,-b/2           : goto 10      ! keskellä
7: add2    0,   0           : goto 10      ! edessä vasemmalla (ei siirtoa)
8: add2 -a/2,   0           : goto 10      ! edessä keskellä
10:

! --- Tartuntapisteet
hotspot2 0,0, unID : unID=unID+1           ! vasen etukulma
hotspot2 a,0, unID : unID=unID+1           ! oikea etukulma
hotspot2 a,b, unID : unID=unID+1           ! oikea takakulma
hotspot2 0,b, unID : unID=unID+1           ! vasen takakulma
! hotspot2 a/2,b, unID : unID=unID+1       ! takana keskellä
hotspot2 a/2,b/2, unID : unID=unID+1       ! keskellä

! --- Mittakaava
if gs_detlevel_2D = `1:200`  then det2d=1
if gs_detlevel_2D = `1:100`  then det2d=2
if gs_detlevel_2D = `1:50`                 then det2d=3
if gs_detlevel_2D = `Muu`                  then det2d=4
if gs_detlevel_2D = `Mittakaavan mukaan` then
              if GLOB_SCALE>150 then det2d=1
              if GLOB_SCALE>75 and GLOB_SCALE<=150 then det2d=2
              if GLOB_SCALE<=75 then det2d=3
endif
if gs_detlevel_2d=`Projisoitu` then det2d=5

! --- Teksti
if txt_on and a_<=txt_raja then
              hotspot2 a/2, b/2+txty, unID,  txtx,  1+128 : unID=unID+1
              hotspot2 a/2+txtx, b/2+txty, unID,  txtx,  2              : unID=unID+1
              hotspot2 -1, b/2+txty, unID,  txtx,  3                    : unID=unID+1

              hotspot2 a/2+txtx, b/2, unID,  txty,  1+128 : unID=unID+1
              hotspot2 a/2+txtx, b/2+txty, unID,  txty,  2              : unID=unID+1
              hotspot2 a/2+txtx, -1, unID,  txty,  3                    : unID=unID+1

    if gs_font_style=`Tavallinen`                        then font_style=0
    if gs_font_style=`Lihavoitu`                         then font_style=1
    if gs_font_style=`Kursivoitu`                        then font_style=2
    if gs_font_style=`Alleviivattu`        then font_style=4

              define style "tekstityyli" gs_font_type,gs_font_size,txta,font_style
              style "tekstityyli"
              pen gs_font_pen
              add2 a/2+txtx,b/2+txty

              if SYMB_ROTANGLE>90 and SYMB_ROTANGLE<=270 then rot2 180 else rot2 0
              rot2 txtr
```

```
               if opeta_txt="Tunnus" and opeta_tieto then  ! kätisyys pitää opettaa myös täällä
                                                           ! peilausvirheen (cmnd+M) vuoksi
                              if opeta_tunnus="ID+kätisyys" then txt=GLOB_ID+katisyys
                              if opeta_tunnus="Tyyppi+ID+kätisyys" then txt=tyyppi+GLOB_ID+katisyys
               endif
               text2 0,0,txt
               del 3
endif

! --- Tilantarve
if gs_min_space=1 then
               if muoto="Ympyrä" then
                              ! opetetaan tilantarpeeseen reikä olion kohdalle
                              put a/2, b/2, 901, a/2, 360, 4001
               else
                              put 0, 0, 1, a, 0, 1, a, b, 1, 0, b, 1, 0, 0, -1
               endif

               fill msfilltype
               poly2_B{2}                    5+(nsp)/3, 2+4, msfillpen, msfillbackpen, 0,0,0,
                                                             -msleft, -msfront, 1,
                                                            a+msright, -msfront, 1,
                                                            a+msright, b+msrear, 1,
                                                             -msleft, b+msrear, 1,
                                                             -msleft, -msfront, -1,

                                                            get (nsp)
endif

! --- Kynä, viivatyyppi ja täyte
pen gs_cont_pen
line_type gs_line_type
fill gs_fill_type

! --- Haetaan symbolin kuvaus
if muoto="Ympyrä" then
               gosub 2000+det2d*100 ! ympyrä
else
               gosub 1000+det2d*100 ! nelikulmio
endif

del top
end

1000:          !
1100:          ! 1:200
1200:          ! 1:100
1300:          ! 1:50
1400:          ! muu
               POLY2_B{2}                    5, 1+2*gs_fill+4, gs_fill_pen, gs_back_pen, 0,0,0,
                                                            0,0,1,
                                                            a,0,1,
                                                            a,b,1,
                                                            0,b,1,
                                                            0,0,1
return
1500:
!              POLY2_B{2}                    5, 0+2*gs_fill+4, gs_fill_pen, gs_back_pen, 0,0,0,
               ! täytteen kanssa
```

```
!                                                       0,0,1,
!                                                       a,0,1,
!                                                       a,b,1,
!                                                       0,b,1,
!                                                       0,0,1
              del top
              project2{2} 3, 270, 32+2+gs_fill, gs_back_pen, 0, 0, 0
              ! 3D:stä projisoitu esitys
return

2000:         !
2100:         ! 1:200
2200:         ! 1:100
2300:         ! 1:50
2400:         ! muu
              POLY2_B{2}                  2, 1+2*gs_fill+4, gs_fill_pen, gs_back_pen, 0,0,0,
                                                      a/2,b/2,901,
                                                      a/2,360,4001
return
2500:
!             POLY2_B{2}                  2, 0+2*gs_fill+4, gs_fill_pen, gs_back_pen, 0,0,0,
!                                                     a/2,b/2,901,
!                                                     a/2,360,4001
              del top
              project2{2} 3, 270, 32+2+gs_fill, gs_back_pen, 0, 0, 0
              ! 3D:stä projisoitu esitys
return
```

3D-OHJELMA

```
! --- Origo
hotspot 0,0,0, 301 : unID=unID+1
if nayta_origo then sphere 0.1              ! näyttää origon paikan (apulainen työvaiheessa)
hotspot 0,0,zzyzx, unID : unID=unID+1

! --- Kallistelut
! rect a,b !taso
roty -tilt
rotx angle

! --- ja sen sijainti
goto origosijainti
1: add    0,  -b, 0                         : goto 10      ! takana vasemmalla
2: add -a/2,  -b, 0                         : goto 10      ! takana keskellä
4: add    0,-b/2, 0                         : goto 10      ! keskellä vasemmalla
5: add -a/2,-b/2, 0                         : goto 10      ! keskellä
7: add    0,   0, 0                         : goto 10      ! edessä vasemmalla (ei siirtoa)
8: add -a/2,   0, 0                         : goto 10      ! edessä keskellä
10:

! --- Tartuntapisteet
hotspot 0,0,0, unID : unID=unID+1           ! vasen etualakulma
hotspot a,0,0, unID : unID=unID+1           ! oikea etualakulma
hotspot a,b,0, unID : unID=unID+1           ! oikea taka-alakulma
hotspot 0,b,0, unID : unID=unID+1           ! vasen taka-alakulma
! hotspot a/2,b,0, unID : unID=unID+1       ! keskellä takana alhaalla
hotspot a/2,b/2,0, unID : unID=unID+1       ! keskellä alhaalla
```

```
hotspot 0,0,zzyzx, unID : unID=unID+1      ! vasen etuyläkulma
hotspot a,0,zzyzx, unID : unID=unID+1      ! oikea etuyläkulma
hotspot a,b,zzyzx, unID : unID=unID+1      ! oikea takayläkulma
hotspot 0,b,zzyzx, unID : unID=unID+1      ! vasen takayläkulma
! hotspot a/2,b,zzyzx, unID : unID=unID+1  ! keskellä takana ylhäälla
hotspot a/2,b/2,zzyzx, unID : unID=unID+1  ! keskellä ylhäällä

! --- Tarkkuus
det3d=0 ! jos tätä objektia kutsutaan tämä "nollaa" arvon
if gs_detlevel_3d=`Tarkka` then det3d=1    ! oletus
if gs_detlevel_3d=`Pois` then end                                    ! lopetetaan, jos 3D pois
if gs_detlevel_3d="Kutsuttu" then                                    ! kutsutaan muuta objektia
geometriaksi
              det3d=2
!             if kutsuttu="" then det3d=0               ! jos ei nimeä niin palikka
endif
if gs_shadow=0 then shadow off                                                    ! varjot
päälle/pois
resol gs_resol
              ! särmiä...
if varaus3d then model wire                                                       !
rautalankaesitys (varaus)

! --- Esitystarkkuus leikkauksissa...
! if GLOB_CONTEXT=4 then                                                          !
leikkaus/julkisivu
! endif

! --- Materiaali ja reunakynä
material gs_frame_mat
pen gs_view_pen

! --- Haetaan geometrian kuvaus
if muoto="Ympyrä" then
              gosub 2000+det3d*100
else
              gosub 1000+det3d*100
endif

del top
end


1000:         ! yksinkertainen
block a,b,zzyzx
return

1100:         ! tarkka
block a,b,zzyzx
return

1200:         ! kutsuttu
if kutsuttu="" then return
add korjaa[1],korjaa[2],korjaa[3]
rotz korjaakulma
call kutsuttu parameters a=a, b=b, zzyzx=zzyzx
del 2
return
```

```
2000:          ! yksinkertainen
add a/2,b/2,0 : cylind zzyzx,a/2 : del 1
return

2100:          ! tarkka
add a/2,b/2,0 : cylind zzyzx,a/2 : del 1
return

2200:          ! kutsuttu
if kutsuttu="" then return
add korjaa[1],korjaa[2],korjaa[3]
rotz korjaa[4]
call kutsuttu parameters a=a, b=b, zzyzx=zzyzx
del 2
return
```

MÄÄRÄLASKENTAOHJELMA

```
drawing
binaryprop

! --- Yleiset tiedot luetteloihin
if laske then
               descriptor ""+tyyppi, "001"
               descriptor ""+tunnus, "002"
               descriptor ""+nimi, "003"
               descriptor ""+kuvaus, "004"

               descriptor ""+str("%.0mm",A), "005"
               descriptor ""+str("%.0mm",B), "006"
               descriptor ""+str("%.0mm",ZZYZX), "007"
               descriptor ""+katisyys, "008"
               descriptor ""+tieto, "009"
               descriptor ""+versio, "010"

               if ro20<>"" then
               ! jos kenttä ei ole tyhjä
                              component strsub(ro20,strstr(ro20," ")+1,strlen(ro20)), 1, "kpl", 1,
                              strsub(ro20,1,strstr(ro20," ")-1)
               !              database_set "PK11 tietokanta"
               !              ref component strsub(ro20,1,strstr(ro20," ")-1), "001"
                              ! mahdollisuus linkittää numerotunnuksella tietokantaan
               else
                              component "Rakennusosaluokkaa ei ole määritelty", 1, "kpl", 1, "Ro"
               endif

               if rt20<>"" then
               ! jos kenttä ei ole tyhjä
                              component strsub(rt20,strstr(rt20," ")+1,strlen(rt20)), 1, "kpl", 1,
                              strsub(rt20,1,strstr(rt20," ")-1)
               !              database_set "PK11 tietokanta"
               !              ref component strsub(rt20,1,strstr(rt20," ")-1), "001"
                              ! mahdollisuus linkittää numerotunnuksella tietokantaan
               else
                              component "Rakennustarvikeluokkaa ei ole määritelty", 1, "kpl", 1, "Rt"
```

```
                endif
endif
```

ARVOLISTAOHJELMA

```
! --- Mittakaava ja tarkkuus
values "gs_detlevel_2d" `Mittakaavan mukaan`,`1:50`,`1:100`,`1:200`,`Projisoitu`, `Muu`
values "gs_detlevel_3d" `Tarkka`,`Yksinkertainen`,`Pois`, "Kutsuttu"

values "gs_resol" range[3,)

! --- Käyttöliittymä
values "gs_ui_current_page" 1, 2, 3, 4, 5, 6
if GLOB_UI_BUTTON_ID = 1 then parameters gs_ui_current_page = 1
if GLOB_UI_BUTTON_ID = 2 then parameters gs_ui_current_page = 2
if GLOB_UI_BUTTON_ID = 3 then parameters gs_ui_current_page = 3
if GLOB_UI_BUTTON_ID = 4 then parameters gs_ui_current_page = 4
if GLOB_UI_BUTTON_ID = 5 then parameters gs_ui_current_page = 5
if GLOB_UI_BUTTON_ID = 6 then parameters gs_ui_current_page = 6

! --- Origopisteen sijainti (kätisyys peilaamalla)
values "origosijainti" 1, 2, 4, 5, 7, 8

! --- Pohjamuoto (täytettä varten)
values "muoto" "Nelikulmio", "Ympyrä", custom

! --- Kynien ja täytteiden "yksinkertaistaminen"
! if gs_cont_pen=0 then parameters gs_cont_pen=4

! --- Kätkettävät (ei tarpeellista...)
 if gs_detlevel_2d=`Projisoitu` then hideparameter "gs_line_type", "gs_cont_pen"
 if gs_fill=0 then hideparameter "gs_fill_type", "gs_fill_pen", "gs_back_pen"
 if gs_min_space=0 then hideparameter "msfront","msside", "msright",
"msleft","msrear","msfilltype","msfillpen","msfillbackpen"
 if txt_on=0 then hideparameter "txt", "opeta_txt", "txt_raja", "gs_font_type", "gs_font_pen",
"gs_font_size", "gs_font_style",

               "txta", "txtx", "txty", "txtr"

! --- Tekstit
dim fontNames[]
n = request("FONTNAMES_LIST", "", fontNames)
values "gs_font_type" fontNames, custom
values "gs_font_style" `Tavallinen`,`Lihavoitu`,`Kursivoitu`,`Alleviivattu`
values "txta" 1,2,3,4,5,6,7,8,9

! --- tekstikentän sisällön opetus
values "opeta_txt" "Vapaa teksti", "Tunnus", "Sijainti", "Nimi" ! , "Oma opetus"

! --- Tunnuksen opetus
values "opeta_tunnus"        "Vapaa teksti",
                             "ID", "ID+kätisyys",
                             "Tyyppi+ID", "Tyyppi+ID+kätisyys",
                             "Nimi"         ! , "Oma opetus"

! --- Rakennusosaluokka      Hankenimikkeistö 13.2.2007
values "ro20"  "",
```

```
                "0 Piirustusmerkinnät",
                "1 Rakennusosat",
                "1.1 Alueosat",
                                "1.1.1 Maaosat",
                                "1.1.2 Tuennat ja vahvistukset",
                                "1.1.3 Päällysteet",
                                "1.1.4 Alueen varusteet",
                                "1.1.5 Alueen rakenteet",
                "1.2 Talo-osat",
                                "1.2.1 Perustukset",
                                "1.2.2 Alapohjat",
                                "1.2.3 Runko",
                                "1.2.4 Julkisivut",
                                "1.2.5 Ulkotasot",
                                "1.2.6 Vesikatot",
                "1.3 Tilaosat",
                                "1.3.1 Tilan jako-osat",
                                "1.3.2 Tilapinnat",
                                "1.3.3 Tilavarusteet",
                                "1.3.4 Muut tilaosat",
                                "1.3.5 Tilaelementit",
                "2 Tekniikkaosat",
                "2.1 Putkiosat",
                "2.2 Ilmanvaihto-osat",
                "2.3 Sähköosat",
                "2.4 Tieto-osat",
                "2.5 Laiteosat",
                                "2.5.1 Siirtolaitteet",
                                "2.5.2 Tilalaitteet",
                "5 Käyttäjätehtävät",
                "5.1 Tilavarustus",
                                "5.1.1 Irtaimisto",
                                "5.1.2 Toiminnan kojeet ja laitteet",
                custom

! --- Rakennustuote 8.9.2003
values "rt20"  "",
                "1 Maa- ja aluerakennustuotteet",
                "11 Louhintatuotteet",
                "12 Pohjarakennustuotteet",
                "13 Maa-ainekset",
                "14 Maaputket",
                "15 Alue- ja pihapäällysteet",
                "16 Vihertuotteet",
                "17 Alue- ja pihavarusteet",
                "18 Alue- ja piharakenteiden tuotteet",
                "2 Runkorakennustuotteet",
                "21 Betonituotteet",
                "22 Metallituoteet",
                "23 Muuraustuotteet",
                "24 Puutavara",
                "25 Vesikatteet",
                "26 Rakennuslevyt",
                "27 Eristeet",
                "28 Rakennuselementit",
                "29 Väestönsuojatuotteet",
                "3 Täydentävät rakennustuotteet",
                "31 Ikkunat",
                "32 Ovet",
```

```
                "33 Julkisivutuotteet",
                "34 Väliseinätuotteet",
                "35 Alakatot",
                "36 Korokelattiat",
                "37 Tulisijatuotteet",
                "38 Täydennysvarusteet",
                "39 Helat ja kiinnikkeet",
                "4 Pintatuotteet",
                "41 Laatat",
                "42 Lattianpäällysteet",
                "43 Sisäverhoukset",
                "44 Liimat, laastit, tasoitteet",
                "45 Listat, nauhat, teipit",
                "46 Saumaustuotteet ja vedeneristeet",
                "47 Maalaustuotteet",
                "49 Erityiset pintatuotteet",
                "5 Rakennusvarusteet ja -kalusteet",
                "51 Yleisvarusteet",
                "52 Asuntovarusteet",
                "53 Toimisto- ja tuotantotilavarusteet",
                "54 Kiinteistövarusteet",
                "55 Julkistilojen varusteet",
                "56 Erityistilojen laitteet ja koneet",
                "6 Talotekniikkatuotteet",
                "61 LVI-tuotteet",
                "62 Sähkönsiirto- ja asennustuotteet",
                "63 Sähkökojeet ja -laitteet",
                "64 Sähköenergian tuotantolaitteet",
                "65 Tietotekniset tuotteet",
                "66 Siirtolaitteet",
                "7 Rakennusvälineet ja kalusto",
                "71 Työmaan rakennukset ja asennustarvikkeet",
                "72 Työvälineet",
                "73 Henkilöturvallisuustarvikkeet ja asusteet",
                "74 Työmaan käyttötarvikkeet",
                "75 Rakennustelineet ja työmaan koneistus",
                "76 Mittaus- ja laadunvalvontavälineet",
                "77 Erityiskalusto",
                "8 Kiinteistön hoito- ja toimintavarusteet",
                "81 Huonekalut",
                "82 Sisustustuotteet",
                "83 Vihersisustustuotteet",
                "84 Kodinkoneet",
                "85 Toimisto- ja teollisuusvarusteet",
                "86 Pihan ja vapaa-ajan tuotteet",
                "87 Kiinteistön hoitovälineet ja -tarvikkeet"
```

KÄYTTÖLIITTYMÄOHJELMA

```
! --- Otsikko
ui_dialog `Objektin asetukset`

! --- Perustyyli
ui_style 0,0

! --- Painikkeet
ui_current_page gs_ui_current_page
ui_page gs_ui_current_page

painikkeita=6                                                    ! määritellään painikkeiden
määrä (järkevä <=6)

dim gs_ui_but_txt[6]                                    ! painikkeiden tekstit
gs_ui_but_txt[1]="Rakenne"
gs_ui_but_txt[2]="Symboli"
gs_ui_but_txt[3]="Materiaalit"
gs_ui_but_txt[4]="Säädöt"
gs_ui_but_txt[5]="Tiedot"
gs_ui_but_txt[6]="Ohje"

painikelev=444/painikkeita                  ! määritellään painikkeen leveys ja tehdään painikkeet

if gs_ui_current_page=1 then ui_outfield gs_ui_but_txt[1], 0, 6, painikelev, 20, 2 else ui_button
ui_function, gs_ui_but_txt[1], 0, 4, painikelev, 20, 1
if painikkeita>1 then
              if gs_ui_current_page=2 then ui_outfield gs_ui_but_txt[2], 1*painikelev, 6, painikelev,
20, 2 else ui_button ui_function, gs_ui_but_txt[2], 1*painikelev, 4, painikelev, 20, 2
endif
if painikkeita>2 then
              if gs_ui_current_page=3 then ui_outfield gs_ui_but_txt[3], 2*painikelev, 6, painikelev,
20, 2 else ui_button ui_function, gs_ui_but_txt[3], 2*painikelev, 4, painikelev, 20, 3
endif
if painikkeita>3 then
              if gs_ui_current_page=4 then ui_outfield gs_ui_but_txt[4], 3*painikelev, 6, painikelev,
20, 2 else ui_button ui_function, gs_ui_but_txt[4], 3*painikelev, 4, painikelev, 20, 4
endif
if painikkeita>4 then
              if gs_ui_current_page=5 then ui_outfield gs_ui_but_txt[5], 4*painikelev, 6, painikelev,
20, 2 else ui_button ui_function, gs_ui_but_txt[5], 4*painikelev, 4, painikelev, 20, 5
endif
if painikkeita>5 then
              if gs_ui_current_page=6 then ui_outfield gs_ui_but_txt[6], 5*painikelev, 6, painikelev,
20, 2 else ui_button ui_function, gs_ui_but_txt[6], 5*painikelev, 4, painikelev, 20, 6
endif


! --- Sivut
sarakelev=444/4
rivikor=20
outlev=sarakelev-5
outkor=rivikor-3
inlev=sarakelev-20
inkor=rivikor-3

! malli
!              rivinro=2
```

```
!              ui_outfield "VASEN", 0, rivinro*rivikor, outlev,outkor
!                              rivinro=rivinro+1
!              ui_outfield "VASEN", 0, rivinro*rivikor, outlev,outkor
!                              ui_infield "ro20", sarakelev,rivinro*rivikor, inlev,inkor
!                                             rivinro=rivinro+1
!              rivinro=2
!              ui_outfield "OIKEA", 2*sarakelev, rivinro*rivikor, outlev,outkor
!                              rivinro=rivinro+1
!              ui_outfield "OIKEA", 2*sarakelev, rivinro*rivikor, outlev,outkor
!                              ui_infield "ro20", 3*sarakelev,rivinro*rivikor, inlev,inkor
!                                             rivinro=rivinro+1

ui_page 1 ! --- sivu ------------------
               rivinro=2
               ui_style 0,1
                              ui_outfield "Mitat", 0, rivinro*rivikor, outlev,outkor
               ui_style 0,0
               rivinro=rivinro+1
               ui_outfield "Pituus/leveys", 0, rivinro*rivikor, outlev,outkor
                              ui_infield "a", sarakelev,rivinro*rivikor, inlev,inkor
                                             rivinro=rivinro+1
               ui_outfield "Leveys/syvyys", 0, rivinro*rivikor, outlev,outkor
                              ui_infield "b", sarakelev,rivinro*rivikor, inlev,inkor
                                             rivinro=rivinro+1
               ui_outfield "Korkeus/paksuus",              0, rivinro*rivikor, outlev,outkor
                              ui_infield "zzyzx", sarakelev,rivinro*rivikor, inlev,inkor
                                             rivinro=rivinro+1
               rivinro=rivinro+1

! --- Yleisen objektin pohjamuoto
               ui_outfield "Muoto",         0, rivinro*rivikor, outlev,outkor
                              ui_infield "muoto", sarakelev,rivinro*rivikor, inlev,inkor
               rivinro=rivinro+1

!              ui_outfield "Istuimen korkeus",0, rivinro*rivikor, outlev,outkor
!                              ui_infield "gs_seat_height", sarakelev,rivinro*rivikor, inlev,inkor
!                                             rivinro=rivinro+1
!              ui_outfield "Jalan korkeus",                0, rivinro*rivikor, outlev,outkor
!                              ui_infield "gs_leg_height", sarakelev,rivinro*rivikor, inlev,inkor
!                                             rivinro=rivinro+1

               rivinro=2
               ui_style 0,1
                              ui_outfield "Tunnus", 2*sarakelev, rivinro*rivikor, outlev,outkor
               ui_style 0,0
               rivinro=rivinro+1
               ui_outfield "Annettu tunnus", 2*sarakelev, rivinro*rivikor, outlev,outkor
                              ui_infield "tunnus", 3*sarakelev,rivinro*rivikor, inlev,inkor
                                             rivinro=rivinro+1
               ui_outfield "on tyypiltään", 2*sarakelev, rivinro*rivikor, outlev,outkor
                              ui_infield "opeta_tunnus", 3*sarakelev,rivinro*rivikor, inlev,inkor
                                             rivinro=rivinro+1

!              rivinro=rivinro+1
!              ui_style 0,1
!                              ui_outfield "Helat ja varusteet",           2*sarakelev, rivinro*rivikor,
outlev,outkor
!              ui_style 0,0
!              rivinro=rivinro+1
```

```
!               ui_outfield "Rakenne",       2*sarakelev, rivinro*rivikor, outlev,outkor
!                               ui_infield "rakenne", 3*sarakelev,rivinro*rivikor, inlev,inkor
!                                       rivinro=rivinro+1
!               ui_outfield "Heloitus",      2*sarakelev, rivinro*rivikor, outlev,outkor
!                               ui_infield "heloitus", 3*sarakelev,rivinro*rivikor, inlev,inkor
!                                       rivinro=rivinro+1
!               ui_outfield "Lukitus",       2*sarakelev, rivinro*rivikor, outlev,outkor
!                               ui_infield "lukitus", 3*sarakelev,rivinro*rivikor, inlev,inkor
!                                       rivinro=rivinro+1
!               ui_outfield "Liittymärakenne",              2*sarakelev, rivinro*rivikor, outlev,outkor
!                               ui_infield "liitosrakenne", 3*sarakelev,rivinro*rivikor, inlev,inkor
!                                       rivinro=rivinro+1
!               ui_outfield "Varuste",       2*sarakelev, rivinro*rivikor, outlev,outkor
!                               ui_infield "varuste", 3*sarakelev,rivinro*rivikor, inlev,inkor
!                                       rivinro=rivinro+1
!               ui_outfield "Laite",         2*sarakelev, rivinro*rivikor, outlev,outkor
!                               ui_infield "laite", 3*sarakelev,rivinro*rivikor, inlev,inkor
!                                       rivinro=rivinro+1


ui_page 2 ! --- sivu ------------------
                rivinro=2
                ui_style 0,1
                                ui_outfield "2D-esitys",     0, rivinro*rivikor, outlev,outkor
                ui_style 0,0
                ui_infield "gs_detlevel_2d", sarakelev,rivinro*rivikor, inlev,inkor
                                rivinro=rivinro+1
                ui_outfield "Viivatyyppi",   0, rivinro*rivikor, outlev,outkor
                                ui_infield "gs_line_type", sarakelev,rivinro*rivikor, inlev,inkor
                                        rivinro=rivinro+1
                ui_outfield "Ääriviivan kynä",              0, rivinro*rivikor, outlev,outkor
                                ui_infield "gs_cont_pen", sarakelev,rivinro*rivikor, inlev,inkor
                                        rivinro=rivinro+1
                ui_style 0,1
                                ui_outfield "Täyte symbolissa",             0, rivinro*rivikor,
outlev,outkor
                ui_style 0,0
                                ui_infield "gs_fill", sarakelev,rivinro*rivikor, inlev,inkor
                                        rivinro=rivinro+1
                if gs_fill then
                                ui_outfield "Täytteen tyyppi", 0, rivinro*rivikor, outlev,outkor
                                ui_infield "gs_fill_type", sarakelev,rivinro*rivikor, inlev,inkor
                                                        rivinro=rivinro+1
                                ui_outfield "Täytteen kynä", 0, rivinro*rivikor, outlev,outkor
                                ui_infield "gs_fill_pen", sarakelev,rivinro*rivikor, inlev,inkor
                                                        rivinro=rivinro+1
                                ui_outfield "Täytteen taustakynä",0, rivinro*rivikor, outlev,outkor
                                ui_infield "gs_back_pen", sarakelev,rivinro*rivikor, inlev,inkor
                                                        rivinro=rivinro+1
                endif

                rivinro=2
                ui_style 0,1
                ui_outfield "Näytä tilantarve", 2*sarakelev, rivinro*rivikor, outlev,outkor
                ui_style 0,0
                                ui_infield "gs_min_space", 3*sarakelev,rivinro*rivikor, inlev,inkor
                                        rivinro=rivinro+1
                if gs_min_space then
                                ui_outfield "Edessä", 2*sarakelev, rivinro*rivikor, outlev,outkor
```

```
                              ui_infield "msfront", 3*sarakelev,rivinro*rivikor, inlev,inkor
                                                        rivinro=rivinro+1
                              ui_outfield "Oikealla", 2*sarakelev, rivinro*rivikor, outlev,outkor
                              ui_infield "msright", 3*sarakelev,rivinro*rivikor, inlev,inkor
                                                        rivinro=rivinro+1
                              ui_outfield "Vasemmalla", 2*sarakelev, rivinro*rivikor, outlev,outkor
                              ui_infield "msleft", 3*sarakelev,rivinro*rivikor, inlev,inkor
                                                        rivinro=rivinro+1
                              ui_outfield "Takana", 2*sarakelev, rivinro*rivikor, outlev,outkor
                              ui_infield "msrear", 3*sarakelev,rivinro*rivikor, inlev,inkor
                                                        rivinro=rivinro+1
                              ui_outfield "Täytteen tyyppi", 2*sarakelev, rivinro*rivikor, outlev,outkor
                              ui_infield "msfilltype", 3*sarakelev,rivinro*rivikor, inlev,inkor
                                                        rivinro=rivinro+1
                              ui_outfield "Täytteen kynä", 2*sarakelev, rivinro*rivikor, outlev,outkor
                              ui_infield "msfillpen", 3*sarakelev,rivinro*rivikor, inlev,inkor
                                                        rivinro=rivinro+1
                    ui_outfield "Täytteen taustakynä",2*sarakelev,rivinro*rivikor, outlev,outkor
                              ui_infield "msfillbackpen", 3*sarakelev,rivinro*rivikor, inlev,inkor
                                                        rivinro=rivinro+1
                    endif

                    rivinro=10
                    ui_style 0,1
                    ui_outfield "Teksti symbolissa", 0, rivinro*rivikor, outlev,outkor
                    ui_style 0,0
                              ui_infield "txt_on", sarakelev,rivinro*rivikor, 20,inkor
                                        rivinro=rivinro+1
                    if txt_on then
                    ui_outfield "<=1:", sarakelev+22,(rivinro-1)*rivikor, inlev/3,inkor
                    ui_infield "txt_raja", sarakelev+outlev/2,(rivinro-1)*rivikor, inlev/2,inkor
                              ui_outfield "Kierto",        0, rivinro*rivikor, outlev/2,outkor
                              ui_infield "txtr", outlev/2,rivinro*rivikor, inlev/2,inkor
                    ui_outfield "Kiinnitys",     sarakelev, rivinro*rivikor, outlev/2,outkor
                    ui_infield "txta", sarakelev+outlev/2,rivinro*rivikor, inlev/2,inkor
                                                        rivinro=rivinro+1

                              ui_outfield "Muotoilu",      0, rivinro*rivikor, outlev,outkor
                              ui_infield "gs_font_type", sarakelev,rivinro*rivikor, inlev,inkor
                              ui_infield "gs_font_pen", 2*sarakelev,rivinro*rivikor, inlev/2,inkor
                    ui_infield "gs_font_size", 2*sarakelev+inlev/2+10,rivinro*rivikor, inlev/2,inkor
                              ui_infield "gs_font_style", 3*sarakelev,rivinro*rivikor, inlev,inkor

                              rivinro=10
                              ui_outfield "Teksti",        2*sarakelev, rivinro*rivikor, outlev,outkor
                              ui_infield "txt", 3*sarakelev,rivinro*rivikor, inlev,inkor
                                                        rivinro=rivinro+1
                              ui_outfield "on tyypiltään", 2*sarakelev, rivinro*rivikor, outlev,outkor
                              ui_infield "opeta_txt", 3*sarakelev,rivinro*rivikor, inlev,inkor
                                                        rivinro=rivinro+1
                    endif

ui_page 3 ! --- sivu ------------------
                    rivinro=2
                    ui_outfield "Päämateriaali",               0, rivinro*rivikor, outlev,outkor
                              ui_infield "gs_frame_mat", sarakelev,rivinro*rivikor, inlev,inkor
                                        rivinro=rivinro+1
!                   ui_outfield "Istuinmateriaali",            0, rivinro*rivikor, outlev,outkor
!                             ui_infield "gs_seat_mat", sarakelev,rivinro*rivikor, inlev,inkor
```

```
!                                           rivinro=rivinro+1
!              ui_outfield "_materiaali",   0, rivinro*rivikor, outlev,outkor
!                              ui_infield "gs_seat_mat", sarakelev,rivinro*rivikor, inlev,inkor
!                                           rivinro=rivinro+1
!              ui_outfield "_materiaali",   0, rivinro*rivikor, outlev,outkor
!                              ui_infield "gs_seat_mat", sarakelev,rivinro*rivikor, inlev,inkor
!                                           rivinro=rivinro+1
!              ui_outfield "_materiaali",   0, rivinro*rivikor, outlev,outkor
!                              ui_infield "gs_seat_mat", sarakelev,rivinro*rivikor, inlev,inkor
!                                           rivinro=rivinro+1
!              ui_outfield "_materiaali",   0, rivinro*rivikor, outlev,outkor
!                              ui_infield "gs_seat_mat", sarakelev,rivinro*rivikor, inlev,inkor
!                                           rivinro=rivinro+1
!              ui_outfield "_materiaali",   0, rivinro*rivikor, outlev,outkor
!                              ui_infield "gs_seat_mat", sarakelev,rivinro*rivikor, inlev,inkor
!                                           rivinro=rivinro+1
!              ui_outfield "_materiaali",   0, rivinro*rivikor, outlev,outkor
!                              ui_infield "gs_seat_mat", sarakelev,rivinro*rivikor, inlev,inkor
!                                           rivinro=rivinro+1
!              ui_outfield "_materiaali",   0, rivinro*rivikor, outlev,outkor
!                              ui_infield "gs_seat_mat", sarakelev,rivinro*rivikor, inlev,inkor
!                                           rivinro=rivinro+1
!              ui_outfield "_materiaali",   0, rivinro*rivikor, outlev,outkor
!                              ui_infield "gs_seat_mat", sarakelev,rivinro*rivikor, inlev,inkor
!                                           rivinro=rivinro+1
!              ui_outfield "_materiaali",   0, rivinro*rivikor, outlev,outkor
!                              ui_infield "gs_seat_mat", sarakelev,rivinro*rivikor, inlev,inkor
!                                           rivinro=rivinro+1


ui_page 4 ! --- sivu ------------------
               rivinro=2
               ui_style 0,1
               ui_outfield "3D-esitys",     0, rivinro*rivikor, outlev,outkor
               ui_style 0,0
                              ui_infield "gs_detlevel_3d", sarakelev,rivinro*rivikor, inlev,inkor
                                           rivinro=rivinro+1
               ui_outfield "Tarkkuus",      0, rivinro*rivikor, outlev,outkor
                              ui_infield "gs_resol", sarakelev,rivinro*rivikor, inlev,inkor
                                           rivinro=rivinro+1
               ui_outfield "Heittovarjot",  0, rivinro*rivikor, outlev,outkor
                              ui_infield "gs_shadow", sarakelev,rivinro*rivikor, inlev,inkor
                                           rivinro=rivinro+1
               ui_outfield "Rautalankamalli",              0, rivinro*rivikor, outlev,outkor
                              ui_infield "varaus3d", sarakelev,rivinro*rivikor, inlev,inkor
                                           rivinro=rivinro+1
               ui_outfield "Ääriviivankynä",               0, rivinro*rivikor, outlev,outkor
                              ui_infield "gs_view_pen", sarakelev,rivinro*rivikor, inlev,inkor
                                           rivinro=rivinro+1

               ! objektin kallistaminen
               rivinro=rivinro+1
               ui_outfield "Kallistus (y)",                0, rivinro*rivikor, outlev,outkor
                              ui_infield "tilt", sarakelev,rivinro*rivikor, inlev,inkor
                                           rivinro=rivinro+1
               ui_outfield "Kallistus (x)",                0, rivinro*rivikor, outlev,outkor
                              ui_infield "angle", sarakelev,rivinro*rivikor, inlev,inkor
                                           rivinro=rivinro+1
```

```
               rivinro=2
               if gs_detlevel_3d="Kutsuttu" then
                              ui_outfield "Kutsu objektia", 2*sarakelev, rivinro*rivikor, outlev,outkor
                              ui_infield "kutsuttu", 3*sarakelev,rivinro*rivikor, inlev,inkor
                                                          rivinro=rivinro+1
               ui_outfield "Muuta origo (x,y,z)", 2*sarakelev, rivinro*rivikor, outlev,outkor
                              ui_infield{2} korjaa[1], 3*sarakelev,rivinro*rivikor, inlev/3,inkor
               ui_infield{2} korjaa[2], 3*sarakelev+inlev/3,rivinro*rivikor, inlev/3,inkor
               ui_infield{2} korjaa[3], 3*sarakelev+2*inlev/3,rivinro*rivikor, inlev/3,inkor
                                                          rivinro=rivinro+1
               ui_outfield "ja kierrä",     2*sarakelev, rivinro*rivikor, outlev,outkor
               ui_infield{2} korjaakulma, 3*sarakelev,rivinro*rivikor, inlev,inkor
               endif

ui_page 5 ! --- sivu ------------------
               rivinro=2
               ui_style 0,1
               ui_outfield "Annettu tunnus",              0, rivinro*rivikor, outlev,outkor
               ui_style 0,0
                              ui_infield "tunnus", sarakelev,rivinro*rivikor, inlev,inkor
                                                 rivinro=rivinro+1
               ui_outfield "Tuotetiedot",   0, rivinro*rivikor, outlev,outkor
               ui_outfield "automaattisesti",             sarakelev, rivinro*rivikor, outlev-20,outkor
                              ui_infield "opeta_tieto", 2*sarakelev-20,rivinro*rivikor, 20,inkor
                                                 rivinro=rivinro+1
               ui_outfield "Tyyppitunnus",  0, rivinro*rivikor, outlev,outkor
                              ui_infield "tyyppi", sarakelev,rivinro*rivikor, inlev,inkor
                                                 rivinro=rivinro+1
               ui_outfield "Kätisyys",      0, rivinro*rivikor, outlev,outkor
                              ui_infield "katisyys", sarakelev,rivinro*rivikor, inlev,inkor
                                                 rivinro=rivinro+1
               ui_outfield "Nimi",          0, rivinro*rivikor, outlev/2,outkor
                              ui_infield "nimi", sarakelev,rivinro*rivikor, inlev,inkor
                              ui_outfield "muu",           outlev/2, rivinro*rivikor, outlev/2-20,outkor
                              ui_infield "muunimi", outlev-20, rivinro*rivikor, 20,outkor
                                                 rivinro=rivinro+1
               ui_outfield "Kuvaus",        0, rivinro*rivikor, outlev,outkor
                              ui_infield "kuvaus", sarakelev,rivinro*rivikor, inlev,inkor
                                                 rivinro=rivinro+1
               ui_outfield "Erityistieto",  0, rivinro*rivikor, outlev,outkor
                              ui_infield "tieto", sarakelev,rivinro*rivikor, inlev,inkor
                                                 rivinro=rivinro+1
               ui_outfield "Asennuskorkeus",              0, rivinro*rivikor, outlev,outkor
                              ui_infield "asennuskorkeus", sarakelev,rivinro*rivikor, inlev,inkor
                                                 rivinro=rivinro+1
               ui_outfield "Korkeusasema",  0, rivinro*rivikor, outlev,outkor
                              ui_infield "korkeusasema", sarakelev,rivinro*rivikor, inlev,inkor
                                                 rivinro=rivinro+1
               ui_outfield "Sijainti",      0, rivinro*rivikor, outlev,outkor
                              ui_infield "sijainti", sarakelev,rivinro*rivikor, inlev,inkor
                                                 rivinro=rivinro+1
               ui_outfield "Asento",        0, rivinro*rivikor, outlev,outkor
                              ui_infield "asento", sarakelev,rivinro*rivikor, inlev,inkor
                                                 rivinro=rivinro+1


               rivinro=2
               ui_style 0,1
               ui_outfield "on tyypiltään",               2*sarakelev, rivinro*rivikor, outlev,outkor
               ui_style 0,0
```

```
                              ui_infield "opeta_tunnus", 3*sarakelev,rivinro*rivikor, inlev,inkor
                                           rivinro=rivinro+2
               ui_outfield "Rakennusosa",   2*sarakelev, rivinro*rivikor, outlev,outkor
                              ui_infield "ro20", 3*sarakelev,rivinro*rivikor, inlev,inkor
                                           rivinro=rivinro+1
               ui_outfield "Rakennustuote",               2*sarakelev, rivinro*rivikor, outlev,outkor
                              ui_infield "rt20", 3*sarakelev,rivinro*rivikor, inlev,inkor
                                           rivinro=rivinro+1
               ui_outfield "Viite",         2*sarakelev, rivinro*rivikor, outlev,outkor
                              ui_infield "viite", 3*sarakelev,rivinro*rivikor, inlev,inkor
                                           rivinro=rivinro+1
               ui_outfield "Versio",        2*sarakelev, rivinro*rivikor, outlev,outkor
                              ui_infield "versio", 3*sarakelev,rivinro*rivikor, inlev,inkor
                                           rivinro=rivinro+1
               ui_outfield "Status",        2*sarakelev, rivinro*rivikor, outlev,outkor
                              ui_infield "status", 3*sarakelev,rivinro*rivikor, inlev,inkor
                                           rivinro=rivinro+1

ui_page 6 ! --- sivu ------------------
               rivinro=2
               ui_outfield "Peruskirjasto 11 -kirjastoelementti \n© M.A.D. Oy, 2007 \nversio 070605", 0,
               rivinro*rivikor, 3*inlev,3*inkor
                              rivinro=rivinro+3
               ui_button ui_link,"M.A.D. Oy", 0, rivinro*rivikor, outlev, outkor, 0,
               "http://www.mad.fi/mad/kirjastot/peruskirjasto11/"
               !ui_pict 1, 2*sarakelev, 2*rivikor

!              ui_outfield "Tietoja", 0, rivinro*rivikor, outlev,outkor
!                             rivinro=rivinro+1
!              ui_outfield "- [tietoja]", 0, rivinro*rivikor, 440, 255-rivinro*rivikor

end
```

KOMMENTTI

```
#http://www.mad.fi/mad/kirjastot/peruskirjasto11/
© M.A.D. Oy, 2007

    P K 1 1

 o b j e k t i
```