

MODULAARISEN TUOTTEEN 3D-MARKKINOINTISOVELLUS

LAHDEN AMMATTIKORKEAKOULU
Mediatekniikan koulutusohjelma
Teknisen visualisoinnin suuntautumisvaihtoehto
Opinnäytetyö
31.5.2007
Satu Härmä

**Lahden ammattikorkeakoulu
Mediatekniikan koulutusohjelma**

HÄRMÄ, SATU: Modulaarisen tuotteen 3d-markkinointisovellus

Teknisen visualisoinnin opinnäytetyö
54 sivua, 2 liitesivua

Kevät 2007

TIIVISTELMÄ

Tässä opinnäytetyössä pohditaan niitä olemassa olevia ja kehittyviä 3d-tekniikoita, joita voitaisiin käyttää modulaarisen tuotteen markkinointisovelluksen tuottamisessa. Modulaarisuuden lajien ja kolmen vapaasti saatavilla olevan kuluttajasovelluksen visuaalisten rakenteiden havainnoimisen avulla selvitetään niitä asioita, jotka liittyvät tällaisiin markkinointisovelluksiin.

Lisäksi esitellään kolme tekniikkaa, joilla voidaan tuottaa erilaisiin käyttöympäristöihin markkinointisovellus: X3D kolmiulotteisuuden esittämiseen internet-ympäristössä, Java-ohjelmointi perinteisiin kiinteiden työasemien markkinointisovelluksiin ja SolidWorksin makro-ohjelmointi teknisen tuotteen sovelluksiin.

Tässä työssä syvennyttään tarkemmin siihen, millä lailla SolidWorksin ja makrojen avulla voidaan luoda modulaarisen tuotteen markkinointisovellus kevyellä ohjelmoinnilla ja mitä ominaisuuksia markkinointisovelluksessa voi olla. Tätä tekniikkaa hyödyntäen tuotetaan Case-osiossa yritykselle sovelluskokeilu, jossa kootaan modulaarinen betonimuotti SolidWorksin API-rajapinnan avulla Visual Basic for Application -ohjelmointikielellä.

Avainsanat: kolmiulotteisuus, modulaarisuus, markkinointisovellus, SolidWorks, Visual Basic for Application

**Lahti University of Applied Sciences
Faculty of Technology**

HÄRMÄ, SATU: 3D Marketing Application for Modular Products

Bachelor's Thesis in Visualization Engineering, 54 pages, 2 appendices

Spring 2007

ABSTRACT

This thesis deals with how to use 3D techniques to produce marketing applications for modular products.

The work examines different types of modularity and uses three free marketing applications as examples to study and introduce visual constructions of 3D marketing applications. In addition, it introduces three techniques of producing marketing applications for different environments: X3D for on-line applications, Java programming language for traditional off-line use and SolidWorks with macro programming for applications of technical products.

With light programming it is possible to produce a marketing application in SolidWorks. By utilizing this technique in the Case study, an experimental application was produced, where modular concrete moulds are assembled. The application was programmed with the API interface and the Visual Basic for Application programming language.

Keywords: three dimensional, modularity, marketing application, SolidWorks, Visual Basic for Application

SISÄLLYS

1	JOHDANTO	1
2	MODULAARISUUS JA KONFIGUROINTI	2
	2.1 Modulaarisuus	2
	2.2 Tuotekonfigurointi.....	3
3	MARKKINOINTISOVELLUS	4
	3.1 Markkinoinnin tarkoitus	4
	3.2 Markkinointisovelluksen hyödyt	5
	3.3 Sovelluksen käyttöympäristö	6
	3.4 Tuotteen 3d-visualisointi	7
	3.5 Tekniikat.....	9
4	MODULAARISTEN TUOTTEIDEN MARKKINOINTISOVELLUKSIDA	13
	4.1 Johdanto	13
	4.2 Vapaasti saatavilla olevia sovelluksia	13
	4.21 IKEA Home Planner Kitchen	13
	4.22 Lego Digital Designer	16
	4.23 Rakenna oma ergonominen työasemasi	18
	4.3 Moduulien visuaalinen sijoittuminen.....	20
5	SOLIDWORKS SOVELLUKSEN TUOTTAMISESSA	21
	5.1 Yleisesittely.....	21
	5.2 Soveltuvuus markkinointisovellukseen.....	22
	5.3 Visual Basic for Application (VBA)	23
	5.4 Makrot.....	25
	5.5 API-funktiot	27
	5.6 Sovelluksen käyttöliittymä.....	29
	5.7 Näkymän hallinta sovelluksessa.....	29
	5.8 Modulaarisuuden tyypillisten ominaisuuksien toteuttaminen.....	30
	5.81 Moduulien tuominen ja poistaminen.....	30
	5.82 Vaihtokelpoisuus ja käyttö useammassa tuotteessa	31
	5.83 Parametrinen modulaarisuus ja väylämodulaarisuus	32

5.84	Rakennuslohkomodulaarisuus.....	33
5.9	Tulostaminen ja tallentaminen sovelluksessa	34
5.91	Tulostettavia ja tallennettavia tietoja	34
5.92	Lomakkeeseen tulostaminen	35
5.93	Näkymän tulostaminen.....	35
5.94	Tiedostojen tallentaminen.....	35
5.95	Teknisten piirustusten luominen.....	36
5.96	Tekstitiedostoon tulostaminen	36
6	CASE: Betonimuottien kokoonpanosovellus	38
6.1	Esittely.....	38
6.2	Tavoitteet sovellukselle	39
6.3	Sovelluksen toteuttaminen	40
6.4	Sovelluksen käyttäminen.....	41
6.5	Jatkokehitysehdotelmia.....	42
6.6	Casen arviointi.....	43
7	YHTEENVETO.....	44
	LÄHTEET	45
	KUVALÄHTEET	51
	TAULUKKOLÄHTEET	52
	LIITTEET	53

TERMIT JA LYHENTEET

API, Application Programming Interface, on ohjelmointirajapinta, jonka avulla eri ohjelmat vaihtavat tietoja keskenään.

Digitaalinen markkinointi on markkinointia, jonka kanavina ovat Internet, sähköposti, mobiililaitteet ja digi-TV.

IDE, Integrated Development Environment, integroitu ohjelmointiympäristö eli joukko ohjelmia, jolla ohjelmoija toteuttaa ohjelmistoa. Sisältää esimerkiksi tekstieditorin ja ohjelmointikielen kääntäjän.

Java 3D, Java-ohjelmointikielen ohjelmointirajapinta, laajennus kolmiulotteista grafiikkaa sisältäviin ohjelmiin.

Konfiguraattori on järjestelmä, jonka avulla tuotetaan konfiguraatioita.

Konfiguroinnissa luodaan tuotteesta eri muunnoksia, tuotevariantteja.

Modulaarisuus on varioitavuutta, mahdollisuutta erilaisiin moduuleista koostuviin yhdistelmiin.

Moduuli on itsenäinen osa, josta kootaan rajapintojen avulla erilaisia tuotevariantteja.

OLE Automation, Object Linking and Embedding Automation, on Microsoftin tekniikka, jolla makrojen avulla isäntäohjelman päältä pystytään ohjaamaan toista ohjelmaa.

OpenGL, Open Graphics Library, avoin ja laitteistoriippumaton ohjelmointirajapinta graafisia toimintoja varten.

PDM, Product Design Management eli tuotetiedon hallintajärjestelmä.

Sovelma, Applet, on selaimen yhteydessä suoritettava epäitsenäinen ohjelma.

VBA, Visual Basic for Application, on Microsoftin sovellusohjelmissa ja makroissa käytetty tulkittava ohjelmointikieli.

VRML, Virtual Reality Modelling Language, on tiedostomuoto, jolla esitetään kolmiulotteista interaktiivista grafiikkaa erityisesti internetissä.

X3D on ohjelmistostandardi kolmiulotteisen sisällön esittämiseen ja kommunikointiin reaaliaikaisesti

XML, eXtensible Markup Language, on merkkauskieli, jolla tiedon merkitys on kuvattavissa tiedon sekaan, eli metakieli, joka sisältää tietoa tiedosta.

1 JOHDANTO

Kolmiulotteisuus on pelimaailmassa ja teknisessä suunnittelussa nykyaikaa. Kuitenkaan sitä ei ole vielä valjastettu markkinoinnin käyttöön eikä kaikkien ulottuville laajemmin sovelluksien, joissa kolmiulotteisuuden sekä viihteelliset että rakenteen kuvauksen hyödyt olisivat tarpeellisia. Kolmiulotteisuudella on vielä uutuusarvoa.

Tässä opinnäytetyössä pohditaan niitä olemassa olevia ja kehittyviä 3d-tekniikoita, joita voitaisiin käyttää markkinoinnissa modulaarisuuden esittämiseen nykykäyttöä laaja-alaisemmin. Yhtenä keinona modulaarisen tuotteen markkinoinnin ja tekniikan ymmärtämiseen on verkossa vapaasti saatavilla olevien 3d-markkinointisovelluksien visuaalisen rakenteen havainnointi ja modulaarisuuden rakenteiden ilmentyminen niissä.

Yksi kolmiulotteisuuden esittämiseen liittyvistä haasteista on verkkosovellukset. Vaikka jatkuva kaistanleveyden kasvu tuo koko ajan uusia ulottuvuuksia internetin käyttöön, vaatii verkkokäyttö mahdollisimman sujuvia sovelluksia. Samalla vaaditaan yhä suurempia elämyksiä, näyttävyyttä ja vuorovaikutusta.

Modulaarisen 3d-markkinointisovelluksen tuottamiseen esitellään tässä työssä kolme erilaisiin käyttöympäristöihin soveltuvaa tekniikkaa. X3D on kehitteillä oleva standardi kolmiulotteisuuden esittämiseen internet-ympäristössä, Java-ohjelmointikieli perinteisille kiinteille työasemille tehtyjen sovellusten ohjelmointiin ja SolidWorksin makro-ohjelmointi teknisen tuotteen markkinointisovelluksen tuottamiseen.

Tässä työssä syvennyttään tarkemmin siihen, millä lailla SolidWorksin ja makrojen avulla voidaan luoda modulaarisen tuotteen markkinointisovellus kevyellä ohjelmoinnilla ja mitä ominaisuuksia markkinointisovelluksessa voi olla. Tätä tekniikkaa hyödyntäen tuotetaan Case-osiossa yritykselle sovelluskokeilu, jossa kootaan modulaarinen betonimuotti SolidWorksin API-rajapinnan avulla Visual Basic for Application -kielellä.

2 MODULAARISUUS JA KONFIGUROINTI

2.1 Modulaarisuus

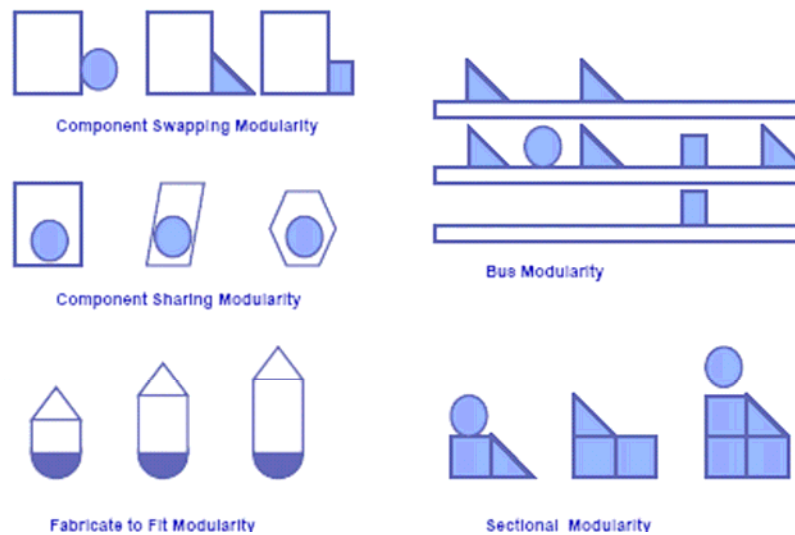
Moduuli terminä esiintyy useissa eri yhteyksissä. Se liitetään muun muassa avaruustekniikkaan, mielen tutkimukseen ja algebraan. Ohjelmoinnissa modulaarisuus on yksittäisistä, itsenäisistä ohjelmamoduuleista koostuvaa ohjelmointia, roolipeleissä erillisiä seikkailuja. (Moduuli 2007.) Kaikessa modulaarisuus eli moduuleista koostuvuus on kuitenkin yhteydessä kokonaisuuden paloittelavuuteen.

Wikipedia määrittelee moduulin tekniikassa itsenäiseksi osaksi, joista kootaan erilaisia kokonaisuuksia (Moduuli 2007). Ominaisuutena modulaarisuus on varioitavuutta, mahdollisuutta erilaisiin yhdistelmiin eli valmiiksi suunniteltujen moduulien avulla tuotevarianttien luomista asiakkaiden vaatimusten mukaan (Modulointi 2006).

Modulaarisuus parantaa tuotteiston hallintaa, kun valmistaja tuottaa aina samoja, keskenään yhteensopivia elementtejä. Moduuleissa tulee olla selkeät rajapinnat, joista se kiinnittyy toisiin moduuleihin. Moduulin tulee toteuttaa yksi tai useampi toiminto, ja moduulien välisten vuorovaikutusten tulee olla tarkasti määriteltäviä ja minimoituja. (Modulointi 2006 ja Sarinko 1999.)

Modulaarisuus jaetaan yleisesti viiteen lajiin (Kuva 1). Yksi tuote voi silti koostua useasta eri modulaarisuuden lajista.

- **Moduulien vaihtokelpoisuus** (Component Swapping Modularity): Moduuli on vaihdettavissa samaan perusyksikköön.
- **Moduulien käyttö useammassa kohteessa** (Component Sharing Modularity): Samaa moduulia käytetään useissa perusyksiköissä.
- **Parametrinen modulaarisuus** (Fabricate to Fit Modularity): Moduulin parametrisiä arvoja muutetaan tapauskohtaisesti.
- **Väylämodulaarisuus** (Bus Modularity): Perusyksikköön liitetään erilaisia moduuleja vapaassa järjestyksessä.
- **Rakennuslohkomodulaarisuus** (Sectional Modularity): Moduuleja yhdistellään rajapintojen avulla mihin tahansa järjestykseen. Ilmiötä voidaan kutsua myös legomoduloinniksi. (Sarinko 1999.)



KUVA 1. Modulaarisuuden viisi lajia (Modulointi 2006).

2.2 Tuotekonfigurointi

Moduuleista kootaan asiakkaan tarvitsema kokonaisuus. Asiakkaalle koostetut kokonaisuudet ovat tuotteen erilaisia konfiguraatioita, tuotevariantteja. (Modulointi 2006.)

Yritysmailmassa tuotteiden nopea konfigurointi on yhä tärkeämpää. Kiristyvässä kilpailussa tuotteiden lyhyet elinkaaret aiheuttavat käytännössä sen, että asiakkaille valmistetaan tuotteet yksilöinä. Näin ollen tehokas konfigurointi edellyttää, että tuote on modulaarinen. (Tuotekonfigurointi 2006.)

Konfiguraatioihin liittyy valtava määrä tietoa. Tämän tiedon hallintaan on kehitetty erilaisia konfiguraattoreita, sovelluksia konfiguraatioiden tuottamiseen. Näitä konfiguraattoreita voidaan jopa tarjota suoraan loppukäyttäjien käytettäväksi vaikkapa Internetissä (Tuotekonfigurointi 2006). Tehokkaan asiakasohjautuvuuden lisäksi tuotekonfiguraattoreilla vähennetään inhimillisten virheiden määrää, kun sovellus ottaa automaattisesti huomioon kaikki sille asetetut vaatimukset (Kerätär 2005). Konfiguraattori voi olla joko järjestelmä tai vaikkapa paperilla oleva tilauskaavake, josta rastitetaan halutut osat (Sarinko 1999).

Konfiguraattorit ovat siis yksilöllisten tuotteiden suunnitteluautomaatteja. Näille sovelluksille on yhteistä suunnittelurutiinien automatisointi, jolloin sovelluksen suorittamista rutiineista vapautuu suunnittelijalle aikaa. Sovelluksiin kuuluu myös yksilöllisistä konfiguraatioista dokumenttien luominen, kuten tekniset piirustukset ja pdf-dokumentit tuotteiden osaluetteloista. Sovellus voi olla niin helppokäyttöinen, että tuotteen mukauttamisen asiakkaan tarpeisiin ei vaadita suunnittelu- tai 3D-

osaamista (Tuotekonfigurointi 2006). CAD-tuotekonfiguraattori voi yksinkertaisimmillaan olla sovellus, jossa tuotteen mittoja vaihdetaan käyttäjän antaman käskyn mukaan (Tuotekonfigurointi 2006).

Konfiguraattorien käytön etuja ovat Saringon (1999) mukaan

- modulaaristen osien ja niiden välisten yhteyksien ymmärtäminen
- standardoitujen ja vain oleellisten osien käyttö
- tilauksen läpimenoajan lyheneminen
- myyjän muistamistarpeen väheneminen
- tilausten virheettömyys ja dokumentoinnin oikeellisuus
- yrityksen kilpailutilanteen ja markkinaosuuden parantaminen.

Tässä opinnäytetyössä tuotekonfiguraattoreista suunnitteluautomaatin ja markkinointisovelluksen erona pidetään pääasiassa tuotetiedon määrää, sovelluksen kytkentää valmistuksen apuvälineisiin ja loppukäyttäjän ammattitaitoa. Termit menevät käytännössä usein osittain päällekkäin. Suunnitteluautomaatti on käytössä ammattisuunnittelijoilla ja se sisältää mahdollisimman täydellisen tietomäärän suunniteltavasta tuotteesta sekä on mahdollisesti kytketty joko MRP- (Manufacturing Resource Planning) tai PDM-järjestelmään (Product Data Management), eli järjestelmiin, jotka hallitsevat tuotannonohjausta tai tuotetietoja. Tuotteen kuluttaja ei juuri koskaan näe suunnitteluautomaattia käytössä. Markkinointisovelluksella tarkoitetaan tässä tuotteen kuluttajalle suunnattua sovellusta, jonka käyttäjä voi olla joko ammattisuunnittelija tai kuluttaja. Markkinointisovellus sisältää nimensä mukaan tuotteen markkinoinnissa tarvittavan tuotetiedon, mutta ei ole laajasti kytketty tuotannon prosesseihin tietolähteeksi.

3 MARKKINOINTISOVELLUS

3.1 Markkinoinnin tarkoitus

Markkinoinnin tehtävä on asiakkaiden hankinta ja tuotteen myynti eli yrityksen kannattavuuden kasvattaminen (Pihkala-Bäckström 2003). Toisin sanoin markkinoinnin tarkoitus on tuottaa voittoa yritykselle luomalla arvoa asiakkaalle tuotteesta (Markkinointi 2007). Täten markkinointisovelluksen tarkoituksena voidaan pitää sen tuomaa lisäarvoa tuotteelle.

Markkinointisovelluksen tuoma lisäarvo modulaariselle tuotteelle on laaja. Siihen sisältyy sovelluksen luoma viesti, tuotteen rakenteen ymmärtäminen, virtuaalisen kokonpanomallin luominen ja sen jatkokäytön

mahdollisuudet tuotteen elinkaaren aikana ympärillä toimivien ryhmien hyväksi.

3.2 Markkinointisovelluksen hyödyt

Markkinointisovelluksen voidaan lukea kuuluvan palveluna digitaaliseen markkinointiin, jonka kanavat ovat Internet, sähköposti, mobiililaitteet ja digi-TV. Etenkin kuluttajille suunnatut sovellukset käyttävät webverkkoa kanavanaan. Vaikkakaan modulaarisen tuotteen kokoonpanosovelluksen kanavana ei käytettäisi Internettiä, voidaan ajatella sille kuuluvan digitaalisen markkinoinnin ominaisuuksista niitä, jotka eivät kytkeydy verkon tuomiin etuihin.

Yahoon ohjelmistojohtaja Tim Sanders toteaa haastattelussa, että markkinoinnissa haetaan koko ajan enemmän elämyksiä. Internet onkin Sandersin mukaan hyvä keino toteuttaa elämyksiä, elämyksillä saadaan viestintätulvassa huomioita. (Aavameri 2002.)

On olemassa useita viihdepelejä, joissa toteutuu modulaarisuuden piirteitä. Takavuosien Tetris lienee niistä tunnetuimpia. Modulaarisen tuotteen sovelluksessa, varsinkaan kuluttajille suunnatuissa, ei pidä väheksyä viihteellisyyden tuomaa lisäarvoa, mutta tärkeimpänä ominaisuutena täytyy Business-to-Business -markkinoinnissa pitää teknistä hyötyä visuaalisessa kokonaisuudessa.

Rope (1998, 157–160) toteaa jo vuonna 1998, että Business-to-Business -markkinoilla on toimittava verkossa. Yritysmarkkinoiden avainhenkilöiden normaaliin työskentelytapaan kuuluvat internet ja tietokoneet. Verkkoviestintä, joilla Rope tarkoittaa www-sivuja ja niihin tehtyjä sovelluksia, on osa yritysten välttämätöntä toimintasisältöä. Edelleen paikkaansa pitävä Ropen väite on, että web-palvelut tulee suunnitella juuri tämän median ominaispiirteiden ja käyttömahdollisuuksien täyden hyödyntämisen pohjalta.

Tämänhetkisen Internet-teknologian hyödyntämiseen kuuluu Ropenkin (1998, 160) mainitsema toimintaan aktivointi, jotta sivuille pysähdyttäisiin sekä verkkoviestinnän, www-sivujen tai kehittyneempien websovellusten suunnittelu toimimaan mahdollisimman laaja-alaisesti kaikissa mahdollisissa sovellusten käyttöfunktioissa.

Markkinointisovelluksissa ja etenkin tuotevarianttien luomisessa ajansäästö on erityisen suuri hyöty markkinoijalle. Merisavonkin mukaan itsepalvelun kasvattaminen säästää asiakaspalvelukustannuksia. Digitaalisen markkinoinnissa myös kohderyhmän tavoitettavuus ja akti-

vointi sekä interaktiivisuuden lisääminen ja personointi ovat Merisavon sanoin keskeisiä etuja markkinoijalle (2006). Interaktiivisuuden katsotaan luovan brandiuskollisuutta ja sitouttavan asiakkaan tuotteeseen (Merisavo 2006). Modulaarisen tuotteen konfiguraattorissa sekä interaktiivisuus että personointi kytkeytyvät vahvasti yhteen ja voidaan jopa ajatella, että asiakas itse personoi interaktiivisella toiminnallaan tuotteen itselleen.

Asiakkaan näkökulmasta digitaalisen markkinoinnin hyötyjä ovat toistojen määrän rajoittamattomuus ja ajasta riippumattomuus. Keskeisiin hyötyihin asiakkaalle lukeutuvat myös nopeampi ja parempi palvelu sekä ostamisen nopeus ja helppous, kun tuotteeseen ja sen ominaisuuksiin on tutustuttu etukäteen ennen ostotapahtumaa, tai kun ostotapahtuma sijoittuu verkkoon. Tutustuminen itsenäisesti tuotteeseen madaltaa ostamisen kynnystä. Merisavo rohkaiseekin kokeilemaan uusia markkinoinnin muotoja, jotta pysyisimme kehityksen mukana. (Merisavo 2006).

Iteroinnin, toistojen määrän, rajoittamattomuus modulaarisen tuotteen markkinointisovelluksessa auttaa myös nopeampaan ja tarkempaan erilaisten vaihtoehtojen kustannusvertailuun, ja tuotteen visualisointi toimii päätöksenteon tukena. Lisäksi sovellukseen voidaan tuoda viihde-elementtejä luomaan elämyksiä, voidaan simuloida käyttöä ja kokeilla tuotetta sekä nähdä se kokonaisuudessaan ennen hankintapäätöksen tekoa (Vuori ym. 2000). Sovelluksen ominaisuuksiin voidaan liittää myös fyysisten toimintojen animaatioita ja hallintalaitteiden käytösesityksiä (Vuori ym. 2000).

3.3 Sovelluksen käyttöympäristö

Mitä tahansa sovellusta luotaessa täytyy jo aikaisessa vaiheessa ottaa huomioon sen tuleva käyttöympäristö. Tämä ohjaa tuottamista ja rajaa pois useita toteuttamisen vaihtoehtoja ja välineitä. Erityisen huomionarvoista on sovelluksen käyttäminen verkossa. Vaikka kaistanleveyden kasvu ja selainten kehittyminen on mahdollistanut uusia toimintoja verkossa käytettäväksi, täytyy sovelluksen ja mallien olla käytöltään kevyitä ja kohderyhmän tavoitettavissa. Verkko tuo myös etuja: päivitettävyys on hallinnassa, kaikilla käyttäjillä on aina uusin versio käytössä. Verkon yli toimivia sovelluksia voidaan toki käyttää myös ilman verkkoa.

Kun markkinointisovellusta käytetään kiinteässä työympäristössä, mahdollistaa se laajempien kokonaisuuksien esittelyä, mallien tarkem-

paa ja realistisempaa kuvausta sekä useampien toimintojen kytkemistä markkinointisovelluksen yhteyteen eli kaiken kaikkiaan monipuolisempia ja samalla raskaampikäyttöisiä sovelluksia.

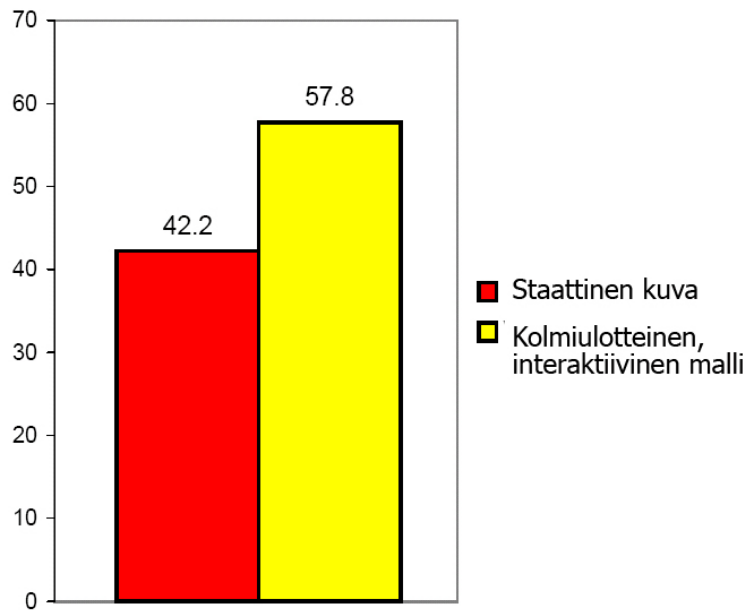
Käyttöympäristössä myös tarvittavat resurssit on otettava huomioon. Resursseihin kuuluvat ohjelmat, käyttöjärjestelmä, prosessorin teho, muistin määrä, kuten myös kaistanleveys verkossa toimiessa. Kohdeyhmä huomioon ottaen voidaan sovellus luoda toimimaan vähintään keskimääräisen kuluttajan resurssien mukaan. Erityistapauksissa, kun kohdeyhmä on suppea ja tarkasti rajattu, voidaan resursseilta vaatia enemmän. Esimerkki tällaisesta on ilman verkkoa sovellusta käyttävät CAD-ohjelmien käyttäjät, joilla on CAD-ohjelmien vaatimat uudehkot käyttöjärjestelmät, tehokkaat prosessorit ja tietyt ohjelmistot käytettävissä.

3.4 Tuotteen 3d-visualisointi

Tuotteen virtuaalisella 3d-mallilla parannetaan ymmärrystä tuotteesta. Se hahmotetaan paremmin ja sitä voidaan tutkia halutusta suunnasta. Rakennetta voidaan kuvata erilaisilla läpileikkauksilla tai kertoa tuotteen toiminnoista havainnollisin animaation verrattuna normaaliin, aitoon tuotteeseen. Erityisen suuri hyöty tavoitetaan, kun tuote tai sen osa on mitoiltaan äärimmäisen pieni tai suunnattoman suuri.

Web 3D Studion White paper -dokumentissa eli tuote-esittelyssä kuvaillaan digitaalisessa mediassa olevia tuotekuvia. Staattinen kuva nojautuu kuluttajan mielikuvitukseen, kun virtuaalinen malli ja sen realistinen ja interaktiivinen tutkiminen eri kuvakulmista lisää tuotteen läsnäolon tunnetta ja vaikuttaa näin ostopäätökseen. (How to Provide Engaging, Relevant and Accurate Product Information to Customers 2006.)

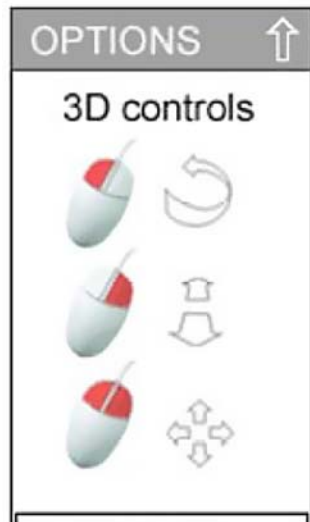
Kandassa, Albertan yliopistossa on tehty 2002 tutkimus, jossa vertailtiin staattisen tuotekuvan ja kolmiulotteisen, interaktiivisen mallin vaikutusta valintapäätökseen verkkokaupassa. Tämän tutkimuksen mukaan ostopäätöksen tehneet henkilöt, jotka ovat aiemmin ostaneet verkkokaupasta tuotteita, lähes 60 % valitsee virtuaalisella 3d-mallilla esitellyn tuotteen ostaakseen kuin täsmälleen saman tuotteen staattisella kuvalla varustettuna (kuva 2). Tuotetiedon määrä ja muu esitystapa olivat tutkimuksessa tuotteelle täysin samat. (Häubl, Figueroa 2002).



KUVA 2. Tuotekuvan vaikutus valintapäätökseen (Häubl, Figueroa 2002).

Erot edellisessä tutkimuksessa eivät ole huimia, mutta selkeitä silti. Tämä virtuaalisen mallin tuoman hyödyn lisäksi modulaarisen tuotteen sovelluksessa käyttäjä pääsee itse laatimaan kokoonpanonsa. Haittana on, että asiakas ei ole välttämättä kiinnostunut itse varioinnista, vaan tuotteen asiakaskohtaisuudesta (Modulointi 2006). Ongelmaa ei ole tuotteessa, jonka asiakas itse kokoaa heille tarjotuista moduuleista tai tuotteessa, jossa moduulit ovat uudelleen järjestettävissä ja käytettävissä kuten legot. Tällöin sovelluksesta on lisähyötyä tuotteen markkinoinnissa, jopa niin, että sovellus toimitettaisiin tuotteen mukana asiakkaan omaan käyttöön.

Joskus kolmiulotteinen visualisointi voi olla sekavaa, jo kappaleen kääntäminen haluttuun kulmaan voi olla pitkän harjoittelun tulos tottumattomille käyttäjille. Eri valmistajien kirjavien käyttöliittymien hallinnassa voi hiiren nappi pohjassa pidettynä pyörittää tuotetta, toisella valmistajalla näpäyttämällä samaa nappia aktivoida näkymän siirtämisen, pan-toiminnon. Tähän sekamelskaan tällä hetkellä pätevin selvityskeino on opastaa näkyvästi käyttäjää valitsemaan oikean toiminnon esimerkiksi kuvaohjeilla (kuva 3).



KUVA 3. 3d-mallin näkymän hallinta hiirellä (Production basics 2007).

3.5 Tekniikat

Modulaarisen tuotteen 3d-markkinointisovelluksen tuottamiseen tarvitaan tekniikoita, joilla voidaan esittää kolmiulotteisia objekteja ja kommunikoida vuorovaikutteisesti reaaliajassa niiden kanssa. On olemassa lukemattomia tekniikoita, joiden avulla voidaan tuottaa tällaisia sovelluksia: ohjelmointikieliä, ohjelmia, rajapintoja, 3d-mallien tiedostomuotoja, liitännäisiä ja 3d-moottoreita. Tekniikkojen valintaan voivat vaikuttaa kokoonpanosovelluksen käyttöympäristö, ohjelmointikielen ominaisuudet, saatavilla olevat ohjelmat tai sovelluksen tuottajan mieltymykset tai taidot. Jopa tuotteen modulaarisuuden laji vaikuttaa käytettävän tekniikan valintaan, moduulien vaihtokelpoisuuden toteuttaminen onnistuu kevyemmällä välineillä kuin rakennuslohkomodulaarisen tuotteen. Tässä luvussa käsitellään lyhyesti tekniikoita, jotka mahdollistavat sekä verkossa että yhteydettömänä todellisia kolmiulotteisia objekteja käyttävän sovelluksen tuottamisen. Luvussa 5 keskitytään tekniikkaan, jolla voidaan SolidWorks-ympäristössä tuottaa modulaarisen tuotteen 3d-markkinointisovellus.

Kiinnostavimmat verkossa toimivat avoimet 3d-tiedostomuodot ovat VRML ja kehitteillä oleva X3D. Molemmat kielet ovat tekniikoita kolmiulotteisten mallien ja maailmojen välittämiseen verkossa. VRML (Virtual Reality Modeling Language) on perinteinen tekstimuotoinen tiedostomuoto, jolla voidaan luoda interaktiivisia, kolmiulotteisia kappaleita ja lisätä niihin muun muassa värejä, kiiltoja, läpinäkyvyyksiä, pinnan teksturointeja, animaatioita, ääniä ja valoja (VRML 2007). VRML:n ongelmana on pidetty sen kunnollisen kielispesifikaation puutetta. Kielen kehitys jäi VRML97-versioon ja se korvattaneen hitaasti X3D:llä.

X3D (Extensible 3D) on kehitteillä oleva ISO-ohjelmistostandardi kolmiulotteisen sisällön esittämiseen ja kommunikointiin reaaliaikaisesti. Se on kehitetty VRML97-versiosta, mutta päivitetty nykyaikaiseksi XML-muotoiseksi standardiksi. X3D sisältää sekä geometrian kuvaamisen että ajonaikaisen käyttäytymisen yhdessä tiedostossa.

X3D tukee useita ominaisuuksia, kuten

- 2d- ja 3d-grafiikkaa ja ohjelmoitavia shadereita
- animaatioita ja muodonmuutoksia
- spatiaalisia video- ja äänitiedostoja
- käyttäjän vuorovaikutusta hiirellä ja näppäimistöllä
- navigoinnissa kameran käyttöä, törmäystarkasteluja, kappaleiden läheisyyttä ja käyttäjän liikkumista 3d-ympäristössä
- fysiikkasimulointia ja CAD-tiedostojen kääntämistä (What is X3D? 2007).

X3D:n arkkitehtuuri koostuu useista solmuista (node) kootuista komponenteista, ja komponenttiryhmit muodostavat eritasoisia profiileja (What is X3D? 2007). Solmu voi olla esimerkiksi laatikon määrittelemisen

```
Box : X3DGeometryNode {
  SFNode   [in,out] metadata NULL [X3DMetadataObject]
  SFVec3f  []      size      2 2 2 (0,•)
  SFBool   []      solid     TRUE
}
```

(Box – Web3D.org 2007). Solmu voi olla myös monimutkaisempi ryhmäsolmu, joka määrittelee vaikkapa törmäystarkastelun ominaisuuksia lapsisolmuillaan (Collision - Web3D.org 2007). Solmuista koostuva komponentti voi olla esimerkiksi Lightning component, valaistuskomponentti, joka määrittelee valonlähteet ja niiden sijoittamisen sekä kuinka valo vaikuttaa kuvan renderöintiin (Lighting component 2006).

Profiileja on tällä hetkellä kuusi, joista laajin eli Full-profiili käsittää kaikki määritellyt solmut ja komponentit. Muita profiileja ovat Interactive, jolla tuotetaan peruslaatuista vuorovaikutusta 3D-ympäristössä ja Immersive, joka tukee täydellistä 3D-grafiikkaa ja vuorovaikutusta muun muassa äänen, törmäystarkastelun ja sumun kanssa (What is X3D? 2007). Kun luodaan X3D-ympäristö, sille valitaan profiili, joka sisältää ne ominaisuudet, jota tuossa ympäristössä tarvitaan. Profiilit käyttäytyvät sopimuksen lailla näyttölaitteessa (usein selain), jolloin näyttölaite huomauttaa, jos kyseistä profiilia ei tueta (What are profiles? – Web3D.org 2006).

Tällä hetkellä X3D:tä käyttäviä sovelluksia on kokeiltu erilaisissa teollisuuden harjoittelutehtävissä ja prosessien simuloimisessa, myynnin esittelymateriaaleina ja lääketieteen visuaalisissa analyyseissä (X3D Scenarios and Case Studies 2007). Aika näyttää, yleistyykö kolmiulotteinen sisältö X3D:n ansiosta verkkoympäristössä, ainakin sen tuomat mahdollisuudet modulaarisen markkinointisovelluksien tuottamiseen ovat mainiot, ainakin tällä hetkellä tuettujen komponenttien perusteella on mahdollista tuottaa modulaarisuuden lajeista vaihtokelpoisia moduuleja, moduulien käyttöä useammassa eri kohteessa sekä parametristä modulaarisuutta. Koska X3D tukee myös raahausta ja törmäystarkasteluja, ei rakennuslohkomodulaarisuuden tuottaminen liene tällä tekniikalla kaukana tulevaisuudessa. X3D-tekniikalla on jo tuotettu toimiva markkinointisovellus timanttisormuksen timantin muokkaamisesta (kuva 4) (American Diamond 2007). XML:n yleistymisen verkossa tiedonvälittämisessä ja X3D:n arkkitehtuuritasojen ja profiilien mahdollistama alustariippumattomuus voivat tuoda uusia 3d-sovelluksia jopa mobiililaitteiden käyttöön .



KUVA 4. Timanttisormuksen esittely ja muuntelu X3D-tekniikalla internet-verkossa (American Diamond 2007).

Toimiakseen selaimessa sekä VRML että X3D tarvitsevat toistaiseksi joko liitännäisen selaimen tai erillisen, näiden formaattien käyttöön tarkoitetun selaimen. Koska X3D on kehitetty VRML:stä, voidaan useimmiten VRML-tiedostoja katsella X3D-liitännäisillä ja selaimilla (X3D FAQ 2007). Toimivia liitännäisiä X3D:n esittämiseen ovat Flux Player, vapaan lähdekoodin ActiveX-liitännäinen windowsille; Octaga Player, joka toimii sekä selaimen liitännäisenä että itsenäisenä selaimena ja Xj3D, Java-pohjainen avoimen lähdekoodin liitännäinen (Web3D.org: X3D Viewers, Browsers and Plug-ins 2007).

X3D:n tuottamiseen on olemassa graafisia käyttöliittymillä varustettuja ohjelmia. Tällaisia ovat esimerkiksi Flux Studio™ 2.0 ja WireFusion 4.1. Flux Studio tukee täysin X3D:n kaikkia ominaisuuksia, sillä voidaan luoda kolmiulotteisia objekteja, ja se on vapaasti ladattavissa eikaupalliseen käyttöön Media Machinesin verkkosivuilta (Media Machines – Make Web 3D virtual worlds and things – Flux Studio 2007). WireFusion on Demicronin kaupallinen tuote, jolla tuotetaan rikkasta verkkosisältöä. Ohjelmaan tuodaan valmiita X3D tai VRML-objekteja, joille tuotetaan vuorovaikutteisia toimintoja. Tällä ohjelmalla tuotettujen verkkosovellusten katseluun ei tarvitse 3d-liitännäistä selaimen, pelkkä Java-liitännäinen riittää. WireFusionista on ladattavissa 30 päivän kokeiluversio verkosta, Standard-version ohjelmaliisenssi Windows-käyttöjärjestelmälle maksaa tätä opinnäytetyötä tehdessä 199€ (Demicron 2007b).

Java on Sun Microsystemsin kehittämä laaja ohjelmistoalusta. Javaan kuuluvat muun muassa Java-ohjelmointikieli ja ajoaikainen ympäristö virtuaalikoneineen ja luokkakirjastoineen (Java 2007). Java-ohjelmointikieli on alustariippumaton olio-ohjelmointikieli. Alustariippumattomuus syntyy, kun ohjelmakoodi käännetään ensin tavukoodiksi (bytecode) ja tavukoodi-versiota ajetaan alustalle sopivalla Java-virtuaalikoneella eli tulkilla. Tulkkaus tapahtuu siis joka kerta, kun ohjelma ajetaan (Wikla 2005). Tavallisten ohjelmien, sovellusten, lisäksi kielellä voi ohjelmoida sovelmia (applet), jotka toimivat selaimessa HTML-sivun elementtinä. Sovelmia käytetään selaimissa ja myös pdf-dokumenteissa toteuttamaan ohjelmien kaltaisia vuorovaikutteisia tapahtumia, kuten kolmiulotteisuutta tai kuvankäsittelyominaisuuksia (Saksa, 2001).

Javalle on olemassa erityinen laajennus, Java3D, kolmiulotteista grafiikkaa sisältäviin ohjelmiin. Java3D:llä voidaan tuottaa visualisointeja, animaatioita ja vuorovaikutteisia sovelluksia ja sovelmia. Vaikka Java3D onkin kilpailija VRML:lle ja X3D, se myös toimii yhdessä niiden kanssa. (Huhta 2005.)

Kuten muukin kolmiulotteinen materiaali, vaatii Java-sovelma selaimen liitännäisen. Sovelman ajamisen voi myös hoitaa joissain selaimissa sisäänrakennettu virtuaalikone (Saksa, 2001). Tyypillisin Java 3d -mallien näyttämiseen tarvittava ympäristö on Sun Microsystemsin Java Runtime Environment, JRE, joka sisältää Java-sovelman katseluun tarvittavan Java virtuaalikoneen (Java SE Technologies at a Glance 2007).

Java-ohjelmointikielen laajojen ominaisuuksien vuoksi se soveltuu 3d-markkinointisovelluksen tuottamiseen sekä verkkoon sovelmana että kiinteäksi sovellukseksi tietokoneelle. Koska Java-ohjelmointiympäristössä voidaan ohjelmoida vaativia pelejä, voidaan tällä perusteella olettaa, että sillä myös kyetään toteuttamaan monimutkaisiakin rakennuslohkomodulaarisia sovelluksia. Java-ohjelmoinnin suurin haitta kolmiulotteisten markkinointisovellusten tuottamisessa on sen vaatima syvä perehtyneisyys Java-kielillä ohjelmointiin, myös sovelmien ja valmiiden ohjelmien muokkaus vaati kielen hallitsemista (Saksa, 2001).

4 MODULAARISTEN TUOTTEIDEN MARKKINOINTISOVELLUKSIA

4.1 Johdanto

Teknisiä tuotekonfiguraattoreita ei ole vapaasti saatavilla tuotesalaisuuksista johtuen. Tästä syystä tässä opinnäytetyössä on tutkittu kuluttajille suunnattujen modulaaristen markkinointisovellusten visuaalista rakennetta. Visuaalinen rakenne näissä on oletettavasti samankaltainen kuin ammattikäyttöön tarkoitetuissa sovelluksissa.

4.2 Vapaasti saatavilla olevia sovelluksia

4.21 IKEA Home Planner Kitchen

IKEAn visio on tarjota laaja valikoima muotoiltuja toimivia kodinsisustustuotteita niin edullisesti, että mahdollisimman monien olisi vara ostaa niitä. Itsepalvelu säästää rahaa. IKEA aktivoi asiakkaan itse hakemaan tuotteen varastosta, kuljettamaan sen kotiin ja kokoamaan tuotteen. (IKEA 2007b.)

Tähän markkinoinnin viestiin on liitetty vapaasti Ikean verkkosivuilta ladattava keittiösuunnitteluohjelma, Ikea HomePlanner 2007:1. Sovel-

luksen ovat tuottaneet kansainväliset yritykset Infor ja Geac. IKEAn verkkoratkaisuista vastaavan Martin Goldbergin mukaan tavoitteena oli tuottaa helposti ladattava ja käyttäjäystävällinen sovellus. (IKEA 2006.)

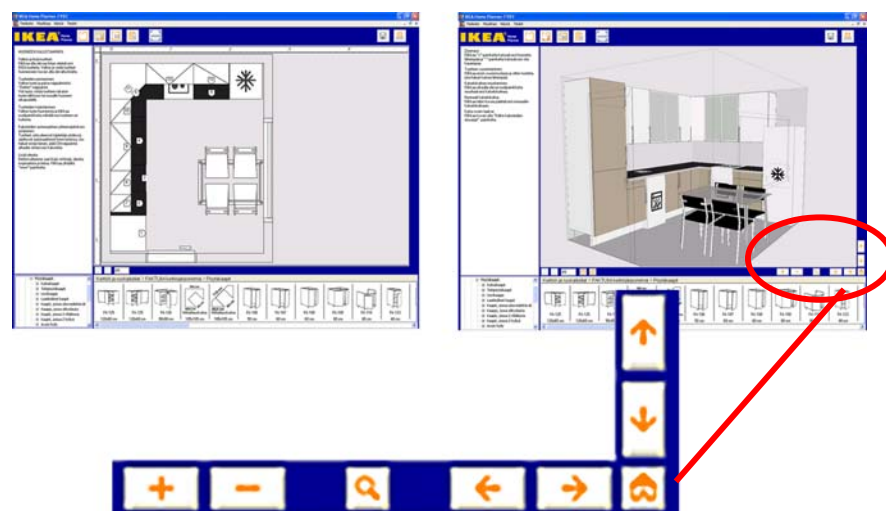
Järjestelmävaatimuksina sovellukselle IKEAn verkkosivuilla luetellaan seuraavat ominaisuudet:

- Pc
- Windows 2000, XP tai NT
- Pentium-prosessori
- 128 Mt muistia
- 25 Mt vapaata kovalevytilaa
- Näytönohjain 8 Mt:n muistilla
- Resoluutio 800x600 (1024x768 suositeltava), 16-bittiset värit.

Ladattavan sovellustiedoston koko on 6,36 Mt ja se käyttää Microsoftin DirectX -rajapintaa grafiikan piirtämisessä. (IKEA 2007a.)

Keittiökaapit ovat moduulityypeiltään väylämodulaarisia, eli niitä voidaan vapaassa järjestyksessä liittää rajapinnastaan takaseinästä perusyksikkönä olevaan huoneen seinäpintaan. Kaapeissa on myös rakennuslohkomaisia piirteitä, niiden sivut voivat olla rajapintoina ja kiinnittyä toisiinsa.

Kaikki moduulit ja niiden ominaisuudet eli kaapit, niiden materiaalit ja värit sekä mitat ja sijoitukset ovat muokattavissa koko suunnittelun ajan. Näkyminä ovat pohjapiirros ja perspektiivikuva (kuva 5). Perspektiivikuvaa hallitaan napeilla (kuva 5) ja sitä voidaan käännellä keskellä huonetta olevan akselinsa ympäri.



KUVA 5. Pohjapiirros ja perspektiivikuva sekä näkymän hallintanapit.

Keittiösuunnitteluohjelmassa käyttäjä valitsee huoneensa pohjamuodon, mitoittaa sen ja asettaa paikalleen ovet, ikkunat, väliseinät ja patterit sekä huonekalut ja kaapistot. Paikalleen asettaminen tapahtuu raahaamalla ne tuotelistasta, jossa tuotteet ovat viivapiirroksin, tuotetunnus ja tärkeimpien mittojen mukaan esiteltyinä ja tiputtamalla näkymään. Näkymässä moduulin rajapintakiinnitystä esitetään raahauksen yhteydessä pienellä nuolella (kuva 6), moduulin alakulma kiinnittyy automaattisesti nuolen osoittamalle paikalle, toinen rajapinta vasten perusyksikköä (kaapin takaseinä vasten huoneen seinää) ja toinen rajapintakiinnitys viereiseen moduuliin. Aktiivisen tartunnan voi halutessaan estää, jolloin moduuli sijoittuu täysin ilman rajapintaliitettä. Raahatessa kaappi liikkuu perspektiiviivapiirroksena.



KUVA 6. Aktiivisen tartunnan esittäminen nuolella raahauksen yhteydessä

Aktiiviseksi näkymästä valitun moduulin tuoteinformaatio on luettavissa näkymän vasemmalla puolella, jossa esitellään viivaperspektiivikuva moduulikaapista, mitat ja tuotetunnus ja jossa valitaan alasvetovalikoista moduulille halutut ominaisuudet (kuva 7). Sovelluksesta voidaan tulostaa näkymän kuva, pohjapiirros tai perspektiivikuva ja tuotelistaus, jossa näkyvät edellä mainittujen tuotetietojen lisäksi myös hinnat, kapalemäärät ja kokonaishinta. Keittiösuunnitelma voidaan tallentaa sovelluksessa IKEAn omalle palvelimelle, josta suunnitelma voidaan hakea paikan päällä missä tahansa IKEAssa. Suunnitelman tallennus onnistuu myös omalle tietokoneelle.



KUVA 7. Aktiivisen moduulin tuoteinformaatio.

IKEAn keittiösuunnitteluohjelma on helppokäyttöinen ja kevyt lähes jokaisen ikeakuluttajan käsissä. Sopiva lisä sovellukseen olisi suunnitelman renderöintimahdollisuus hieman todenmukaisemmaksi, tosin sovelluksen koko kasvaisi tuosta ominaisuudesta

4.22 Lego Digital Designer

Legon tuotekehittelijät jo vuonna 1999 miettivät keinoja, joilla Legon käyttäjät iästä riippumatta voisivat koota Legoista, mitä ikinä voisivat kuvitella ja edelleen suunnitella itse omat legokokoelmansa. Ratkaisuna tähän 2003 ilmestyi ilmainen virtuaalinen legorakennusohjelma, josta tätä opinnäytettä tehtäessä Legon verkkosivuilta (<http://idd.lego.com/>) on ladattavana Lego Digital Designer versio 2.0.32. (Lego Digital Designer 2007.)

Ladattava sovellustiedosto on 15 Mt, ja ohjelman suositusvaatimukset PC:lle ovat

- Windows XP SP 2
- prosessori 2 GHz
- 512 Mt muistia
- 100 Mt vapaata kovalevytilaa
- Näytönohjain 256 Mt:n muistilla.

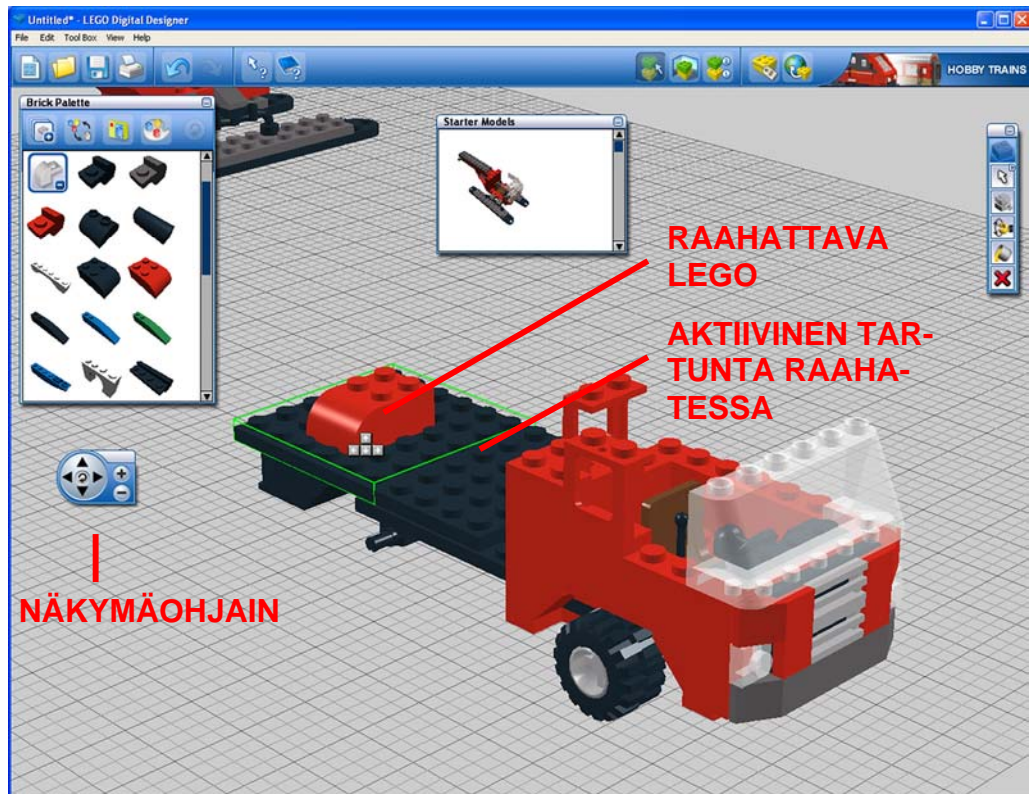
LDD käyttää grafiikan piirtämiseen OpenGL-rajapintaa. (the LEGO Group 2007a.)

Legot ovat tyypillisiä rakennuslohkomoduuleja, joita voidaan vapaassa järjestyksessä latoa rajapintojen, tyypillisen legopalikan ylä- ja alapintojen, kiinnityksillä. Lego Digital Designerissä (myöhemmin LDD) virtuaalisilla legopalikoilla rakentamisen lisäksi voidaan julkaista rakennelmat verkkoon muiden LDD:n käyttäjien saataville Legon sivuille online 3D-galleriaan, näyttää selainikkunassa rakennusohjeet tai tulostaa ne. Koska ohjelman palikat perustuvat aitoihin ja olemassa oleviin LEGO-pakkauspaketteihin, rakennelman hinta voidaan tarkastaa kohdemaan hinnoilla (kuva 8) ja rakennelmaan tarvittavat osat voidaan ohjelman avulla tilata verkkokaupasta. (Lego Digital Designer 2007).



KUVA 8. Rakennelmaan käytettyjen legopalikoiden hinta valitussa maassa

Kokooanotekniikka LDD:ssä on hiirellä raahaaminen. Uusia palikoita voidaan raahata kelluvasta valikkolaatikosta ja käytössä olevia voidaan siirrellä paikasta toiseen. Tartuntaa muihin palikoihin raahauksen aikana kuvataan värillisellä rautalankalaatikolla (kuva 9). Näkymää voidaan pyörittää vapaasti sekä hiiren omien nappien avulla tai näkymäohjaimen avulla (kuva 9). Legopalikoiden oman akselinsa ympäripyörittäminen on LDD:ssä toteutettu niin, että hiirellä raahaamisen aikana palikkaa voi pyöritellä nuolinäppäimin. Legokokoonpanon rakennusohjeet voidaan näyttää animaationa näkymässä tai tulostaa selaimen osaluettelona ja numeroituina rakennusohjeina.



KUVA 9. Näkymäohjain, kelluvat valikot ja raahauksen aikana tartunnan kuvaaminen toiseen legopalikkaan rautalankalaatikolla

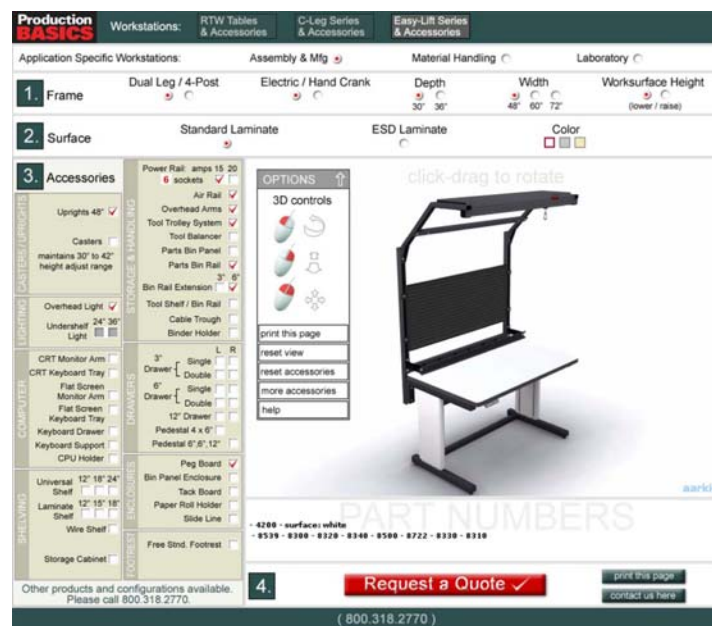
Järjestelmävaatimukset LDD:n pyörittämiseen ovat aika korkeat. Tämä varmasti vähentää muuten kiinnostavan sovelluksen käyttöä. Myös raahattavan legopalikan paikan hallinta on tuskastuttavan hankalaa ja vaatii harjaantumista. Muutoin sovelluksessa on käytetty huolitellusti verkon tuomia etuja: Legorakennelman julkistaminen verkossa muiden käyttäjien katsottavaksi on tehty helpoksi, nykyinen trendihän on saada verkon käyttäjät luomaan itse sisältöä internetmaailmaan. Helpoksi on tehty myös rakennelmaan vaadittujen palikoiden tilaaminen Legon omasta verkkokaupasta.

4.23 Rakenna oma ergonominen työasemasi

Production Basics on teollisuuden ergonomisia työasemia valmistava yritys. Se tarjoaa verkkosivuillaan sovellusta, jolla käyttäjä rakentaa haluamansa työaseman. Sovellus toimii verkon yli, ja sen on tuottanut Aarkid, sähköiseen kauppaan erikoistunut 3D-palveluja tarjoava yritys (Production Basics 2007). Käyttöliittymä sovellukseen on toteutettu Wi-reFusion-ohjelmalla (esitelty lyhyesti luvussa 3.5), sovelluksen esittämiseen ei tarvita erillistä 3d-liitännäistä, vaan se toimii Java-sovelmana automaattisesti, vaikka malli on joko X3D- tai VRML-muotoinen. Luon-

nollisesti sovelman näkymiseen vaaditaan kuitenkin Java-liitäntäinen (Demicon 2007a).

Työaseman moduulit ovat tyypiltään vaihtokelpoisia moduuleja ja kokoonpano rakennetaan täysin lomakkeen avulla (kuva 10). Tästä sovelluksesta ei rajapintaliitäntän visuaalista kuvausta tarvita, vaan sovellus automaattisesti sijoittaa moduulit omille paikoilleen. Kolmiulotteista työaseman näkymää ohjataan pelkästään hiiren avulla (Kuva 10). Sovelluksen kolmiulotteiseen malliin on toteutettu lyhyitä animaatioita havainnollistamaan moduulien toimintaa ja sijoittumista. Nämä animaatiot aktivoituvat, kun moduuli valitaan lomakkeelta.



KUVA 10. Lomakkeella hallittava työaseman konfiguraattori ja näkymän hallinta hiirellä.

Työaseman rakentamisen jälkeen koko sivu voidaan tulostaa tai jatkaa osaluettelon tulostamiseen. Osaluettelossa näytetään tärkeimmät tuotetiedot ja työaseman yhteenlaskettu hinta, moduuleja voidaan poistaa ja muokata tässäkin vaiheessa lomakkeen tapaan (kuva 11). Osaluettelo voidaan lähettää suoraan sovelluksesta sähköisesti yritykseen yhteydenottopyyntönä.

Production Basics

+ CONTACT US SEARCH

COMPANY PRODUCTS APPLICATIONS HOW TO PURCHASE CUSTOMER CENTER

→ Quote Request

Item	Frame Color	Worksurface Color	Each	Quantity	Total	Remove
WIRE SHELF, 12"D x 48"W (Item #8400)			\$68.00	1	\$68.00	<input type="checkbox"/>
PEDESTAL DRAWER, 6-6-12" DRAWERS (Item #8570)	Black		\$339.00	1	\$339.00	<input type="checkbox"/>
C-LEG SERIES CASTER OPTION, 4", SET OF 4 (Item #8970)	Black		\$104.00	1	\$104.00	<input type="checkbox"/>
OVERHEAD LIGHT, 48"W (Item #8500)	Black		\$177.00	1	\$177.00	<input type="checkbox"/>
KEYBOARD DRAWER (WITH MOUSE TRAY) (Item #8681)	Black		\$110.00	2	\$220.00	<input type="checkbox"/>
TACK BOARD, 48"W (Item #8730)			\$105.00	1	\$105.00	<input type="checkbox"/>
WORKSTATION, ESD, 30"D x 48"W (Item #1100)	Black	White	\$572.00	1	\$572.00	<input type="checkbox"/>
Total - \$1,585.00						

Return to Products Update Finalize Quote

Company | Products | Applications | How to Purchase | Customer Center | Contact Us | Site Map Production Basics © 2007 All rights reserved.

KUVA 11. Työaseman osaluettelo tuotetietoineen.

Työaseman rakentaminen konfiguraattorilla on sujuvaa ja helppoa hallita. Ainakin pienyritysten työasemahankintoja sovellus voi olennaisesti helpottaa, mutta toimiiko sovellus myös suurten yritysten hankintakanavana?

4.3 Moduulien visuaalinen sijoittuminen

Ensimmäisessä luvussa esitellyistä modulaarisuuden tyypistä riippuen toteutetaan sovelluksen rakenteessa rajapintaliitokset. Monimutkaisinta rakennetta vaatii legomoduloinniksikin kutsuttu rakennuslohkomodulaarisuus, sovelluksessa on otettava huomioon rajapinnoista riippuen lähes rajaton määrä moduuliyhdistelmiä. Kevyin rakenne sovellukseen syntyy tuotteesta, jossa modulaarisuus ilmentyy ainoastaan moduulien vaihtokelpoisuutena.

Tutkittujen vapaasti saatavilla olevien kokoonpanosovellusten perusteella moduulien visuaalinen sijoittamistapa on jaettu kolmeen eri tyyppiin:

- **Automaattinen** eli sovellus automaattisesti sijoittaa käyttäjän halumat moduulit paikalleen. Sovellus voi laskea paikan annettujen ehtojen mukaan tai sovelluksessa tietyille moduleille on olemassa vain tietty paikka.

- **Käyttäjän määrittelemä**, jolloin käyttäjä valitsee moduulin ja määrittelee täysin paikan moduulille.
- **Ehdollinen**, jossa sovellus osoittaa moduulin paikan ja käyttäjä valitsee moduulin sovelluksen tarjoamista vaihtoehdoista tai käyttäjä valitsee moduulin ja sovelluksen osoittamista vaihtoehtoista valitsee sille paikan.

Moduulien sijoittumisen tyyppi valitaan sovellukseen modulaarisuuden lajin, tuotteen ominaisuuksien ja markkinointisovelluksen loppukäyttäjän suunnittelutaidon perusteella. Kun moduulissa on vain yksi rajapinta, oli loppukäyttäjä ammattisuunnittelija tai ei, on perusteltua valita moduulien automaattinen sijoittamistyyppi, kuten edellä tutkitussa työaseman konfiguraattorissa. Tässä vaihtoehdossa inhimillisen erehdyksen ja väärinymmärryksen tuottamat virheet minimoidaan, kun oletetaan sovelluksen ottavan huomioon kaikki tarvittavat ehdot.

Kun rajapintoja on moduuleissa useampia ja oletetaan loppukäyttäjällä olevan lopullisen tuotteen suunnittelutaitoa, voidaan moduulien sijoittamistapa valita käyttäjän määrittelemän (edellä tutkitut Legorakennus- ja keittiösuunnitteluohjelma) ja ehdollisen tyyppin väliltä. Tällöin sijoittamistavan näiden väliltä voi ratkaista käyttöympäristön tai valitun tuottamistekniikan vaatimukset.

5 SOLIDWORKS SOVELLUKSEN TUOTTAMISESSA

5.1 Yleisesittely

SolidWorks on mekaniikkasuunnitteluun tarkoitettu 3d-CAD-suunnitteluohjelmisto, jonka käyttäjäkuntana ovat tekniset suunnittelijat ja insinöörit ympäri maailmaa. SolidWorks-ohjelmistoon on mahdutettu erilaisia tuotesuunnittelun apuvälineitä muottisuunnittelusta lujuusanalyysiin ja komponenttikirjastoihin sekä piirustusten luomiseen ja mallien tulostamiseen, toimintojen animointiin ja simulointiin, tiedonhallintaan ja mallien internetjulkaisuun. SolidWorks-ohjelmaa voidaan myös vapaasti räätälöidä nopeuttamaan rutiineja tai tuottamaan helpompaa hallittavuutta ohjelmistolle makrojen ja API-rajapinnan avulla.

SolidWorksin käytön sujuvuudeksi vaaditaan työasemakoneelta kohtalaista tehokkuutta. Jälleenmyyjän vähimmäissuositus on

- PC
- Windows 2000
- 2 GHz prosessori
- 2 GB keskusmuistia

- näyttöohjaimiksi NVIDIA Quadro tai ATI FireGL -kiihdytinpiireillä varustettuja ohjaimia. (CadWorks Oy 2007.)

SolidWorks käyttää näkymässään kolmiulotteisen osan reaaliaikaiseen renderöintiin automaattisesti näyttöohjainta tai valinnaisesti OpenGL-ohjelmointirajapintaa (SolidWorks 2006 Online User's Guide - SP4.0 2006a).

5.2 Soveltuvuus markkinointisovellukseen

SolidWorksin soveltuu markkinointisovelluksen pohjaksi vain siinä tapauksessa, että ohjelmistolisenssi on olemassa. Jos ohjelmalisenssiä ei muusta syystä ole hankittu, on investointi kallis (peruspaketti n. 7000€) ja turha pelkän markkinointisovelluksen pyörittämiseen. Toinen merkittävä piirre SolidWorksin toimiessa markkinointisovelluksen pohjana on sen vaatimus ohjelmiston pyörimisestä aktiivisesti sovelluksen osana, sovellus ei siis toimi internetverkossa. Sovelluksessa kootun mallin voi kuitenkin julkaista verkossa.

Vaikka SolidWorksin käyttämiseen markkinointisovelluksen osana on rajoitteita, ovat sovelluksen hyödyt suuret, etenkin teknisessä teollisuuden tuotteessa. SolidWorksin avulla on mahdollista kytkeä markkinointisovellus ja sen tuottama kokoonpano koko tuotteen elinkaareen, tuotannon hallinnoimiseen ja tuotetiedon hallintaan (PDM, Product Design Management). Koska kokoonpanosovellus käyttää hyväkseen SolidWorksin julkaisemaa avointa APIa, ohjelmointirajapintaa, ovat lähes kaikki SolidWorks-ohjelman toiminnot mahdollisia API-funktioiden välityksellä sovelluksen automaattisiksi toiminnoiksi. Erityisesti modulaarisessa tuotteessa ominaiset rajapintaliittymät ovat helppoja toteuttaa SolidWorksin omien liitäntäpiirteiden, Mate-suhteiden avulla, mikä muilla aiemmin esitellyillä tekniikoilla vaatisi usein täydellistä paneutumista ohjelmointikielien saloihin.

Markkinointisovelluksen ja SolidWorks-ohjelman yhdistämisellä viestitään vahvasti teknistä ammattitaitoa ja asiallisuutta, vaikka se voi toimia myyntitilanteessa myös viihdyttävänä kokemuksena siinä tapauksessa, että tuote visuaalisestikin rakentuu kokoonpanoksi. Tuotetiedon julkaiseminen sovelluksessa on mahdollista piirustuksien tuottamisesta vaikkapa materiaalivalintoihin tai osaluettelon tulostamiseen. Moduuleja voidaan päivittää yksinkertaisesti pelkällä SolidWorks-suunnittelutaidolla osatiedostoihin (.SLDPRT-pääte) tai alikokoonpanoihin (.SLDASM-pääte) normaalilla CAD-suunnittelulla ilman ohjelmointitaitoa, kunhan rajapintoihin ei kosketa. SolidWorks ja samalla

markkinointisovellus käyttävät kokoonpanoihin näitä erillisiä osatiedostoja ja alikokoonpanoja. Lisäksi sovellusta luotaessa, mikäli SolidWorks on kytketty aiemmin modulaarisen tuotteen suunnitteluun, 3d-moduulimallit ovat valmiina, eikä niille juurikaan tarvitse tehdä muutoksia ellei sitä erityisesti vaadita esimerkiksi tuotesalaisuuksien takia.

Tässä opinnäytetyössä tutkitaan tarkemmin, kuinka modulaarisen tuotteen kokoonpanosovellus voidaan toteuttaa VBA (Visual Basic for Application) -ohjelmointikielillä makrojen nauhoittamista apuna käyttäen. Makrojen nauhoitusta apuna käyttäen on helppoa perehtyä API-funktioiden toimintaan, ja kevyellä lisäohjelmoinnilla saavuttaa tarpeenmukaisia ja monipuolisia toimintoja kokoonpanosovellukseen. Kokoonpano voidaan koota sovelluksessa automaattisesti tai käyttäjän ohjaamana lomakkeiden avulla. SolidWorks-ohjelmaa voidaan ohjata myös muilla ohjelmointikielillä, kuten Visual Basic 6.0 tai VB .NET – kielillä

5.3 Visual Basic for Application (VBA)

VBA (Visual Basic for Applications) on Microsoftin sovellusohjelmissa ja makroissa käytetty tulkittava ohjelmointikieli (Heinonen 2005). Kuten nimestäkin voi päätellä, se kuuluu Visual Basic -kieliperheeseen, mutta VBA:n ohjelmakoodi voidaan normaalisti ajaa vain isäntäohjelman sisällä, ei itsenäisenä sovelluksena. Se voi kuitenkin ohjata toista ohjelmaa isäntäohjelman päältä OLE Automation -tekniikan avulla. VBA:n lisenssi sisältää VBA-kehitysympäristön (integrated development environment, IDE): Microsoftin lomakkeet, virheiden jäljittäjän eli debuggerin, koodieditorin, objektiselaimen ja kääntäjän. (Visual Basic for Application Frequently Asked Questions 2007.)

Visual Basicin syntaksi muodostuu yhden rivin ohjelmalauseista, joka voi olla

- esittelylause
- määrittelylause
- suoritettava ohjelmalause

tai näiden yhdistelmä. Esittelyohjelmalauseessa esitellään muuttuja, määrittelyohjelmalauseessa annetaan sille arvo ja suoritettavassa ohjelmalauseessa suoritetaan jokin tapahtuma. Tapahtuma voi olla metodi, funktio tai ohjausrakenne (esimerkiksi ehto tai luuppi). (Halvorson 2006, 117–326.)

Ohjelmalauseessa metodia kutsuttaessa sille määritellään ensin luokkaobjekti, joka voi olla lomake tai isäntäohjelman elementti, esimerkiksi

UserForm.Show. Myös määrittelylauseessa luokkaobjektin ominaisuuksia asetettaessa täytyy kutsua luokkaobjektia: UserForm.Caption = "Lomake". (Visual Basic Reference 2006d.)

Vaikka VBA ei vaadi Option Explicit -asetusta, eli asetusta siitä, että kaikki muuttujat on esiteltävä, sitä suositellaan käytettävän (Spens 2004, 11). Jos muuttujia ei esitellä, voi olla hankalaa löytää virheitä, ja tilanvaraus on suuri, kun ohjelma itse määrittelee esittelemättömien muuttujien tyypit variant-tyyppiseksi (Visual Basic Reference 2006b). Variant voi sisältää sekä luvun, totuusarvon tai merkkijonon. Variant-tyyppi varaa tilaa jopa 22 tavua, kun normaali lukutyyppi Integer varaa vain 2 tavua (Visual Basic Reference 2006c).

Yleistä ohjelmoinnille, myös VBA:lle, on, että jokin ohjelman osio suoritetaan useassa kohdassa ohjelmaa. Tällöin käytetään aliohjelmaa, joiden avulla ohjelmakoodi saadaan pidettyä lyhyempänä ja selkeämpänä sekä vähennetään virheiden mahdollisuutta. Aliohjelman ominaisuuksiin kuuluu, että se kirjoitetaan vain kerran, mutta sitä voidaan kutsua eri kohdissa ohjelmaa. (Aliohjelma 2006.) Aliohjelmien käyttöä puoltaa myös se, että kerran kirjoitettua aliohjelmaa voidaan käyttää aina uudestaan myös toisissa ohjelmissa (Laaksonen 2007).

Visual Basicissa ensisijaiset aliohjelmat ovat funktio- ja sub-aliohjelmat (Halvorson 2006, 245). Tapahtumien hallinnassa käytetään sub-aliohjelmaa, kuten Button_Click tai Form_Load. Visual Basicin piirre on, että kaikki ohjelmointilausekkeet sijoitetaan aliohjelmien sisälle. Poikkeus tästä on ainoastaan yleiset esittelyt ja kääntäjälle tarkoitetut ohjeet, jotka voidaan sijoittaa aliohjelman ulkopuolelle. (Laaksonen 2007.)

Muuttujien näkyvyys VBA:ssa määritellään niin, että lomakkeessa esiteltävät muuttujat ovat käytössä ja kelvollisia vain siihen lomakkeeseen, jossa ne ovat esitelty. Aliohjelmien ulkopuolelle sijoitetut yleiset muuttujaesittelyt näkyvät kaikissa lomakkeissa.

Moduulit Visual Basicissa ovat erikoistiedostoja, joiden päätte on .bas. Kun tarvitaan ohjelman kaikissa osissa käytettäviä muuttujia tai tapahtumankäsittelijöitä, ne ohjelmoidaan näihin moduuleihin ja määritellään Public-avainsanalla. (Halvorson 2006, 237.)

Visual Basic itsenäisenä ohjelmointikielenä on nykyään kehittynyt .NET arkkitehtuuria tukevaksi Visual Basic 2005:ksi. On siis mahdollista päivittää edelleen nauhoitetut makrokoodit kehittyneempään .NET ympäristöön ohjelmalla, joka tukee 2005-versiota ja .NET arkkitehtuuria. Eräs tällainen ohjelma on vapaasti Microsoftin sivuilta ladattava Vi-

sual Basic 2005 Express Edition. Päivitys vaatii perehtymistä VBA:n ja Visual Basic 2005 -kielien eroihin olemassa olevien päivitysapuohjelmien lisäksi.

Visual Basic -kielessä syntaksit luetaan yhdeltä riviltä. Kapean palstanleveyden takia käytetään esimerkeissä kielen yleistä merkintää " _" kuvaamaan syntaksin jatkumista samalla rivillä (Spens 2004, 3). Syntaksissa ' on merkinä kommentin alkamisesta eikä kääntäjä käsittele eikä suorita niitä. Kommentteja käytetään yleisesti ohjelmakoodin selvittämiseen ja dokumentointiin (Halvorson 2006, 85).

5.4 Makrot

Makro on kokoelma komentoja ja siis nimi menettelytavalle tai funktiolle ohjelmoinnissa. Makrot sovelluksissa ovat tietokoneohjelman apuohjelmia. Makroja käytetään yleisimmin automatisoimaan usein käytettyjä toimintoja Microsoft Office -ohjelmissa. (Krieger 2007.)

Vaikka makroilla saadaan aikaan hyödyllisiä toimintoja, liittyy niihin tietoturvariskejä. Makrovirukset ovat tunnettuja haittaohjelmia, jotka on ohjelmoitu makrosovelluskielillä ja niitä tavataan pääasiassa Office-tuotteiden yhteydessä. Makrovirukset leviävät asiakirjojen mukana ja ne ovat ongelmallisia asiakirjojen välittämisen yleisyyden sekä makrovirusten helpon luomisen ja muokkauksen vuoksi. Ajantasainen virus-turva ja terve kriittisyys makroja kohtaan ovat siis paikallaan. (F-Secure Anti-Virus 5 manual 2005.)

Makroa nauhoitettaessa SolidWorks-ohjelmassa kirjoittaa ohjelmakoodia kaikista toiminnoista, joita suoritetaan. Nauhoitetut makrot tallennetaan .swp-tiedostoina. Kun makro nauhoituksen jälkeen ajetaan, suoritetaan koodin ohjaamana toiminnot automaattisesti. SolidWorks-ohjelmassa makrot ajetaan normaalilla valikon Run-käskyllä, jonka jälkeen haluttu makro haetaan tiedostoista. Tiheään käytetyille makroille voidaan luoda räätälöidyt napit ohjelman työkalukuvakkeisiin. Näin makrosovellus on vaivattomasti käytettävissä jatkuvasti. (Spens 2006, 12–13.)

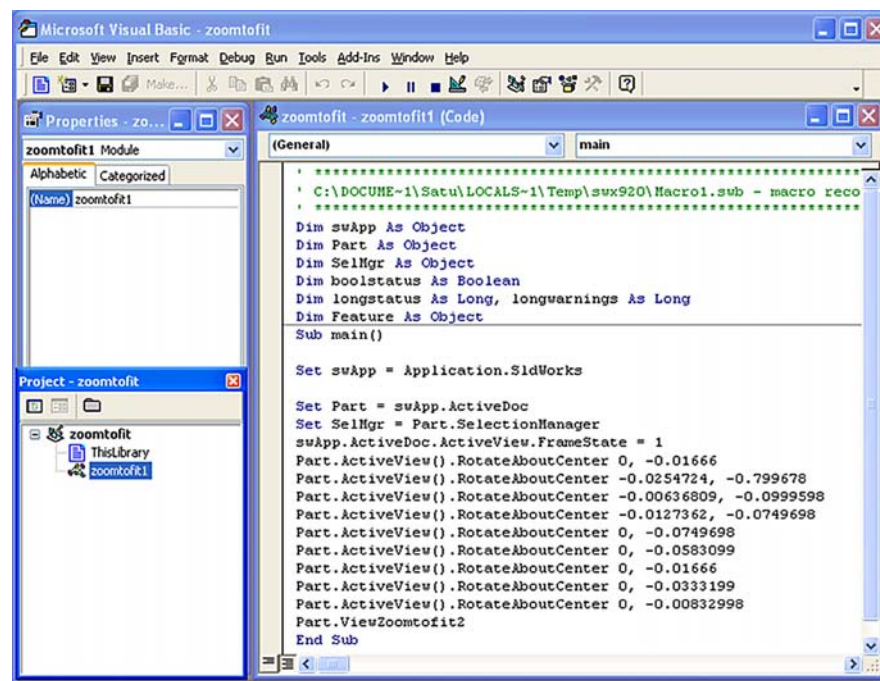
Makron nauhoittamaa koodia päästään muokkaamaan editorin avulla (Spens 2004, 2). SolidWorks 2006 -versiossa editorina on Microsoft Visual Basic 6.4 versio 1001, ja makron nauhoituskielenä on Visual Basic for Application (VBA) 6.5.1001, joka käyttää toiminnoissa API-funktioita.

Makrojen käyttö SolidWorks-ohjelmassa voidaan jaotella kolmeen tyyppiin:

- Itsenäisellä makrolla (Independent) mallin geometriaa ei muuteta ja makro on vain vuorovaikutteinen mallin kanssa.
- Muokkaava makro (Modify) muokkaa olemassa olevan mallin geometriaa.
- Luova makro (Create) luo uuden piirteen malliin. (Hasz 2006.)

Makrotyypeistä itsenäiset ja muokkaavat makrot soveltuvat kumpikin markkinointisovelluksen osaksi. Itsenäisen makrotyyppin koodi voi markkinointisovelluksessa olla esimerkiksi yksinkertainen Suppress-tai Hide-tilan muutos, muokkaava makro vastaavassa sovelluksessa käyttäjän syötteestä mallin mittoja muuttava makro. Luova makro sopii kehojen modulaariseen tuotteeseen, sillä muutokset modulaarisuuden määritysten mukaan voivat olla hyvin vähäisiä, eikä rajapintoihin voida lainkaan koskea.

Vaikkakin nauhoitettuja makroja voidaan käyttää sellaisenaan, on niiden optimointi tarpeellista. Nauhoittamalla makroja SolidWorks kirjoittaa ylimääräisiä koodeja ja esittelee muuttujia, joita ei välttämättä tarvita (Spens 2006, 20). SolidWorks-ohjelmassa mallin pyörittämisestä ja Zoom to Fit -toiminnosta nauhoitetusta makrosta nähdään, että yksinkertaisessakin toiminnassa koodia kasaantuu tarpeettoman pitkäksi (Kuva 12).



```
Microsoft Visual Basic - zoomtofit
File Edit View Insert Format Debug Run Tools Add-Ins Window Help
zoomtofit1 - zoomtofit1 (Code)
(Properties - zoomtofit1)
zoomtofit1 Module
Alphabetic Categorized
(Name) zoomtofit1
Project - zoomtofit
zoomtofit
ThisLibrary
zoomtofit1
(General)
main
C:\DOCUME-1\Satu\LOCALS-1\Temp\sux920\Macro1.sub - macro reco
Dim swApp As Object
Dim Part As Object
Dim SelMgr As Object
Dim boolstatus As Boolean
Dim longstatus As Long, longwarnings As Long
Dim Feature As Object
Sub main()
Set swApp = Application.SldWorks
Set Part = swApp.ActiveDoc
Set SelMgr = Part.SelectionManager
swApp.ActiveDoc.ActiveView.FrameState = 1
Part.ActiveView().RotateAboutCenter 0, -0.01666
Part.ActiveView().RotateAboutCenter -0.0254724, -0.799678
Part.ActiveView().RotateAboutCenter -0.00636809, -0.0999598
Part.ActiveView().RotateAboutCenter -0.0127362, -0.0749698
Part.ActiveView().RotateAboutCenter 0, -0.0749698
Part.ActiveView().RotateAboutCenter 0, -0.0583099
Part.ActiveView().RotateAboutCenter 0, -0.01666
Part.ActiveView().RotateAboutCenter 0, -0.0333199
Part.ActiveView().RotateAboutCenter 0, -0.00832998
Part.ViewZoomtofit2
End Sub
```

KUVA 12. Zoom To Fit -toiminto nauhoitettuna makroksi koodieditorissa

Optimoinnin, eli tarpeettomien muuttujien ja koodirivien poistamisen sekä rivien järkevien sisennyksien jälkeen, sama koodi huomattavasti helpompi lukea.

```
Option Explicit

Dim swApp           As SldWorks.SldWorks
Dim Part            As Object

Sub main()
    Set swApp = Application.SldWorks
    Set Part = swApp.ActiveDoc

    Part.ViewZoomtofit2
End Sub
```

5.5 API-funktiot

Avoimen ohjelmointirajapinnan (API) julkaisu on hyvin yleistä toimijoilla, kuten Microsoft, jotka hyötyvät välittömästi tai välillisesti rajapinnan avulla tuotetuista sovelluksista. Ne laajentavat ohjelmistojen käyttöä ja kasvattavat ohjelmistojen myyntiä, Microsoftin tapauksessa Windows-käyttöjärjestelmän myyntiä. (Ohjelmointirajapinta 2006.)

SolidWorks 2006:n sisältämä ohjelmointirajapinta, SolidWorks API, sisältää satoja funktioita, joita voidaan kutsua useammalla eri ohjelmointikielellä tuotetuissa tiedostoissa. Dokumentoituina ovat ohjelmointikielien VBA, VB (Visual Basic) 6.0, VB .NET, Visual C#. NET, Visual C++ 6.0 ja Visual C++ .NET. (SolidWorks and ADD-Ins API Help 2006a.)

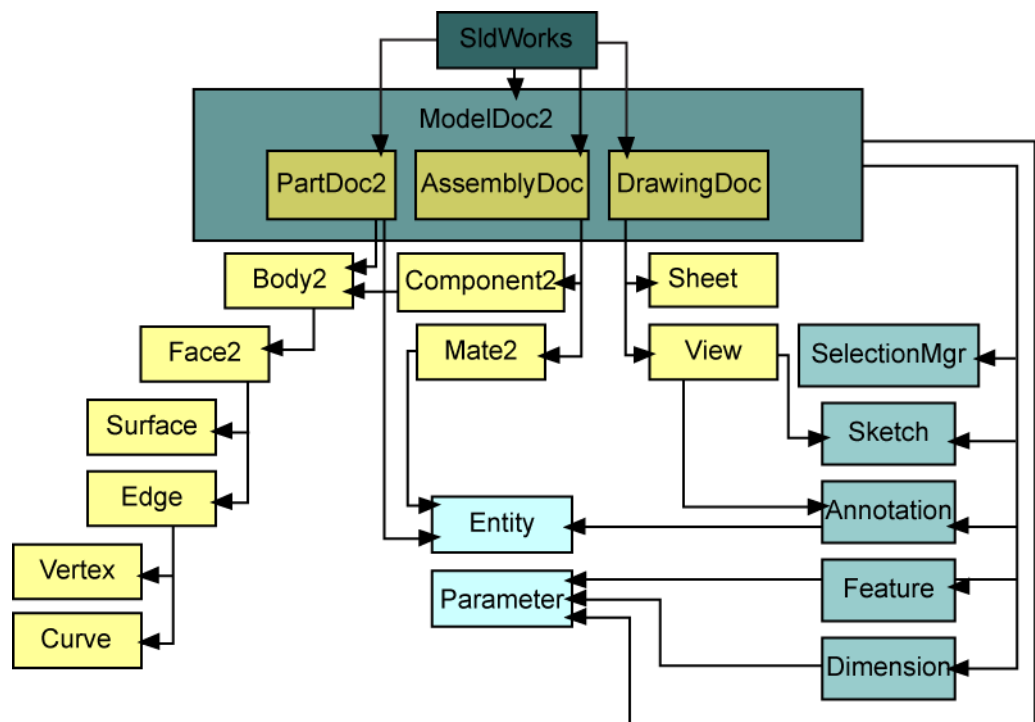
API funktioilla on suora pääsy SolidWorksin toimintoihin, esimerkiksi viivan piirtämiseen, osan lisäämiseen tai pintojen parametrien tarkistukseen. API-rajapintaa käytetään yleisesti toimintojen automatisointiin ja SolidWorks-ohjelman räätälöintiin, API-funktioilla ohjelmoimalla saadaan sovellus tekemään lähes kaikki ohjelmatoiminnot, jotka ovat mahdollisia SolidWorksin käyttöliittymässä.

API-funktioiden yleinen rakenne koostuu luokkaobjekteista ja niiden metodeista (methods) sekä ominaisuuksista (properties) ja mahdollisista tapahtumailmoituksista (events).

Useimmat API-luokkaobjektit ovat suoraan verrattavissa SolidWorksin käyttöliittymässä näkyviin objekteihin, kuten Feature (piirre), mutta osa API-funktioiden toiminnoista vaatii objektin, jona voi toimia usea käyttöliittymässä näkyvä objekti. (SolidWorks and Add-Ins API Help 2006e). Esimerkiksi Parameter-luokkaobjektin metodilla GetType voidaan tutkia sekä mitan (Dimension) että piirteen (Feature) tyyppiä.

Jokaisella SolidWorks-ohjelman dokumenttityypillä on oma luokkaobjektinsa ja näillä omat funktiot. SolidWorksin API-funktioissa on myös funktioita, jotka ovat yhteisiä kaikille dokumenttityypeille, kuten tulostamis- ja tallennusfunktiot. Näillä dokumenttitason funktioilla käytetään yhteistä luokkaobjektia ModelDoc2, jona voi toimia mikä tahansa dokumenttiobjektityyppi. (SolidWorks and Add-Ins API Help 2006g)

SolidWorks-ohjelman API-luokkaobjekteista kaikkein ylimmällä tasolla hierarkiassa on SldWorks (Kuva 13). Osaan alemmalla tasolla olevien luokkaobjektien metodeihin ja ominaisuuksiin on suora pääsy, mutta suurin osa vaatii ylempien tasojen luokkaobjektien määrittämistä (SolidWorks and Add-Ins API Help 2006e). Kuvasta 13 voidaan seurata luokkaobjektin määrittelyä, kun esimerkiksi halutaan tarkastella kokoonpanodokumentissa komponentin tietyn pinnan ominaisuuksia: SldWorks > AssemblyDoc > Component2 > Body2 > Face2 > Surface.



KUVA 13. Osa SolidWorksin API-luokkaobjekteista ja niiden hierarkiajärjestys.

Koska API-funktioita on satoja, ja niillä on pääsy lähes kaikkiin SolidWorks-ohjelman toimintoihin, käsitellään tässä opinnäytetyössä vain modulaarisen tuotteen kokoonpanosovellusta VBA:lla toteutettaessa keskeisimmät funktiot ja menettelytavat. Eri tavoin ohjelmoimalla on mahdollista tuottaa näitä markkinointisovelluksen ominaisuuksia, joten

tässä työssä esitellään pääasiassa nauhoitettuihin makroihin perustuvia toimintojen ratkaisuja.

5.6 Sovelluksen käyttöliittymä

Makrosovelluksien käyttöliittymät toteutetaan yleisesti Microsoftin lomakkeilla, kun halutaan käyttäjän pystyvän vaikuttamaan makron toimintaan. Lomakkeen elementit ovat tutun näköisiä Microsoftin ohjelmiin tottuneille, eikä niiden käyttöä tarvitse erikseen opetella.

Lomakkeissa on saatavilla VBA-sovellusten tuottamiseen yleisimmät Microsoftin lomakkeen elementit. Lomakkeesta pystytään ohjaamaan sovelluksen toimintaa esimerkiksi nappien, syöttökenttien ja listojen avulla, ryhmittelemään toimintoja useammille sivuille ja antamaan käyttäjälle ohjeita. Näitä elementtejä ohjataan VBA-koodilla API-funktioilla. Esimerkiksi päivitä-nappia painamalla makro päivittää SolidWorksin näkymän:

```
Private Sub paivita Click()  
  
    Set swApp = Application.SldWorks  
    Set Part = swApp.ActiveDoc  
    Part.EditRebuild3  
  
End Sub
```

Lomakkeesta voidaan myös ottaa vastaan käyttäjän syötteitä ja käyttää niitä ominaisuuksina luokkaobjekteissa tai tulostaa luokkaobjektien ominaisuuksia lomakkeelle.

5.7 Näkymän hallinta sovelluksessa

Makroa ajaessa SolidWorksin käyttöliittymä on jäädytetty käyttäjältä, se ottaa vastaan käskyjä vain makron käyttöliittymän kautta. Tästä syystä kokoonpanon pyörittämistä ohjataan lomakkeella. Lomakkeen napeille voidaan määritellä pyörittämiset, lähennykset ja loitonukset sekä SolidWorksin normaalit perusnäkyvät, kuten isometrinen näkymä tai Right eli näkymä suoraa kohden oikeaa sivua. Esimerkiksi nappia painamalla näkymä muuttuu Dimetric-näkymäksi ja kokoonpano sovitetään näkymään ruudulla kokonaan.

```
Private Sub naytaKaikki Click()  
  
    Set swApp = Application.SldWorks  
    Set Part = swApp.ActiveDoc
```

```
Part.ShowNamedView2 "*Dimetric", 7
Part.ViewZoomtofit
End Sub
```

Lähennys ja loitonnus toteutetaan ModelView-luokkaobjektin ZoomBy-Factor-metodilla, johon yksinkertaisesti määritellään kerroin mallin näkymälle. Arvot yli yhden suurentavat mallia eli lähentävät näkymää, alle yhden vastaavasti loitontavat (SolidWorks and Add-Ins API Help 2006h).

5.8 Modulaarisuuden tyypillisten ominaisuuksien toteuttaminen

5.81 Moduulien tuominen ja poistaminen

Moduulien tuomiseen kokoonpanossa käytetään Assemblydoc-luokkaobjektin metodia AddComponen4. Sen syntaksi on AssemblyDoc.AddComponent4(compName, configName, x, y, z), jossa parametreinä annetaan tiedot komponentin, tässä tapauksessa moduulin nimestä ja polusta (compName) ja sen mahdollisen konfiguraation nimestä (configName). Jos moduuli halutaan sijoittaa suoraan paikalleen, annetaan syntaksiin x, y ja z -koordinaattipisteet. Koordinaattipisteiden osoittama paikka ei välttämättä ole haluttu piste, sillä se riippuu moduulin bounding box -laatikosta eli virtuaalisesta laatikosta, jonka sisälle moduulimalli täsmälleen mahtuu (SolidWorks and Add-Ins API Help 2006c).

```
Part.AddComponent4 "C:\kansio\moduuli.SLDPRT",
0, 0, 0, 0
```

Tuotava moduuli täytyy olla ladattuna muistiin. Muistiin moduuli ladataan joko SldWorks.OpenDoc6 -metodilla ennen osan tuomista tai avaamalla kokoonpano, joka jo sisältää vähintään yhden lisättävän moduulitiedoston. (SolidWorks and Add-Ins API Help 2006c.)

Moduulin konfiguraatio voidaan tarvittaessa vaihtaa myös osan tuonnin jälkeen. Tällöin käytetään AssemblyDoc.CompConfigProperties4 -metodia (SolidWorks and Add-Ins API Help 2006d). Seuraavassa esimerkissä valitaan SelectByID2-metodilla component-tyyppinen moduuli, aktivoidaan siitä konfiguraatio nimeltään 1000_4000 ja poistetaan valinta.

```
boolstatus = Part.Extension.SelectByID2
("moduuli@kokoonpano", "COMPONENT", 0, 0, 0,
False, 0, Nothing, 0)
boolstatus = Part.CompConfigProperties4 2, _
```

```
0, True, True, "1000 4000", False  
Part.ClearSelection2 True
```

Kun moduuleja poistetaan kokoonpanosta, täytyy ne ensin valita ja käyttää tämän jälkeen yksinkertaista funktiota ModelDoc2.EditDelete

5.82 Vaihtokelpoisuus ja käyttö useammassa tuotteessa

Moduulien tyypeistä riippuen kokoonpano voidaan toteuttaa pelkillä näkyvyyden muutoksilla. Tällöin kokoonpanossa on valmiina kaikki moduulit omilla, täysin määrätyillä paikoillaan ja suppress-tilaa tai näkyvyyttä muuttamalla toteutetaan haluttu kokoonpano. Piilottaminen ja näyttäminen valitulle komponentille tapahtuu:

```
ModelDoc2.HideComponent2  
ModelDoc2.ShowComponent2
```

ja osalle:

```
FeatureManager.HideBodies  
FeatureManager.ShowBodies
```

Suppressointi poikkeaa osan piilottamisesta siten, että kokoonpanoa tai osaa käsitellään, kuten se ei olisi olemassa. Piilotettu osa on vain poissa näkyvistä, eikä ole editoitavissa. Piirrosten tulostamisessa kummatkaan osat eivät näy. Taulukossa 1. esitellään suppressoidun ja piilotetun komponentin eroavaisuuksia. Sen mukaan esimerkiksi moduulia käsiteltäessä suppressoituun piirteeseen yhteydessä olevat piirteet tai peilaukset ovat myös suppressoitu, piilotetussa osassa ne ovat editoitavissa. Suppressoinnissa luokkaobjekti voidaan valita ModelDoc2 tai Component2-luokkaobjektien väliltä riippuen suppressoitavan luokkaobjektityypistä. SetSuppression-metodi tarjoaa valittavaksi kolme tasoa (state), täysin suppressoidun ja täysin käytössä (ottaa myös alikomponentit käyttöön) olevan tason väliltä myös mahdollisuuden ottaa käyttöön vain valittu komponentti.

```
Component2.SetSuppression2 (state)
```

	Käytössä	Suppressoitu	Piilotettu
Ladattu muistiin	Kyllä	Ei	Kyllä
Näkyvissä	Kyllä	Ei	Ei
Mate käytössä	Kyllä	Ei	Kyllä
Yhteydessä olevat piirteet käytössä	Kyllä	Ei	Kyllä
Kokoonpanon piirteet käytössä	Kyllä	Ei	Kyllä
Editoitavissa	Kyllä	Ei	Ei
Latausnopeus	Normaali	Nopeampi	Normaali

TAULUKKO 1. Suppressoidun ja piilotetun komponentin erot (SolidWorks 2006 Online User's Guide - SP4.0 2006a).

5.83 Parametrinen modulaarisuus ja väylämodulaarisuus

Kun moduulille halutaan käyttäjän antama parametri, esimerkiksi moduulin pituus tai etäisyys toisen komponentin osasta, voidaan tieto lukea lomakkeen syöttölaatikosta. Vain muutama API-metodi palauttaa tiedon käyttäjän yksiköissä, muut funktiot vaativat ja palauttavat arvot metreinä ja radiaaneina (Spens 2004, 25). Tämä aiheuttaa sen, että sovelluksen käyttäjän syötteinä antama millimetrimitta on jaettava 1000:lla, jotta saadaan mitta metreiksi.

Käyttäjän syöte luetaan pituus-nimisestä tekstiruudusta ja asetetaan suoraan Sketchin pituus-nimiseksi mitaksi:

```
Part.Parameter("pituus@Sketch").SystemValue
= pituus.Text / 1000
```

API:n avulla myös Microsoft Exceliä voidaan käyttää tietojen hallintaan. Parametrisen ja väylämodulaarisuuden toteuttamisessa voidaan moduulien mitat tai etäisyydet lukea Excel-tilukoista. SolidWorks pystyy kaappaamaan yhteyden muihin VBA:ta tukeviin sovelluksiin, tässä tapauksessa Exceliin (SolidWorks and Add-Ins API Help 2006b). Esimerkiksi lomakkeen nappia painamalla luodaan yhteys avoimena olevaan Excel-sovellukseen ja kerätään tieto avoimena olevan taulukon solusta (Spens 2004, 135).

Metodi, jota käytetään Excel-tilukoiden aktivoimiseen, on GetObject. Tässä ohjelma tallentaa arvo-muuttujaan lueSolu-nappia painamalla avoimena olevasta taulukosta halutut tiedot.

```
Option Explicit
```

```

Dim swApp      As SldWorks.SldWorks
Dim Part      As SldWorks.ModelDoc2
Dim Excel     As Object
Dim arvo      As Integer

Private Sub lueSolu Click()

    'Tarttuu aktiiviseen SolidWorks-ohjelmaan
    Set swApp = Application.SldWorks

    'Tarttuu aktiiviseen tiedostoon
    Set Part = swApp.ActiveDoc

    'Tarttuu aktiiviseen Exceliin
    Set Excel = GetObject(, "Excel.Application")

    'Lukee arvon Excelin solussa A1
    arvo = Excel.Cells(1,1)

End Sub

```

5.84 Rakennuslohkomodulaarisuus

Koska rakennuslohkomoduulien paikat perustuvat täysin niiden välisiin rajapintaliitoksiin, ne voidaan sijoittaa paikalleen Mate-suhteen avulla, Assembly-luokkaobjektin API-metodilla AddMate3. Parametrinä funktiossa asetetaan Mate-suhteen tyyppin lisäksi useita piirteiden tyypeille ominaisia arvoja, kuten Distance Mate -tyypin etäisyys tai Angle Mate -tyypin kulma (SolidWorks and Add-Ins API Help 2006i).

Ongelmallista on rakennuslohkomodulaarisuudessa moduulien sijoittuminen toistensa suhteen. Muissa modulaarisuuden tyypeissä ne moduulit, joihin rajapintaliittyminen tapahtuu, ovat tiettyjä moduuleita ja perusyksiköitä, rakennuslohkomoduloinnissa ne voivat olla lähes pelkästään toisiinsa kiinnittyneitä ja koska valintaa ei voida makroa ajaessa tehdä vapaasti esimerkiksi hiirellä osoittamalla, vaihtoehtona on moduulien sijoittaminen ehdollisena, jolloin sovellus osoittaa moduulin paikan ja käyttäjä valitsee siihen moduulin.

Looginen tapa rakentaa kokoonpanoja on liittää moduuli suhteessa edellisenä tuotuun moduuliin. Koska SolidWorks sijoittaa piirrepuussa uusimman komponentin alimmaiseksi, on se näennäisesti helppo valita API-funktioiden getChildren ja getNextChildren avulla. Valitettavasti SolidWorks ei tunnista näillä funktioilla piirrepuun järjestystä siinä muodossa, kun se visuaalisesti on, vaan getNextChildren-funktio käy piirrepuuta läpi sattumanvaraisessa järjestyksessä. Ratkaisuksi tähän SolidWorksin maahantuojaan tukihenkilö esitti kirjaamattoman ominaisuuden, jossa tutkitaan AssemblyDoc-luokkaobjektin Feature-piirteiden tyyppisiä ja käydään läpi piirteitä FirstFeature ja GetNextFeature -

metodien avulla. Nämä metodit käyvät läpi piirrepuuta siinä järjestyksessä, millaisena ne näkyvät. Kun Featuren tyyppi on Reference, on se komponentti ja se pystytään valitsemaan Mate-suhteen toiseksi osapuoleksi. (Leivo 2006)

Jotta AddMate3-funktio voidaan toteuttaa, täytyy valittuina olla piirteet, joille Mate-komento annetaan. Esimerkissä toteutetaan Coincident Mate kahdelle pisteelle.

```
boolstatus =  
    Part.Extension.SelectByID2(piste a,  
    "DATUMPOINT", 0, 0, 0, False, 0, Nothing, 0)  
  
boolstatus =  
    Part.Extension.SelectByID2(piste b,  
    "DATUMPOINT", 0, 0, 0, True, 0, Nothing, 0)  
  
Set mateFeature = Part.AddMate3  
    (0, 0, False, 0, 0, 0, 0, 0, 0, 0, 0, 0,  
    False, longstatus)
```

Myös väylämodulaarisuus voidaan toteuttaa Mate-suhteen ja sijoittumisen mukaan Distance-typin avulla joko edellisenä tuodun osan ja uusimman osan väliin tai perusyksikön ja moduulin väliin.

Teknisesti oikein rakennetussa kokoonpanossa komponenttien paikat ovat täydellisesti määriteltynä. Tällaiseen paikan määrittelyyn täytyy olla tarvittava määrä Mate-suhteita tai mittoja. Jos Mate-suhteita tai mittoja on mahdotonta määrittää, voidaan vaihtoehtoisesti määrittely suorittaa AssemblyDoc-luokkaobjektin FixComponent-metodilla, joka kiinnittää moduulin liikkumattomaksi paikalleen.

5.9 Tulostaminen ja tallentaminen sovelluksessa

5.91 Tulostettavia ja tallennettavia tietoja

Modulaarisen tuotteen kokoonpanosovelluksessa tulostettavia ja tallennettavia asioita on useita. Sekä tulostus että tallentaminen ovat osa tiedon säilyttämistä ja julkistamista. Sovelluksessa voidaan tulostaa osaluetteloita ja tuotetietoa lomakkeelle tai tekstitiedostoon, tulostaa kokoonpanon käyttöliittymänäkymä, luoda tekniset piirustukset tai tallentaa kokoonpano toisessa tiedostomuodossa vaikkapa verkossa jaettavaksi tai säilytettäväksi tuotetiedon hallintajärjestelmissä.

5.92 Lomakkeeseen tulostaminen

Tuotetietoa tulostetaan käyttäjälle eri tavoin. Kun kyseessä on tuotteen CAD-malliin kytketty tieto, kuten mitta tai materiaali, voidaan se tulostaa käyttäjälle API-funktiolla. Muuta tuotetietoa voidaan tallentaa, säilyttää ja muuttaa sovelluksen muuttujataulukoissa, joista sitä voidaan julkistaa tarvittaessa käyttäjälle. Kun halutaan julkistaa tuotetietoa, se voidaan tulostaa suoraan lomakkeen listaan muuttujataulukosta seuraavasti:

```
Private Sub paivitaOsatNappi Click()  
  
    osaLuettelo.Clear  
    osaLuettelo.List() = osaTaulukko  
  
End Sub
```

Toisessa ohjelmakoodiesimerkissä Sketchin mitta "pituus" tulostetaan tekstikenttään ja luokkaobjektimallin hierarkian mukaan täytyy määritellä ylemmällä tasolla olevat luokkaobjektit.

```
boolstatus = Part.Extension.SelectByID2  
("pituus@Sketch", "DIMENSION", 0, 0, 0,  
False, 0, Nothing, 0)  
  
Set swDispDim = swSelMgr.GetSelectedObject5(1)  
Set swDim = swDispDim.GetDimension  
pituus.Value = swDim.SystemValue
```

5.93 Näkymän tulostaminen

Markkinointisovelluksessa hyödyllinen ominaisuus on tulostaa kokoonpanon näkymä SolidWorks-käyttöliittymästä esimerkiksi nopeaksi muistiinpanoksi. Se toteutetaan ModelDoc2-luokkaobjektin PrintPreview ja PrintDirect-metodein, jolloin näkymä tulostuu nopeasti ja suoraan työaseman oletustulostimesta.

Koska luokkaobjektina on ModelDoc2, käytetään tätä funktiota myös teknisten piirustusten tulostamiseen. Oletustulostin voi olla perinteinen tulostin tai virtuaalinen tulostin, kuten Adoben PDF Converter.

5.94 Tiedostojen tallentaminen

Vaikka SolidWorksillä tuotettu markkinointisovellus ei toimi verkon yli, voidaan siitä tulostaa kolmiulotteinen kokoonpanomalli tai tekninen piirustus jaettavaksi sähköisessä muodossa. SolidWorks-ohjelmassa

voidaan malli tallentaa sähköiseen julkaisemiseen soveltuvaksi tiedostotyyppiä, nimeltään eDrawings (.EASM kokoonpanoille, .EPRT osille ja .EDRW piirustuksille).

API-funktion ModelDoc2.SaveAs4 avulla tallennetaan mikä tahansa SolidWorksin tukema tiedostomuoto vaihtamalla vain tiedostolle parametrinä annettavan polun ja tiedoston nimen perään haluttu tiedostomuodopäätte (SolidWorks and Add-Ins API Help 2006j). Esimerkiksi eDrawings-tiedoston ja natiivin SolidWorks-tiedoston tallentamisen ohjelmakoodit poikkeavat vain tiedostopäätteen osalta.

```
Model.SaveAs4 ("D:\samples\kokoonpano.EASM", 0,
0, e, w)
Model.SaveAs4 ("D:\samples\kokoonpano.SLDASM", 0,
0, e, w)
```

SaveAs4-metodin syntaksissa ModelDoc2.SaveAs4 (Name, Version, Options, &Errors, &Warnings) päästään kahden viimeisen parametrin avulla tulostamaan mahdolliset virheilmoitukset ja varoitukset, jos tallennus ei onnistu (SolidWorks and Add-Ins API Help 2006j). Koska funktio on modelDoc2-luokkaobjektin metodi, se soveltuu tallentamaan niin kokoonpanoja, osia kuin piirustuksia.

5.95 Teknisten piirustusten luominen

Modulaarisesta tuotteesta voidaan makrojen avulla tuottaa tekniset piirustukset automaattisesti. Peruspiirustukset voidaan toteuttaa kolmella funktiokutsulla: SldWorks.NewDocument (templateName, paperSize, width, height) -funktiolla, joka avaa uuden dokumenttipohjan, piirustuksen; DrawingDoc.Create3rdAngleViews2 (modelName) –funktiolla, joka tuottaa kolmen käännön menetelmällä toteutetut projektiot tuotteesta ja DrawingDoc.InsertModelAnnotations3 (option, types, allViews, duplicateDims, hiddenFeatureDims, usePlacementInSketch) –funktiolla, joka lisää sille annettavien option-parametrien mukaiset mitat. (Spens 2004, 142–149.)

Ennen automaattista piirustusten luomista tulee tarkistaa, että avattu pohjadokumentti ja sen mitat ovat sopivia tuotteen piirustusten luomiseen.

5.96 Tekstiedostoon tulostaminen

Kun makroilla tallennetaan muuttujiin tietoa, se varastoidaan väliaikaisesti RAM-muistiin ja kun makron ajaminen lopetetaan ja kone

suljetaan, kaikki tieto näistä muuttujista häviää. Makroilla voidaankin tulostaa tietoa sekä lukea sitä ulkoisista tekstitiedostoista, mitä SolidWorks-ohjelmalla ei voi tehdä. (Spens 2004, 86.)

Ulkoinen tiedosto avataan ohjelmakoodissa Open-asetuksella, jolloin joko avataan olemassa oleva tiedosto tai luodaan sellainen. Asetukseen lisätään myös tunnisteluku väliltä 1-511, jolla päästään myöhemmin samaan tiedostoon käsiksi. Tieto tulostetaan tiedostoon Print # tai Input #-käskyllä, johon lisätään sama tunnisteluku kuin Open-asetuksessa. (Spens 2004, 86–91.)

Esimerkissä tulostetaan hinta-muuttuja hinnat-tekstitiedostoon yhdelle riville ja tämän jälkeen avataan se muistiossa.

```
Open "hinnat.txt" For Output As #1
  Print #1, "Tuotteen hinta: " & hinta
Close #1
Shell "notepad " & "hinnat.txt"
```

Tiedon lukeminen ja tallentaminen yhdeltä riviltä hinta-muuttujaan voidaan suorittaa seuraavasti:

```
Open "hinnat.txt" For Input As #1
  Line Input #1, hinta
Close #1
```

Markkinointisovelluksessa tiedostoon tulostamista ja niistä lukemista voidaan hyödyntää esimerkiksi osaluetteloissa tai hinnoissa. Erityisen oivallista tämä on silloin, kun tiedot ovat muuttuvia, eikä ohjelmakoodiin haluta koskea. Sovellus voi lukea aina käynnistettäessä tekstitiedostoista päivitettyt muutokset vaikkapa moduulien hinnoista ja näiden tietojen perusteella laskea tarjouksen moduulituotteelle.

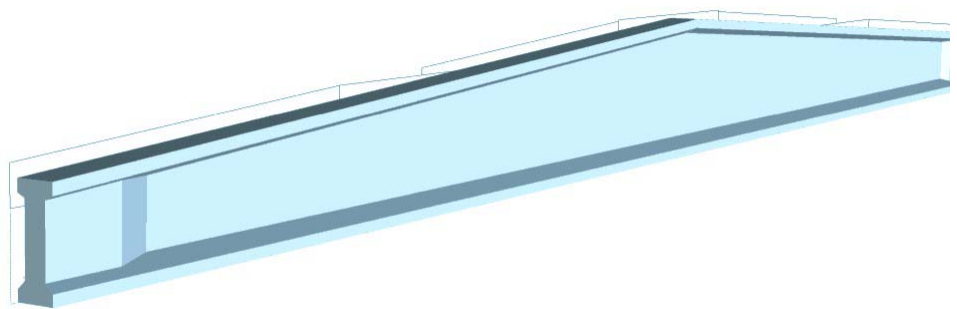
6 CASE: Betonimuottien kokoonpanosovellus

6.1 Esittely

Mecad Oy on tuotesuunnittelutoimisto, kone- ja laitesuunnitteluun keskittynyt muotoilun jälleenjalostaja. Yritys on erikoistunut 3d-suunnitteluun jo vuodesta 1994. Toimitusjohtaja Hannu Lahti on kiinnostunut jalostamaan 3d-ympäristössä tuotesuunnittelusta pidemmälle, tuotteiden käyttämisen ja myynnin apuvälineiksi. Tähän haasteeseen vastattiin luomalla sovelluskokeilu, joka toimisi myynnin ja markkinoinnin apuvälineenä tuottamaan moduuleista kokoonpano.

Tässä opinnäytetyössä keskityttiin etsimään ratkaisua teknisen tuotteen esittelyyn ja kohteeksi valittiin muottimoduuleista koostuvat betonimuotit. Tarkoituksena oli löytää yksi vaihtoehto yleispätevästä, visuaalisesta ja älykkästä keinosta tuottaa paketti, jota voitaisiin käyttää lähes minkä tahansa modulaarisen tuotteen kokoonpanossa. Sovelluksen tuottamiseen etsittiin kevyttä ohjelmointikeinoa. Näistä syistä ei tähän opinnäytetyöhön sisälly lopullista sovellusta, vaan edelleen kehitettävä versio modulaarisen tuotteen visuaalisesta kokoonpanosovelluksesta. Myöskään tuotteen mittojen ja teknisten ominaisuuksien oikeellisuus ei ole olennaista.

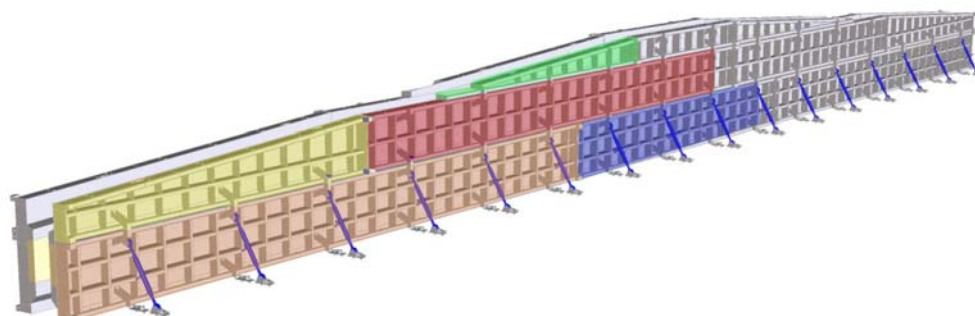
Sovellus on tuotelähtöinen. Asiakkaalla on valmiina tuotteen, betonisen HI- (kuva 14) tai I-palkin mitat. Sovelluksen lähtöasetelma on siis tuoteperhe, samanmuotoisia tuotteita, joiden mitoitus vaihtelee.



KUVA 14. HI- eli harja I-palkki (Lahti 2006).

Modulaarinen tuote on palkkien muotti, joka koostuu muottimoduuleista, muotokonsoleista, viistelijoista ja täytevanereista. Muottimoduulit (kuva 15) ovat modulaariselta lajiltaan rakennuslohkoja,

joiden rajapintoja ovat muottien päädyt, ylä- ja alapinnat sekä sivu, joka sijoittuu vasten palkkia. Konsolit, listat ja vanerit ovat tyypiltään vaihtokelpoisia moduuleita.



KUVA 15. Muottimoduulit (Lahti 2006).

6.2 Tavoitteet sovellukselle

Nykytilanne betonimuottiyrityksen myynnissä on esitellä tuotettaan 2d-piirroksin. Tämä on hankalaa ja vaikeasti ymmärrettävää, eikä se ole tarkoituksenmukaista. Tavoitteena on teknisen tuotteen tuki- ja myyntimateriaalin tuottaminen, markkinoinnin apuvälinesovellus, jolla myyjä esittelee tuotetta 3d-mallinnuksin asiakkaalle käyttöympäristön ollessa myyntitilanne ja kokoaa hetkessä asiakkaan tarpeita vastaavan tarjouskokoontalon.

Valmiita moduulimallien kuvia tai teknisiä liitosyksityiskohtia ei ole tuotesalaisuuden takia järkevää antaa asiakkaalle esittelytilanteessa, vaan mallit ovat yksinkertaistettuja tai peitettyjä 3d-malleja. Silti sovelluksen tulisi viestiä ammattimaisuudesta ja teknisestä osaamisesta, ei tuottaa lelumalleja asiakkaalle. Lisäksi uusia moduuleja tulisi voida liittää käyttöön.

Tehtävänä on pyrkiä yksinkertaiseen, helppokäyttöiseen käyttöliittymään, jossa ilman suurempaa 3d-ohjelmien hallintaa ja koulutusta pystyy tuottamaan kokoonpanon. Lisäksi sovellus kertoisi tarpeenmukaiset tiedot yksittäisistä moduuleista ja kokoonpanosta: moduulien mitat ja osaluettelon. Sovelluksen käyttäjänä toimisivat betoniteollisuuden laitteiden myyjät, kohderyhmänä betoniteollisuuden laitehankkijat.

6.3 Sovelluksen toteuttaminen

Koska kohteena on teollisuuden tekninen tuote, on oletettu, että myynti- ja markkinointitilanteessa on käytössä SolidWorks-ohjelma. Tällöin ilman ylimääräisiä ohjelmistohankintakustannuksia voidaan rakentaa makroja apuna käyttäen toimiva kokoonpanosovellus, joka voidaan tuottaa ilman laajaa ohjelmointitaustaa. Sovelluksen käyttöliittymä toteutettiin lomakkeella. Sen käyttöä puoltavat Windows-ympäristöön tottuneelle tutut napit ja ulkonäkö.

Tiedostojen rakenne toteutettiin niin, että kokoonpanoon on tuotu valmis I-palkki, jonka mittoja käyttäjä muuttaa lomakkeessa. Palkkitiedostossa on mukana palkin lisäksi viistelilat, muotokonsolit ja vanerit, joita tarpeen mukaan piilotetaan tai näytetään lomakkeen napeilla. Muottitiedosto koostuu yhdestä muotista, jossa on kaikkien tarvittavien muottimitoitusten mukaiset konfiguraatiot, joita kutsutaan uusiksi muotteja tuotaessa. Näin muottia kasattaessa kokoonpanotiedoston koko pysyy hallinnassa, sillä se käyttää vain kahta osatiedostoa.

Muottilohkojen rajapintakiinnitys muodostuu Coincident Mate-suhteella, jota varten muottitiedostojen kulmiin lisättiin nimetyt pisteet. Suurimmaksi hankaluudeksi muodostui muottilohkojen valinta Mate-suhteen luomiseksi. Pelastus oli SolidWorksin maahantuojaan tukipalveluhenkilön keino käyttää kirjaamatonta Feature-piirteen tyyppi-ominaisuutta, jonka avulla piirrepuun viimeinen osa saadaan valittua. Funktiolla viimeinenKomponentti käydään läpi järjestyksessä piirrepuu ja viimeisen komponentin nimi jää komponentti-muuttujaan talteen.

```
Public Function viimeinenKomponentti(komponentti
    As String) As Variant

    Dim swApp          As SldWorks.SldWorks
    Dim swModel        As SldWorks.ModelDoc2
    Dim swFeat         As SldWorks.feature
    Dim swComponent    As SldWorks.Component2

    Set swApp = Application.SldWorks
    Set swModel = swApp.ActiveDoc
    Set swFeat = swModel.FirstFeature

    Do Until swFeat Is Nothing
        If swFeat.GetTypeName = "Reference" Then

            Set swComponent = swFeat.GetSpecificFeature
            komponentti = swComponent.Name2

        End If
    End Do
```

```
Set swFeat = swFeat.GetNextFeature
Loop
End Function
```

Koska muotti ja palkki kokonaisuudessaan ovat symmetrisiä, on muottilohkoista piilotettu näkyvistä toinen puoli. Tämä auttaa näkemään muotin keskellä olevien konsolien ja vanerien muodot. Vasta kun muotti on valmis, näytetään muottilohkojen toiset puolet.

Osaluettelo toteutettiin lomakkeeseen ja tekstitiedostoon taulukkomuuttujalla osaTaulukko, johon on talletettu muottilohkojen tunnistemittanimet ja niiden lukumäärät muotissa. Tekstitiedostoon osaluettelon kirjoittaminen tapahtuu for-luupilla.

```
Open "osaluettelo.txt" For Output As #1

Print #1, "Osaluettelo: " & osaLuetteloNimi
Print #1,
Print #1, "Muottilohkot:"
Print #1,

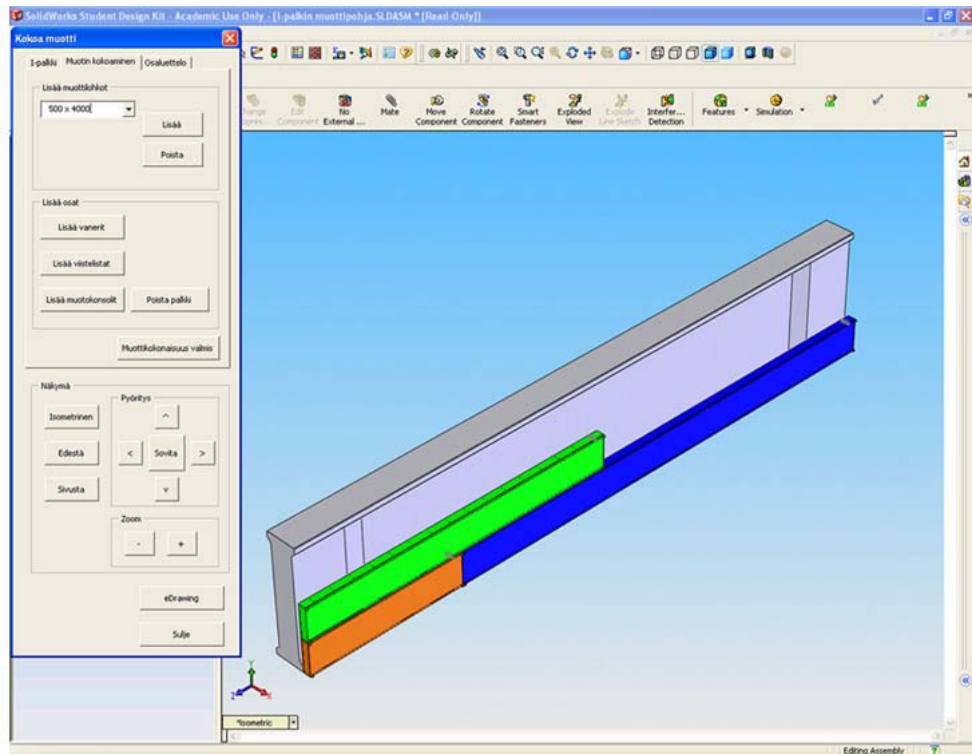
For i = 0 To UBound(osaTaulukko)

    If osaTaulukko(i, 1) <> 0 Then
        Print #1, osaTaulukko(i, 0); Tab(20);
        osaTaulukko(i, 1)
    End If

Next i
Close #1
```

6.4 Sovelluksen käyttäminen

Tuotteen kokoonpano pohjustetaan avaamalla I-palkin muottipohja.SLDASM -niminen kokoonpanotiedosto SolidWorks-ohjelmaan ja käynnistämällä Muotin kokoaminen.swp -makro. Tuotteen varsinainen kokoonpano aloitetaan määrittelemällä lomakkeen syöttökenttiin mitat I-palkille, jonka jälkeen tuodaan yksitellen muottimoduulit alavetovalikosta (kuva 16).



KUVA 16. Muotin kokoaminen sovelluksessa.

Kun muotti on kokonaisuudessaan valmis, siihen lisätään viistelistat, muotokonsolit ja täytevanerit. Lomakkeelle saadaan näkyviin osaluettelo, joka voidaan tulostaa myös tekstitiedostoon. Lisäksi muotti voidaan tallentaa eDrawing-tiedostoksi ja hallita näkymää lomakkeen napeilla SolidWorksin käyttöliittymän näkymässä pyörittämällä sekä lähentämällä ja loitontamalla. Liitteessä 1. nähdään lomake ja kokoonpanon eri vaiheet. Suljettaessa lomaketta muotti voidaan tallentaa käyttäjän haluamalla nimellä normaalina SolidWorksin kokoonpanotiedostona.

6.5 Jatkokehitysehdotelmia

Seuraavana haasteena on ohjelmakoodin päivittäminen uudempaan Visual Basic -kielen versioon (Visual basic 2005) ja sovelluksen kehittäminen makropohjaisesta ohjelmakoodista SolidWorksin Add-In -sovellukseksi.

Koska sovellus on kokeilu kokoonpanon tuottamisesta visuaalisin keinoin, ovat jatkokehityksen mahdollisuudet laajat. Jotta sovellus vastaisi loppukäyttäjän tarpeita, olisi niitä tutkittava perusteellisemmin. Luonnollisesti tulisi varmistaa sovelluksen yleinen virheidenhallinta

sekä mallien tekniset riippuvuudet ja oikeellisuus. Näiden käytännön varmistuksien jälkeen voisi sovellusta kehittää edelleen betonimuottiyrityksessä laajempaan käyttöön.

Kokoonpanosovelluksen yksi laajennusmahdollisuus olisi sisällyttää ohjelma muottitoimitukseen ja liittää siihen päivitettyt tiedot asiakkaan ostamista muottilohkoista, lohkothan eivät ole kertakäyttöisiä. Tällöin betonimuottiyrityksen asiakas voisi tarkistaa nopeasti uusiin palkkeihinsa tarvitsemat muottilohkot ja tarvittaessa tilata niitä lisää. Yksi kehitysidea olisi levittää sovellusta SolidWorksin Add-in -muodossa ladattavaksi muottilohkovalmistajan verkkosivuilta halukkaille asiakkaille. Näin toteutuisivat laajemmin sovelluksen markkinointihyödyt, kokoonpanon toiston mahdollisuudet kytkettynä ajankäyttöön. Tosin casen kaltaisessa tapauksessa, kun sovellus toimii toisen ohjelman päällä, täytyy asiakasyrityksessä olla alla toimiva ohjelma, tässä tapauksessa SolidWorks.

Jos sovellusta kehitettäisiin niin, että se toimisi yleispätevänä kokoonpanosovelluksena, tulisi ohjelmakoodia laajentaa rutkasti ja tarjota mahdollisuudet kaikkien modulaarisuuden lajityyppien toteuttamiseen moduulien rajapintojen ehdoilla. Tällä hetkellä sovellus on tuotettu toimimaan vain määrättyjen betonilohkomoduulien kanssa.

6.6 Casen arviointi

Casen tuottaminen antoi kovasti haasteita. Ohjelmoimaan sangen tottumattomana jouduin kaivamaan tietoa laajasti, eikä suoria ohjeita juurikaan löytynyt. Kuitenkin onnistuin Mecad Oy:n mielestä täyttämään suurimmat sovellukselle asetetut haasteet, sovelluksella on helppo luoda kokoonpano ja esitellä modulaarista tuotetta ammattimaisesti ilman laajaa 3d-ohjelmien tuntemusta.

Mieleeni jäävät silti kaiheartamaan muut ratkaisumallit, joilla voitaisiin tuottaa vastaava sovellus ilman sen alla pyörivää teknistä suunnitteluohjelmaa, SolidWorks-ohjelmaa. Tällöin sen käyttö ei riippuisi koneella olevista ohjelmista ja sovellusta pystyttäisiin käyttämään myös muunlaisen ratkaisun tuottamiseen kuin teknisen teollisuustuotteen kokoonpanoon. Tämän opinnäytetyön perusteella tällainen sovellus vaatisi kuitenkin varsin perusteellista ohjelmointiosaamista.

7 YHTEENVETO

Modulaarisen tuotteen 3d-markkinointisovelluksen tuottamistekniikan valinta riippuu täysin sen käyttöympäristöstä ja tuotteen luonteesta. Kun tuote on kuluttajalle suunnattu verkkosovellus, on otettava huomioon viihteellisyyden ohessa verkon tuomat vaatimukset tekniikkaan. Erityisen kiinnostavana tällaisen sovelluksen tuottamiseen tämän opinnäytetyön tutkimuksen perusteella pidän X3D-standardia, josta toivon kehittyvän Flashin kaltaisen multimediaformaatin. Kolmiulotteisuuden on jo aika tulla rytinällä internetiympäristöön.

SolidWorks-ohjelmalla markkinointisovelluksen tuottaminen lienee harvinaista, tämän opinnäytetyön yhteydessä havaitut sovellukset olivat suunnitteluautomaatteja, eli ammattisuunnittelijoille työvälineeksi tarkoitettuja sovelluksia, eivät markkinoinnin käyttöön. Case-tapauksessa teknisen suunnitteluohjelman päälle toteutettava sovellus on kuitenkin mainio vaihtoehto markkinointisovelluksen tuottamiseen. Tällä taataan tekninen hyöty ja oikeellisuus markkinointisovelluksessa luotavista kokoonpanoista mutta myös luodaan yksinkertainen sovellus SolidWorks-ohjelmaa tuntemattoman käyttöön. Suurimpana hyötynä tällä tekniikalla tuotetussa sovelluksessa pidän sen mahdollisuutta kytkeä sovelluksen tuottama kokoonpano koko tuotteen elinkaareen, tuotannon hallinnoimiseen ja tuotetiedon hallintaan.

Kaikilla esittämilläni tavoilla voidaan toteuttaa tarkoituksenmukainen modulaarisen tuotteen kolmiulotteisuutta käyttävä markkinointisovellus, kunhan sovelluksesta saatava hyöty ja olemassa olevat ohjelmat sekä taidot tunnistetaan.

LÄHTEET

Aavameri, Leena. Yahooon Sanders: Elämystalous tekee tuloaan [verkkojulkaisu]. Digitoday, Media. 22.10.2002 [viitattu 18.2.2007]. Saatavissa:

http://www.digitoday.fi/page.php?page_id=11&news_id=20028583

Aliohjelma [verkkojulkaisu]. Wikipedia, 2006 [viitattu 11.2.2007].

Saatavissa: <http://fi.wikipedia.org/wiki/Aliohjelma>

American Diamond [verkkosovellus]. Construct 3D, 2005 [viitattu 20.3.2007]. Saatavissa:

<http://www.construct3d.com/x3diamond/x3diamond.htm>

API [verkkojulkaisu]. Tietotekniikan termitalkoot Sanastokeskus TSK, 2001 [viitattu 16.1.2007]. Saatavissa: <http://www.tsk.fi/termitalkoot/>

Box - Web3D.org [verkkojulkaisu]. Web3D Consortium, 2007 [viitattu 15.2.2007]. Saatavissa: <http://www.web3d.org/x3d/wiki/index.php/Box>

CadWorks Oy. SolidWorks [verkkojulkaisu]. CadWorks Oy, 2007 [viitattu 13.3.2007]. Saatavissa;

<http://www.cadworks.fi/asp/system/empty.asp?P=53&VID=default&SID=529913244027997&S=0&C=19835>

Collision - Web3D.org [verkkojulkaisu]. Web3D Consortium, 2007 [viitattu 15.2.2007]. Saatavissa:

<http://www.web3d.org/x3d/wiki/index.php/Collision>

Demicron – Interactive Web3D – Streaming Rich Media [verkkojulkaisu]. Demicron, 2007a [viitattu 28.3.2007]. Saatavissa:

<http://www.demicron.com/wirefusion/index2.html>

Demicron – Plug-in free Rich Media – Interactive 2d and Web3D – Streaming Video and Audio [verkkojulkaisu]. Demicron, 2007b [viitattu 28.3.2007]. Saatavissa:

http://www.demicron.com/store/store_europe.html

F-Secure Anti-Virus 5 manual [verkkojulkaisu]. F-Secure Oyj, 2005 [viitattu 12.3.2007]. Saatavissa: http://www.f-secure.com/export/system/fsgalleries/manuals/fsav_54x_fin.pdf

Halvorson, Michael. 2006. Visual Basic 2005 Tehokas hallinta. readme.fi. 2006. Helsinki.

Heinonen, Petri. VBA perusteet erittäin lyhyesti [verkkojulkaisu]. Jyväskylän yliopiston avoin yliopisto, 2005 [viitattu 18.1.2007]. Saatavissa: <http://appro.mit.jyu.fi/doc/tiedonhallinta/vba/>

How to Provide Engaging, Relevant and Accurate Product Information to Customers: White paper [verkkojulkaisu]. UK: Web 3D Studio, 2006 [viitattu 16.2.2007]. Saatavissa: <http://www.web3d.co.uk/pdf/Web3DStudioWhitepaper0306.pdf>

Huhta, Esko. Java 3D [verkkojulkaisu]. Helsinki: Helsingin teknillinen korkeakoulu. Informaatioverkostot Studio 4, Essee, 2005 [viitattu 10.1.2007]. Saatavissa: <http://users.tkk.fi/~ehuhta/studio4/java3D/Java3D.html>

Häubl, Gerald ja Figueroa, Pablo. Interactive 3D Presentations and Buyer Behavior [verkkojulkaisu]. Kanada, University of Alberta, 2002 [viitattu 16.2.2007]. Saatavissa: <http://www.aarkid.com/research/p744-figueroa.pdf>

Java SE Technologies at a Glance [verkkojulkaisu]. Sun Microsystems, 2007 [viitattu 15.2.2007]. Saatavissa: <http://java.sun.com/javase/technologies/index.jsp>

IKEA | Lataa suunnitteluohjelma [verkkojulkaisu]. Inter IKEA Systems B.V., 2007a [viitattu 15.1.2007]. Saatavissa: http://www.ikea.com/ms/fi_FI/complete_kitchen_guide/planner_tool/download/index.html

IKEA | Visiomme – Parempi arkipäivä [verkkojulkaisu]. Inter IKEA Systems B.V., 2007b [viitattu 15.1.2007]. Saatavissa: http://www.ikea.com/ms/fi_FI/about_ikea/our_vision/better_life.html

IKEA uses the Furnish Home Edition supplied by Geac as the basis of the consumer-based application, IKEA Home Planner [verkkojulkaisu]. Infor, 2006 [viitattu 15.2.2007]. Saatavissa: http://www.runtime.geac.com/object/IKEA_RT.html

Java [verkkojulkaisu]. Wikipedia, 2007 [viitattu 11.2.2007]. Saatavissa: <http://fi.wikipedia.org/wiki/Java>

Kerätär, Matti. 2005. Tehokasta asiakasohjautuvuutta tuotekonfiguroinnilla. Future maailma, Future CAD Oy:n asiakaslehti 4/2005, 12-13.

Krieger, Stephanie. Makrot tutuiksi: Makrojen kuvaus ja käyttötarkoitus [verkkojulkaisu]. Microsoft Corporation, 2007 [viitattu 2.3.2007]. Saatavissa: <http://office.microsoft.com/fi-fi/help/HA100072101035.aspx>

Laaksonen, Antti. Visual Basic -opas osa 5 - Aliohjelmat ja funktiot [verkkójulkaisu]. Ohjelmointiputka [viitattu 11.2.2007]. Saatavissa: http://www.ohjelmointiputka.net/opas.php?tunnus=vbo_5

the LEGO Group. 2007a. LEGO® Digital Designer 2.0 Readme. Readme-tiedosto.

Leivo, Heikki. CadWorks Oy, Helsingintie 44, 04430 Järvenpää. Sähköpostihaastattelu 29.11.2006.

Lighting component. All X3D Specifications and Encodings, X3D Public Specifications, Extensible 3D (X3D), ISO/IEC 19775-1:2004. Part 1 : Architecture and base components [verkkójulkaisu]. Web3D Consortium, 2006 [viitattu 15.2.2007]. Saatavissa: <http://www.web3d.org/x3d/specifications/>

Macro [verkkójulkaisu]. Wikipedia, 2007 [viitattu 3.3.2007]. Saatavissa: <http://en.wikipedia.org/wiki/Macro>

Markkinointi [verkkójulkaisu]. Wikipedia, 2007 [viitattu 11.2.2007]. Saatavissa: <http://fi.wikipedia.org/wiki/Markkinointi>

Media Machines – Make Web 3D virtual worlds and things – Flux Studio [verkkójulkaisu]. Media Machines, 2007 [viitattu 28.3.2007]. Saatavissa: <http://www.mediamachines.com/make.php>

Merisavo, Marko. Digitaalinen markkinointi. Miten lujitat asiakassuhteita digidialogilla? [verkkójulkaisu]. Helsinki: Helsingin kauppakorkeakoulu, 2006 [viitattu 18.2.2007]. Saatavissa: <http://www.divia.fi/?action=file&id=124&file=124.10>

Modulointi [verkkójulkaisu]. Tampereen teknillinen yliopisto, 2006 [viitattu 17.1.2007]. Saatavissa: [http://pe.tut.fi/courses/TTE5026.nsf/262de9cadd394639c2257051003d498a/c3ff470cf62d6796c22568e90032c288/\\$FILE/Modulointi_K2006.pdf](http://pe.tut.fi/courses/TTE5026.nsf/262de9cadd394639c2257051003d498a/c3ff470cf62d6796c22568e90032c288/$FILE/Modulointi_K2006.pdf)

Moduuli [verkkójulkaisu]. Wikipedia, 2006 [viitattu 19.1.2007]. Saatavissa: <http://fi.wikipedia.org/wiki/Moduuli>

Ohjelmointirajapinta [verkkójulkaisu]. Wikipedia, 2006 [viitattu 16.1.2007]. Saatavissa: <http://fi.wikipedia.org/wiki/Ohjelmointirajapinta>

Pihkala-Bäckström, Erja. Media Debut 4: Strategiana markkinointi. [verkkójulkaisu]. Espoo, Vantaa: EVTEK Mercuria Business School 2003 [viitattu 18.2.2007]. Saatavissa:

<http://users.evtek.fi/~maijal/zv03/Luentomateriaalit/MDStrategiat403.ppt>

Production Basics [verkkajulkaisu]. Production Basics, 2007 [viitattu 16.2.2007]. Saatavissa: <http://www.pbasics.com/>

Production Basics, 3d Configurator [verkkosovellus]. Production Basics, 2007 [viitattu 16.2.2007]. Saatavissa: <http://www.pbasics.com/Configurator/c-leg.html>

Raulas, Mika ja Merisavo, Marko. Markkinointi digitaalisessa murroksessa. DiVia - Digitaalinen Viestintä ja Asiakkuudet – Tutkimusprojektin loppuraportti [verkkajulkaisu]. Suoramarkkinointi-instituutti ja LTT – Tutkimus Oy Elektronisen Kaupan Instituutti, 2004 [viitattu 20.2.2007]. Saatavissa: <http://www.divia.fi/?action=file&id=52&file=52.pdf>

Rautajoki, Marja. Johdatus verkkoteknologioihin: Kaistat [verkkajulkaisu]. VirtuaaliAMK, 2005. [viitattu 20.2.2007] Saatavissa: <https://www.virtuaaliamk.fi/opintokokonaisuudet/56QfjzFJ/1094208209451/1094209510107/1094209817544/1094211042942.html.stx>

Rope, Timo. Business to business markkinointi [verkkajulkaisu, elektroninen kirja]. Porvoo: WS Bookwell Oy, 1998, 2006

Saksa, Tommi. Java-Applet [verkkajulkaisu]. HAMK, tietojenkäsittelyn koulutusohjelma, 2001 [viitattu 16.2.2007]. Saatavissa: <http://trade.hamk.fi/~saksa/www/javaapplet.htm>

Sarinko, Kati. Asiakaskohtaisesti muunneltavien tuotteiden massaräätälöinti, konfigurointi ja modulointi, Diplomityö [verkkajulkaisu]. Espoo: Teknillinen korkeakoulu, 1999 [viitattu 14.3.2007]. Saatavissa: <http://www.soberit.hut.fi/pdmg/papers/Sari99Mas.pdf>

SolidWorks and Add-Ins API Help. 2006a. SolidWorks API Help > Welcome to SolidWorks API Help

SolidWorks and Add-Ins API Help. 2006b. SolidWorks API Help > APIs > Assembly Objects > Physical Simulation > Animation > Object Description

SolidWorks and Add-Ins API Help. 2006c. SolidWorks API Help > APIs > Application Objects > AssemblyDoc > Methods > AddComponent4

SolidWorks and Add-Ins API Help. 2006d. SolidWorks API Help > APIs > Application Objects > AssemblyDoc > Methods > CompConfigProperties4

SolidWorks and Add-Ins API Help. 2006e. SolidWorks API Help > Programmers Guide > SolidWorks API Object Model > Overview

SolidWorks and Add-Ins API Help. 2006f. SolidWorks and Add-Ins API Help Overview > Using SolidWorks and Add-Ins API Help > Syntax Conventions

SolidWorks and Add-Ins API Help. 2006g. SolidWorks API Help > APIs > Application Objects > ModelDoc > ModelDoc2 > Object Description

SolidWorks and Add-Ins API Help. 2006h. SolidWorks API Help > APIs > UserInterfaceObjects > ModelView > Methods > ZoomByFactor

SolidWorks and Add-Ins API Help. 2006i. SolidWorks API Help > APIs > Application Objects > AssemblyDoc > Methods > AddMate3

SolidWorks and Add-Ins API Help. 2006j. SolidWorks API Help > APIs > Application Objects > ModelDoc2 > Methods > SaveAs4

SolidWorks 2006 Online User's Guide - SP4.0. 2006a. Troubleshooting > Tips > Graphics Adapters and Drivers

Spens, Mike. 2004. Automating SolidWorks 2004 using Macros. Schroff Development Corporation.

Tuotekonfigurointi [verkkojulkaisu]. Cadon Oy, 2006 [viitattu 15.2.2007]. Saatavissa: <http://www.cadon.fi/default.asp?viewID=2272>

Visual Basic for Application [verkkojulkaisu]. Wikipedia, 2007 [viitattu 16.1.2007]. Saatavissa: http://en.wikipedia.org/wiki/Visual_Basic_for_Applications

Visual Basic for Application Frequently Asked Questions [verkkojulkaisu]. Microsoft Corporation, 2007 [viitattu 16.1.2007]. Saatavissa: <http://msdn2.microsoft.com/en-us/isv/bb190540.aspx>

Visual Basic Reference. 2006a. Visual Basic Conceptual Topics > Writing Visual Basic Statements

Visual Basic Reference. 2006b. Visual Basic Language Reference > Statements > M-Z > Option Explicit Statement

Visual Basic Reference. 2006c. Visual Basic Language Reference > Groups > Data Type Summary

Visual Basic Reference. 2006d. Visual Basic Conceptual Topics > Understanding Objects, Properties, Methods, and Events

VRML [verkkajulkaisu]. Wikipedia, 2007 [viitattu 16.1.2007]. Saatavissa: <http://en.wikipedia.org/wiki/VRML>

Vuori, Matti ja Kivistö-Rahnasto, Jouni. WWW ja muut internetteknikat tuotteiden muuntajina [verkkajulkaisu]. 2000 [viitattu 16.1.2007]. <http://www.kotiposti.net/mvuori/julkaisuluettelo/liitteet/internet-tuotteissa.pdf>

Web3D.org: X3D Viewers, Browsers and Plug-ins [verkkajulkaisu]. Web3D Consortium, 2007 [viitattu 15.2.2007]. Saatavissa: http://www.web3d.org/tools/viewers_and_browsers/

What are profiles? - Web3D.org [verkkajulkaisu]. Web3D Consortium, 2006 [viitattu 15.2.2007]. Saatavissa: http://www.web3d.org/x3d/wiki/index.php/What_are_profiles%3F

What is X3D? [verkkajulkaisu]. Web3D Consortium, 2007 [viitattu 15.2.2007]. Saatavissa: <http://www.web3d.org/about/overview/>

Wikla, Arto. Ohjelmoinnin perusteita Java-kielellä: 1.2 Java-kielestä [verkkajulkaisu]. Helsinki: Helsingin yliopiston tietojenkäsittelytieteen laitos, 2005 [viitattu 5.2.2007]. Saatavissa: <http://www.cs.helsinki.fi/u/wikla/Ohjelmointi/Sisalto/1/Javat.html>

X3D FAQ [verkkajulkaisu]. Web3D Consortium, 2007 [viitattu 15.2.2007]. Saatavissa: <http://www.web3d.org/about/faq/#general-4>

X3D Scenarios and Case Studies [verkkajulkaisu]. Web3D Consortium, 2007 [viitattu 15.2.2007]. Saatavissa: <http://www.web3d.org/casestudies/>

Xj3D – Java Based X3D Toolkit and X3D Browser [verkkajulkaisu]. Web3D Consortium, 2007 [viitattu 15.2.2007]. Saatavissa: <http://www.web3d.org/x3d/xj3d/>

KUVALÄHTEET

KUVA 1. Modulointi [verkkojulkaisu]. Tampereen teknillinen yliopisto, 2006 [viitattu 17.1.2007]. Saatavissa:

[http://pe.tut.fi/courses/TTE5026.nsf/262de9cadd394639c2257051003d498a/c3ff470cf62d6796c22568e90032c288/\\$FILE/Modulointi_K2006.pdf](http://pe.tut.fi/courses/TTE5026.nsf/262de9cadd394639c2257051003d498a/c3ff470cf62d6796c22568e90032c288/$FILE/Modulointi_K2006.pdf)

KUVA 2. Häubl, Gerald ja Figueroa, Pablo. Interactive 3D Presentations and Buyer Behavior [verkkojulkaisu]. Kanada, University of Alberta, 2002 [viitattu 16.2.2007]. Saatavissa:

<http://www.aarkid.com/research/p744-figueroa.pdf>

KUVA 3. Kuvakaappaus sovelluksesta. Production Basics, 3d Configurator [verkkosovellus]. USA 2007 [viitattu 16.2.2007]. Saatavissa:

<http://www.pbasics.com/Configurator/c-leg.html>

KUVA 4. Kuvakaappaus sovelluksesta. American Diamond [verkkosovellus]. Construct 3D, 2005 [viitattu 20.3.2007]. Saatavissa:

<http://www.construct3d.com/x3diamond/x3diamond.htm>

KUVA 5. Kuvakaappaus sovelluksesta. IKEA Home Planner Kitchen. 2006.

KUVA 6. Kuvakaappaus sovelluksesta. IKEA Home Planner Kitchen. 2006.

KUVA 7. Kuvakaappaus sovelluksesta. IKEA Home Planner Kitchen. 2006.

KUVA 8. Kuvakaappaus sovelluksesta. the LEGO Group. 2007. LEGO® Digital Designer 2.0

KUVA 8. Kuvakaappaus sovelluksesta. the LEGO Group. 2007. LEGO® Digital Designer 2.0

KUVA 10. Kuvakaappaus sovelluksesta. Production Basics, 3d Configurator [verkkosovellus]. USA 2007 [viitattu 16.2.2007]. Saatavissa:

<http://www.pbasics.com/Configurator/c-leg.html>

KUVA 11. Kuvakaappaus sovelluksesta. Production Basics, 3d Configurator [verkkosovellus]. USA 2007 [viitattu 16.2.2007]. Saatavissa:

<http://www.pbasics.com/Configurator/c-leg.html>

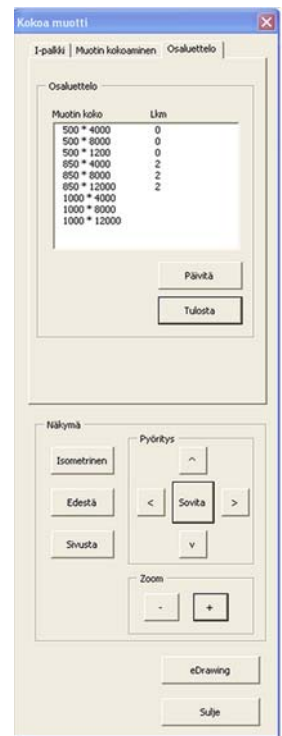
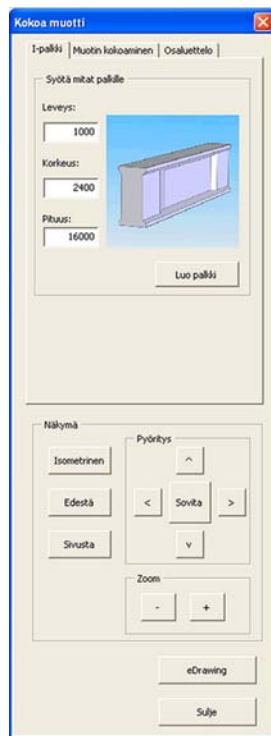
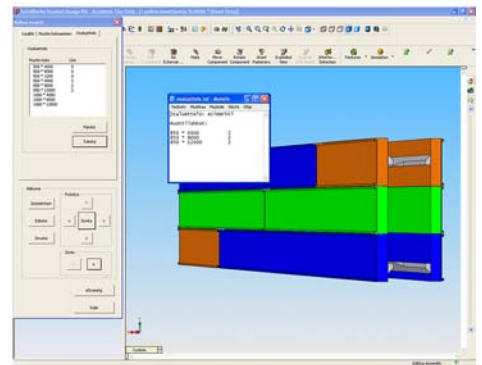
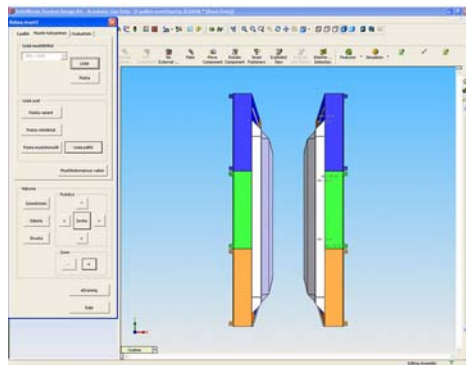
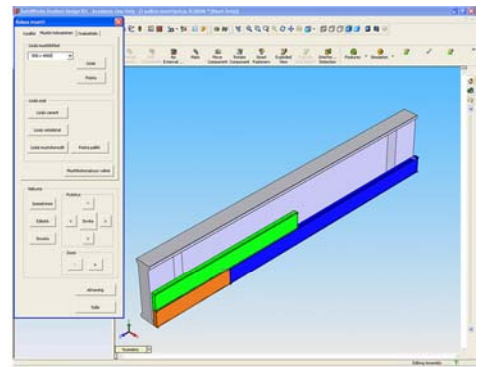
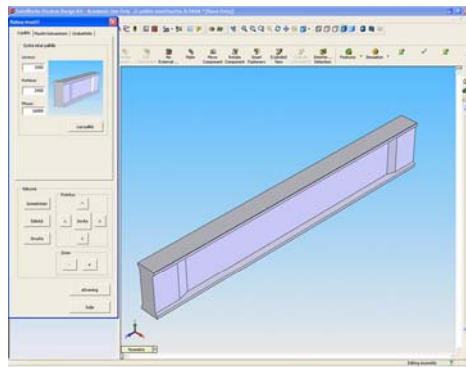
KUVA 14. Lahti, Hannu. HI-palkki 2006.

KUVA 15. Lahti, Hannu. Muottimoduulit 2006.

TAULUKKOLÄHTEET

TAULUKKO 1. SolidWorks 2006 Online User's Guide - SP4.0. 2006a.
Assemblies > Improving Assembly Performance > Comparing Components Suppression States

LIITTEET



LIITE 1. Kuvakaappauksia tuotetusta sovelluksesta.

LIITE 2. Cd-levyllä modulaarisen tuotteen markkinointisovellus Solid-Works-ohjelmaan