

Jari M. Ahola, Jani Hovila, Eero Karhunen, Kalervo Nevala,
Timo Schäfer & Tom Nevala

Moniteknisen piensarjatuotteen digitaalinen tuoteprosessi

Moniteknisen piensarjatuotteen digitaalinen tuoteprosessi

Jari M. Ahola, Jani Hovila, Eero Karhunen, Kalervo Nevala,
Timo Schäfer & Tom Nevala



ISBN 978-951-38-7773-6 (nid.)

ISSN 1235-0605 (nid.)

ISBN 978-951-38-7774-3 (URL: <http://www.vtt.fi/publications/index.jsp>)

ISSN 1455-0865 (URL: <http://www.vtt.fi/publications/index.jsp>)

Copyright © VTT 2011

JULKAISIJA – UTGIVARE – PUBLISHER

VTT, Vuorimiehentie 5, PL 1000, 02044 VTT

puh. vaihde 020 722 111, faksi 020 722 4374

VTT, Bergsmansvägen 5, PB 1000, 02044 VTT

tel. växel 020 722 111, fax 020 722 4374

VTT Technical Research Centre of Finland, Vuorimiehentie 5, P.O. Box 1000, FI-02044 VTT, Finland
phone internat. +358 20 722 111, fax +358 20 722 4374

Toimitus Marika Leppilahti

Kopijyvä Oy, Kuopio

Jari M. Ahola, Jani Hovila, Eero Karhunen, Kalervo Nevala, Timo Schäfer & Tom Nevala. Moniteknisen piensarjatuotteen digitaalinen tuoteprosessi [Digital product process of a multitechnological small series product]. Espoo 2011. VTT Tiedotteita – Research Notes 2592. 121 s. + liitt. 37 s.

Avainsanat product life-cycle management, product development, concurrent design, virtual design, simulation

Tiivistelmä

Viime vuosikymmenten nopea teknologinen kehitys on tarjonnut mahdollisuuden ajatella tuotekehitystä ja tuotteistamista uudella tavalla, jonka mukaan tuoteprosessi kattaa tuotteen koko elinkaaren tarvekartoituksesta uudelleen kierrätykseen. ”Moniteknisen tuotteen digitaalisen tuoteprosessin kehittäminen tehokkaaseen piensarjatuotteiden suunnitteluun ja tuotantoon” (MoniDigi) -tutkimusprojektin tavoitteena on kehittää moderneilla virtuaalisuunnittelun työkaluilla varustettu moniteknisten tuotteiden kehitysympäristö ja digitaalisen tuoteprosessin konsepti, joka on hyödynnettävissä laajasti pienissä ja keskisuurissa (pk) yrityksissä.

Pk-yritysten nykytilan kartoituksessa selvisi, että suunnittelussa käytetään usein moderneja ohjelmistoja ja digitaalista tuotetietoa hyödynnetään osittain suoraan valmistuksessa. Tuotetiedon ja tuotteen elinkaaren hallinnan järjestelmien (PLM-järjestelmät) hyödyntäminen on kuitenkin vielä vähäistä. Yritykset ovat kiinnostuneita digitaalisesta tuoteprosessista, mutta niillä ei ole siitä riittävästi tietoa eivätkä ne pysty panostamaan tarvittaviin selvityksiin.

Pk-yrityksille suunnatussa digitaalisen tuoteprosessin konseptissa tuotteen elinkaari jaettiin neljään päävaiheeseen: tuotekehitys, valmistus, jälkimarkkinointi ja käytöstä poisto. Konseptin perusta on yrityksiä yhdistävä tietoverkko ja PLM-järjestelmä, jolla hallitaan niin tuotekehitysprosesseja, tuotetietoa kuin alihankintaakin. Tuotekehitysprosessia pilotoitiin tuotteilla, joiden konseptin kehittämisessä hyödynnettiin virtuaalisuunnittelua. Konseptin käyttöä pk-yrityksissä saattavat rajata virtuaalisuunnittelun sekä tuotetiedon hallintajärjestelmän käyttöönoton ongelmat.

Jari M. Ahola, Jani Hovila, Eero Karhunen, Kalervo Nevala, Timo Schäfer & Tom Nevala. Digital product process of a multitechnological small series product [Moniteknisen piensarjatuotteen digitaalinen tuoteprosessi]. Espoo 2011. VTT Tiedotteita – Research Notes 2592. 121 p. + app. 37 p.

Avainsanat product life-cycle management, product development, concurrent design, virtual design, simulation

Abstract

The fast development of technology during past decades has enabled to think product development and productization in a new way in which the product process covers the whole product life-cycle from marketing research to recycling. The aim of the MoniDigi-project was to develop a modern virtual design environment for multitechnological products and a concept for digital product process (DPP) which can be utilized widely in small and medium sized enterprises (SMEs).

The interview study about the present state of SMEs revealed that modern design softwares are used commonly and digital product data is exploited in manufacturing partially. However, the product data management and product life-cycle management (PLM) systems are used very little and e-mail is the most common way to transfer data which sets challenge for version management and data security. According to the surveys the SMEs are interested in DPP but they do not have sufficiently information and they cannot invest on required investigations.

In the developed DPP concept the product life-cycle was divided in four main phases: product development, manufacturing, after-sales and dismantling. The concept is based on a data network connecting companies and a PLM system which is used to manage product development processes, product data and subcontracting. The product development process was verified with real product cases which were developed using virtual design tools. The commissioning of virtual design and product data management systems may delimit implementation of the DPP concept in SMEs.

Alkusanat

Tämä tutkimusraportti liittyy Tekesin Digitaalinen tuoteprosessi -ohjelman projektiin ”Moniteknisen tuotteen digitaalisen tuoteprosessin kehittäminen tehokkaaseen piensarjatuotteiden suunnitteluun ja tuotantoon (MoniDigi)”. Raportti on suunnattu pk-teollisuudelle ja pyrkii toimimaan ikkunana MoniDigi-projektissa tehtyyn työhön.

Raportin ovat kirjoittaneet Jari M Ahola (VTT), Jani Hovila, Eero Karhunen, Kalervo Nevala ja Timo Schäfer (Oulun yliopisto). Liitteen A on kirjoittanut Tom Nevala (Oulun yliopisto).

MoniDigi-projektia ovat rahoittaneet TEKES, Oulun yliopisto, VTT sekä seuraavat yritykset:

Normet Oy
Erax Engineering Oy
Blastman Robotics Ltd
Insinööritoimisto Polartek Oy
Ocotec Oy
Girafe Oy
Sah-Ko Oy
IWS International Oy
Wisetime Oy
Terrasolid Oy
Winwind Oy
SK-Monitoimialat Oy/SK-Lift.

Kiitämme kaikkia projektiin osallistuneita erinomaisesta yhteistyöstä ja saadusta palautteesta.

Oulu 6.7.2011

Tekijät

Sisällysluettelo

Tiivistelmä	3
Abstract	4
Alkusanat	5
Symboliluettelo	8
1. Johdanto	10
2. Tuoteprosessi	13
2.1 Asiakastarve	14
2.2 Konstruktiosuorituksen määrittely	17
2.3 Tuoterakenne	18
2.4 Konseptien luonti ja luonnostelu	19
2.5 Tuotteen suunnittelu, kehittäminen ja viimeistely	19
2.6 Tuotanto ja jälkimarkkinointi (ylläpito)	20
2.7 Myynti ja markkinointi	21
3. Simulointipohjaisen suunnittelun tutkimus Suomessa	23
3.1 Liikkuva työkalu (TYÖ) 1993–1998	24
3.2 Tuotekehityksen tehostaminen valmistavassa teollisuudessa (RAPID) 1996–1999	25
3.3 Huomisen koneet ja järjestelmät (SMART) 1997–2000	26
3.4 Koneenrakennuksen teknologiaohjelma (MASINA) 2002–2007	28
3.5 Mallinnus ja simulointi (MASI) 2005–2010	31
3.6 Digitaalinen tuoteprosessi (DTP) 2008–2012	33
4. Nykytila ja kehittämismahdollisuudet pk-yrityksissä	35
4.1 Tuotesuunnittelu	35
4.2 Tuotetieto ja tuotteen elinkaaren hallinta	36
4.3 Digitaalinen osavalmistus ja tuotanto	37
4.4 Haastattelututkimukset	38
4.5 Kyselyn tulokset	39
5. Tuotetiedon hallinta	43
5.1 Tuotetieto	43
5.2 Tuotetiedon hallinta	43
5.3 Tuoteprosessiin liittyvät tietojärjestelmät	44
5.4 Tuotetiedon hallintajärjestelmän valintaprosessi	49
6. Digitaalinen tuoteprosessi – uusi konsepti	56
6.1 Tuotekehitys	59
6.1.1 Markkinaselvitys ja projektisuunnitelma	62
6.1.2 Vaatimusmäärittely	66
6.1.3 Turvallisuusprosessi	74
6.1.4 Virtuaalisuunnittelu	77

6.1.5	Testaus ja verifiointi.....	94
6.2	Tuotannon suunnittelu ja valmistus.....	97
6.3	Jälkimarkkinointi.....	98
6.4	Käytöstä poisto ja kierrätys.....	98
6.5	Tarjouspyynnöt ja komponenttien valinta.....	99
6.6	Tyyppihyväksyntä.....	103
6.7	Dokumentointi.....	104
6.8	PLM-järjestelmän rajapinnat muihin järjestelmiin.....	106
7.	Yhteenveto.....	112
	Lähdeluettelo.....	116

Liitteet

Liite A: Tutkimus myyntikonfiguraattorin kehittamisestä ENOVIA V6 -tuotetiedonhallinta-järjestelmään

Liite B: Matlab Simhydraulics -mallinnusesimerkki

Symboliluettelo

CAD	Computer Aided Design, tietokoneavusteinen suunnittelu
CAM	Computer Aided Manufacturing, tietokoneavusteinen valmistus
CAP	Computer Aided Planning, tietokoneavusteinen suunnittelemine (esim. aikataulutus)
CRM	Customer Relationship Management, asiakkuuksien hallinta
DH	Denavit-Hartenberg parametrit, määrittelevät robotin nivelkoor- dinaatistot
DTP	Digitaalinen tuoteprosessi
EAI	Enterprise Application Integration
EBOM	Engineering Bill Of Materials, suunnittelun tuoterakenne
EDM	Engineering Data Management
ERP	Enterprise Resource Planning, yritysten toiminnanohjausjärjestelmä
FEM	Finite Element Method, elementtimenetelmä
HIL	Hardware-in-the-Loop, tuoteprototyyppi, jossa osa tuotteen moduu- leista on reaalisia ja osa virtuaalisia
IEEE	Institute of Electrical and Electronics Engineers, kansainvälinen tekniikan alan järjestö
IP	Internet Protocol, paketteihin perustuva tietoliikenneprotokolla
MBOM	Manufacturing Bill of Materials, valmistuksen tuoterakenne
MBS	Multi Body System, monikappalesysteemi
MoniDigi	Moniteknisen tuotteen digitaalisen tuoteprosessin kehittäminen tehokkaaseen piensarjatuotteiden suunnitteluun ja tuotantoon -projekti
PDM	Product Data Management, tuotetiedonhallinta
Pk-yritys	Pieni ja keskisuuri yritys
PLM	Product Lifecycle Management, tuotteen elinkaaren hallinta

QFD	Quality Function Deployment
RFQ	Request for Quote, tarjouspyyntö
SA	Structured Analysis, rakenteellinen analyysi
SCM	Supply Chain Management, toimitusketjun hallinta
SIL	Software-in-the-Loop, ohjelmiston simulointi mallinnetussa virtuaalisessa ohjaimessa
STL	StereoLithography, 3D-mallitiedostoformaatti
TCP	Transmission Control Protocol, tietoliikenneprotokolla
VRML	Virtual Reality Modeling Language, virtuaalitodellisuuden kuvauskieli

1. Johdanto

Viime vuosikymmenten nopea teknologinen kehitys on tarjonnut mahdollisuuksia ajatella tuotekehitystä ja tuotteistamista uudella, modernilla tavalla. Suunnitteluohjelmistojen ja tietokoneiden kehittyminen ja yleistyminen sekä monet muutokset esimerkiksi ohjaustekniikassa, tuotannonohjauksessa ja materiaali-tekniikassa ovat muuttaneet merkittävästi ajattelutapaa. Uusia suunnitteluohjelmistoja pystytään perinteisen tuotedokumentaation lisäksi hyödyntämään tehokkaasti tuotteiden toiminnan simuloinnissa, tuotannon suunnittelussa ja tuotannossa.

Samaan aikaan tuotevalmistajille on syntynyt tarve siirtyä teknologiapainotteisesta ajattelutavasta yhä enemmän asiakastarvepohjaisen tuotteistamisen suuntaan. Internet on helpottanut informaation välitystä, mikä on merkinnyt yhä läheisempiä kontakteja asiakkaisiin ja kilpailijoihin. Tämä on aiheuttanut sekä uudenlaisen kilpailutilanteen että antanut aivan uusia mahdollisuuksia tuotteiden markkinoille saattamisessa. Esimerkiksi tuotevalmistajasta kaukanakin sijaitseva asiakas voi spesifioida ja tilata koneen, laitteen tai komponentin toimittajan vuorovaikutteisten www-sivujen kautta, jos tuotteen valmistaja tällaiseen palveluun on panostanut. Myös tietoverkottunut rinnakkainen tuotesuunnittelu on yleistynyt suunnitteluohjelmistojen verkkolisenssien myötä.

Modernin ajattelutavan mukaan tuoteprosessi kattaa tuotteen koko elinkaaren asiakastarvekartoituksesta tuotteen kierrätykseen tai jälleenkäyttöön. Tavoitteena on luoda kilpaileviin tuotteisiin nähden sellaisia moniteknisiä innovatiivisia tuotteita, jotka sisältävät merkittävästi parempia ratkaisuja sekä tuotteissa että valmistus- ja tuotantotekniikassa. Tämä merkitsee tuotekehityksen, suunnittelun sekä valmistuksen ja tuotannon osaajien yhteisprojekteja, joiden tavoitteena on suunnitella tuotteita moderneilla virtuaalisuunnittelun työkaluilla sekä hyödyntää pitkälle vietyä suunnittelutietoa suoraan automatisoidussa tuotannossa.

Tutkimusprojektin ”Moniteknisen tuotteen digitaalisen tuoteprosessin kehittäminen tehokkaaseen piensarjatuotteiden suunnitteluun ja tuotantoon (MoniDi-gi)” tavoitteena on kehittää työkonoiden ja erikoisajoneuvojen puomiratkaisuja valmistaville yrityksille moderneilla virtuaalisuunnittelun työkaluilla varustettu moniteknisten tuotteiden kehitysympäristö, digitaalisen tuoteprosessin konsepti sekä verkottunut toimintamalli, joka on sovellettavissa ja hyödynnettävissä laajasti pienissä ja keskisuurissa yrityksissä. Moniteknisessä tuotteessa tämä merkitsee tuotesuunnittelun osalta perinteisten koneenosien suunnittelun ja valmistuksen lisäksi mm. elektroniikan, ohjauksen, säädön ja ohjelmistojen suunnittelua sekä toiminnan verifiointia virtuaaliympäristössä.

Moniteknisen tuotteen kehitysympäristön ja digitaalisen tuoteprosessin kehittämisen tavoitteena on tehostaa pienten ja keskisuurten yritysten tuoteprosessia seuraavasti:

- Lyhentää moniteknisten tuotteiden tuotekehitysaikaa sekä nopeuttaa tuoteparannuksien ja uusien tuotteiden ominaisuuksien arviointia, alentaa tuotesuunnittelun kustannuksia ja varmistaa tuotteen parempi laatu.
- Tehostaa yritysten tuoteprosessia koko tuotteen osalta (sisältäen mekaaniset osat, elektroniikan ja ohjelmiston) tuomalla esille mahdollisuudet hallita modernien työkalujen avulla tuotetta ja sen valmistusta tuotespesifioinnista sekä käyttöympäristöstä lähtien koko tuotteen elinkaaren ajan.
- Lyhentää moniteknisten tuotteiden elektroniikka- ja ohjelmisto-osuuden tuotekehitysaikaa sekä alentaa tuotesuunnittelun kustannuksia.
- Tehostaa ja laajentaa työkonoiden ja erikoisajoneuvoja suunnittelevien sekä niihin automatisoituja puomiratkaisuja valmistavien yritysten ja alihankkijoiden verkostoitumista.
- Lisätä pk-yritysten tietämystä tuotetiedon ja tuotteen elinkaaren hallinta-ohjelmien ominaisuuksista sekä madaltaa kynnystä ohjelmistojen hankintaan ja käyttöönottoon.
- Mahdollistaa pk-yrityksille uusien, entistä laajempien tuote- ja palvelukonseptien kehittäminen ja tarjoaminen.

Moniteknisen tuotteen digitaalisen tuoteprosessin kehittämisen tavoitteena on linkittää tuotteen suunnitteluprosessiin tuotetiedon hallinta, digitaalinen valmistus sekä tuotteen elinkaaren hallinta. Tutkimusprojektin tavoitteena on yrityksistä valittujen tuotepilottien avulla testata ja verifioida projektissa kehitettävän

1. Johdanto

digitaalisen tuoteprosessin konsepti ja luoda yritysten liiketoiminnan lähtökohdista verkottunut toimintamalli. Tuotepilotit on valittu siten, että ne täyttävät moniteknisen tuotteen vaatimukset ja samalla rajaavat ja konkretisoivat sinänsä laajan tutkimuksen aihealueen. Projektin yhtenä osatavoitteena on edistää eri toimialojen yritysten keskinäistä verkostoitumista ja nostaa moniteknisten pientuotteiden suunnitteluosaaminen sekä digitaalinen tuote- ja tuotantoosaaminen korkealle kansalliselle ja kansainväliselle tasolle.

MoniDigi-projektissa kehitetyssä digitaalisen tuoteprosessin konseptissa on – nimensä mukaisesti – esitetty moniteknisten tuotteiden kehittämisen ja verkostomaisen suunnittelun yleiset periaatteet, kun hyödynnetään nykyaikaisia tietojärjestelmiä ja simulointiohjelmistoja. Digitaalisen tuoteprosessin käytännön toteutuksen määrittävät ensisijaisesti yrityksen resurssit sekä kehitettävät tuotteet. Digitaalinen tuoteprosessi on tässä julkaisussa esitetyssä muodossa todennäköisesti liian työläs noudatettavaksi pienille yrityksille, joiden tulisikin keskittyä omalta kannaltaan olennaisiin prosessivaiheisiin. Jokaisella yrityksellä on kuitenkin hyvät mahdollisuudet noudattaa konseptissa esitettyjä periaatteita, koska merkittävin (eikä välttämättä kallein) edellytys konseptin hyödyntämiselle ovat henkilöstön ajattelu- ja toimintatavat.

2. Tuoteprosessi

Yleisesti pienten ja keskisuurten teollisuusyritysten ydinprosesseja ovat tuote- ja tilaus-toimitusprosessit. Tuoteprosessi, jonka tarkoituksena on tuoda markkinoille uusi tuote, jakautuu kahteen vaiheeseen: tuotekehitys- tai konstruktioprosessiin sekä ylläpito- ja kehitysprosessiin. Tilaus-toimitusprosessissa keskitytään fyysisen tuotteen hallintaan koko toimintaketjun läpi. (Sääksvuori & Immonen 2008.)

Nämä kaksi prosessia ovat yleisesti integroituneet toisiinsa. Tuotteen elinkaaren alussa informaatio suunnitelluista komponenteista ja osista toimitetaan tuotesuunnittelusta ostoon. Käytettävissä oleva tuoterakenne vaihtoehtoiseen välitetään hyvin aikaisessa vaiheessa suunnittelusta myyntiin. Kun konstruktioprosessi saadaan viimeistelyä, tuotedokumentaatio toimitetaan tuotantoon ja ylläpitoon. Tämän jälkeen tuotteen muutokset hallitaan ylläpidossa, joka toimittaa tuotteen ja varaosiin liittyvät muutokset tuotantoon.

Tuotesuunnittelun lähtökohtana on yleensä kaksi vaihtoehtoa: markkinoiden vaatimukset uudesta tuotteesta tai uusi tuoteidea, jollaista ei vielä ole markkinoilla. Noin 80 % uusista tuotteista on markkinavetoisia. Markkinavetoisten tuotteiden pitää sisältää menestyäkseen myös viimeisintä teknologiaa, ellei niiden tarkoituksena ole olla luksustuotteita. Yritykset haluavat kuitenkin usein kehittää tuotteita uusille markkinoille. Useimmiten kyseiset tuotteet perustuvat uuden teknologian hyödyntämiseen, mikä vaatii yleensä runsaasti investointeja ja mahdollisesti vuosien tutkimuksen ja kehityksen. Vaikka tuoteideat ovat innovatiivisia ja kehityskelpoisia, niin valmista tuotetta ajatellen ne saattavat olla käyttökelvottomia, ellei niitä saada kohdennettua nykyisille markkinoille tai jos niitä varten ei saada luotua uusia markkinoita. Tuoteprosessi vaihtelee tuotteiden ja eri teollisuudenalojen välillä, mutta yleiset prosessivaiheet täytyy suorittaa kaikissa projekteissa. (Ullman 2009.)

2.1 Asiakatarve

Useimmissa yrityksissä myynti- ja markkinointiosastoilla on pitkä lista mahdollisista uusista tuotteista tai tuoteparannuksista, jotka haluttaisiin toteuttaa. Lisäksi näissä uusissa tuotteissa tulisi olla vähintään samat ellei paremmat toiminnallisuudet kuin kilpailijoilla. Jos myynti ja markkinointi saisi päättää, uusia tuotteita uusilla ominaisuuksilla tulisi jatkuvasti, jotta kaikkien potentiaalisten asiakkaiden vaatimukset tulisi täytettyä. (Ullman 2009.)

Tuotteen tarve perustuu joko markkinoiden vaatimukseen uudesta tuotteesta tai uuden teknologian tarpeesta vanhan tuotteen uudistamiseksi. Uudelleen suunnittelulla voidaan myös korjata jo olemassa olevan tuotteen puutteita, alentaa tuotteen kustannuksia, parantaa valmistettavuutta tai vastata markkinoilta tulleeseen muutosvaatimukseen. Usein halu muutokseen on asiakasvetoinen eli asiakas haluaa tuotteen olevan halvempi, sisältävän uusia ominaisuuksia tai kestävä pidempään. Suurin osa suunnittelutyöstä on yleensä uudelleensuunnittelua, joka alkaa olemassa olevasta tuotteesta. (Ullman 2009.)

Vaatimusten keräämiseen, analysointiin, valintaan, dokumentointiin, vertaamiseen ja hallintaan on olemassa useita menetelmiä. Useimmat niistä on tarkoitettu ohjelmistokehitykseen, mutta vaatimusten hallintamenetelmiä on yhä enenevässä määrin alettu käyttää tuotesuunnittelussa, kun vaatimusten hallinnan merkitys on havaittu. (Baxter ym. 2008.)

Useimmat suunnittelumetodit sisältävät vaiheita, joiden avulla asiakkaan vaatimukset saadaan määriteltyä tuotteeseen. Pahl ja Beitz (1992) edustaa systemaattista metodia, jossa tuote saadaan määriteltyä lähtöinformaation systemaattisella analyysillä ja synteessillä. Kirjoittajien mukaan tehtävänasettelun selvitys tarkoittaa informaation hankintaa sekä tehtävälle asetetuista vaatimuksista että pysyvistä yleisistä reunaehdoista ja niiden merkityksestä. Tämä tehtävä johtaa vaatimuslistan laatimiseen, jossa otetaan huomioon konstruktiivisen kehittämisen tarve ja sen mukaisesti suunniteltavat seuraavat työaskeleet (kuva 1). (Pahl & Beitz 1992.)

Tämän metodin muunnelmana voidaan pitää intuitiivista metodia, jossa perehtymistä syvennetään tutkimalla tehtäväkenttää riippumattomissa yksityiskohdissa. Näiden yksityiskohtien määrittelemiseksi tuotteen vaatimukset rikkianalysoidaan eli puretaan osakokonaisuuksiksi, jolloin oleellisten vaatimusten havaitseminen helpottuu. Tällöin tuotteen problematiikkaa lähestytään uudella tavalla, toisin kuin systemaattisessa metodiikassa (Tuomaala 1995).

Quality Function Deployment (QFD) -metodissa asiakkaan tarpeet tunnistetaan ja määritellään sekä muunnetaan teknisiksi vaatimuksiksi, joita mitataan ja verrataan siihen, kuinka hyvin asiakkaan vaatimukset täytetään. QFD:n tarkoituksena on parantaa tuotteen laatua, sillä huono tuotemäärittely on syynä viiveisiin useimmissa tapauksissa konstruktioprosesissa. Ullmanin (2009) mukaan QFD:n avulla voidaan luoda mitattavissa olevat suunnittelutavoitteet ja havaita tuotteen määrittelyn puutteita.

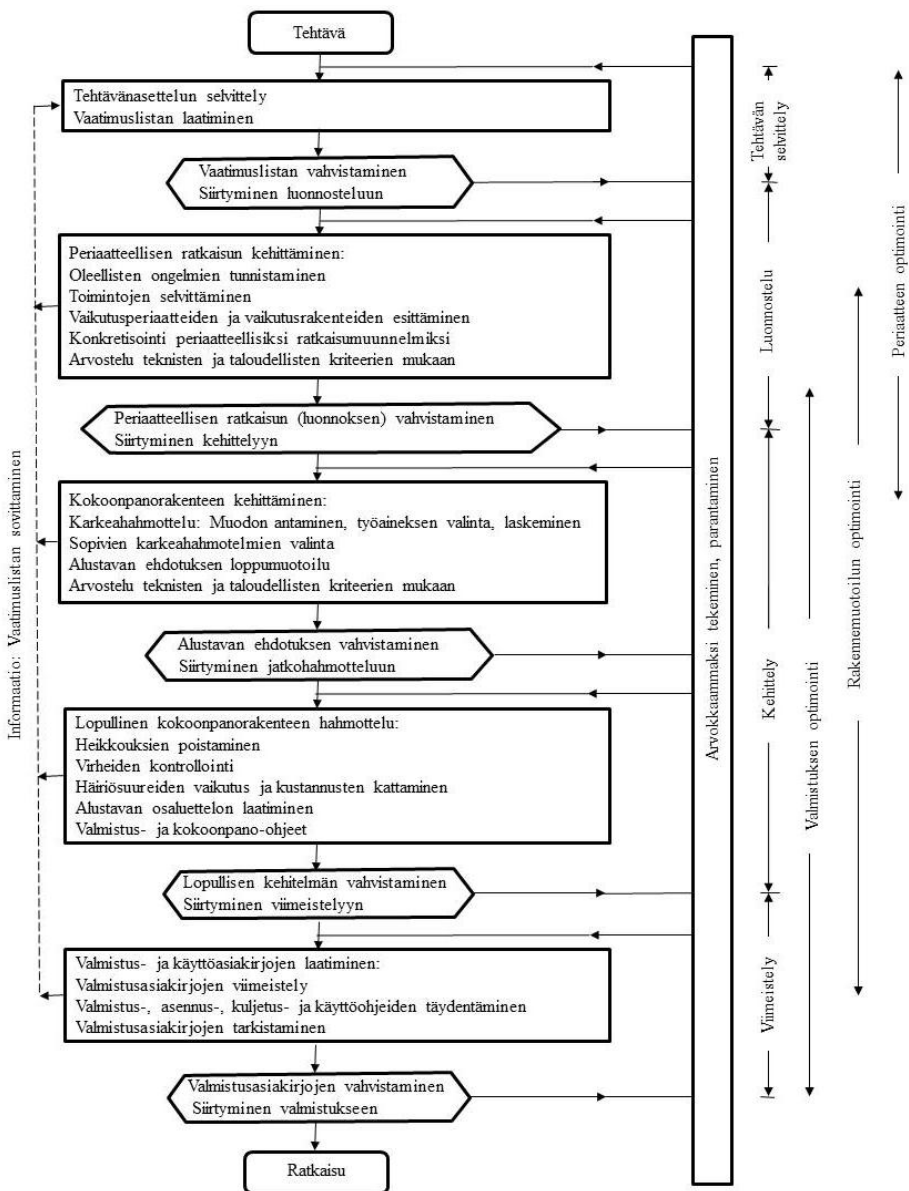
Suurin osa tuotteen vaatimuksista havaitaan konstruktioprosessin aikana, eivätkä ne ole asiakaslähtöisiä. Konstruktioprosessin aikana suunnittelijat asettavat tuotteelle yksityiskohtaisempia teknisiä vaatimuksia. Siksi vaatimusten hallinta on yhä merkittävämpi osa konstruktioprosessia. Pidemmälle vietyinä vaatimusten hallinnan avulla voidaan tunnistaa tuotevaatimusten ja ratkaisuperiaatteiden väliset suhteet, jolloin tätä tietoa voidaan käyttää uudelleen seuraavissa tuotesukupolvissa. Toisaalta lisätietämys tuotannosta, käytöstä, testauksesta ja huollosta voidaan myös linkittää tuoteratkaisuun. (Baxter ym. 2008.)

Yritykselle on hyötyä tuotekehitystiedosta kolmesta syystä:

1. Kun tuotetiedon ja tuotteen kehityshistoria tunnetaan hyvin, voidaan tuottaa laadukasta tuotetietoa.
2. Seuraavan sukupolven tuotteet ovat merkittävästi kehittyneempiä edeltäjiinsä verrattuna.
3. Tuotetiedon uudelleenkäyttö mahdollistaa sen, että kilpailukykyä edistävälle innovaatioille jää enemmän aikaa.

Tuotekehitykseen käytetty aika, tuotteen laadukkuus ja asiakkaalle tuotettu arvo ovat kaikki asioita, joita tuotetiedon hallinta voi tehostaa. Varmistamalla, että oikeat vaatimukset täytetään, asiakastytyväisyyttä voidaan parantaa ja tuotekehitykseen käytettyä aikaa lyhentää. Tuotteen laatu ja havaittu arvo ovat korkeammat, jos asiakkaan vaatimukset ymmärretään paremmin ja vaatimukset on kohdennettu systemaattisesti. (Lee ym. 2008)

2. Tuoteprosessi



Kuva 1. Konstruoinen työaskeleet (Pahl & Beitz 1992).

2.2 Konstruktioprosessin määrittely

Konstruktioprosessin tehokkuutta voidaan arvioida kolmella mittarilla: tuotteen kustannus, laatu ja toteuttamiseen käytetty aika (Sääksvuori & Immonen 2008). Ehkäpä tästä johtuen perinteisesti konstruktioprosessia määriteltäessä keskitytään aikataulukseen, kustannusarvioon ja henkilöresursseihin. Suunnitteluprosessi on yksi tärkeimmistä prosesseista tuotekehityksessä. Prosessin tuloksena saadaan toimintamalli (kaavio) siitä, miten konstruktioprosessin aktiviteetit on ajoitettu. Lopuksi suunnitteluprosessilla saadaan generoitua toimintamalli tuotekehityksessä tarvittavan tiedon luomiseksi ja jakamiseksi oikeille henkilöille oikeaan aikaan. Informaatio sisältää tuotevaatimukset, luonnokset, systeemin toiminta-kaaviot, osien piirustukset, kokoonpanopiirustukset, materiaalivalinnat sekä muita tuotesuunnitteluun liittyviä dokumentteja. Yleensä yrityksillä on olemassa yleinen konstruktioprosessi, joka räätälöidään tuotekohtaiseksi. (Ullman 2009.)

Suunnitelmat vaihtelevat yksinkertaisesta yhden sivun suunnitelmasta useisiin, satojen sivujen yksityiskohtaisiin vaatimuksiin uuden tuotteen kehittämiseksi. Konstruktioprosessin kompleksisuus voidaan jakaa neljän ryhmään (Ullman 2009):

- Muunnelma tuotteesta, jolloin muutamia tuotteen parametreja muutetaan. Tässä tapauksessa projektiin liittyy vain vähän konstruointia. Joissain tapauksissa tuote on jo alun perin suunniteltu parametrisesti, jolloin muutoksen voi toteuttaa jopa myynti.
- Tuotteen parantaminen, jolloin joitain osia tuotteesta pitää suunnitella uudelleen. Uudelleen suunnittelun tarve voi johtua useista seikoista:
 - Asiakas haluaa uusia ominaisuuksia tuotteeseen tai parempaa suorituskykyä tuotteelle.
 - Alihankkija ei enää pysty toimittamaan käytössä olevaa materiaalia tai komponenttia (tai niihin on tullut muutoksia).
 - Tuotanto, kokoonpano tai ylläpito on havainnut laatuun tai kustannukseen liittyvän parannuksen.
 - Uusi teknologia tai parannukset vanhaan teknologiaan mahdollistavat parannuksen suorituskykyyn tai säästön valmistuskustannuksiin. Tämän tyyppinen kehityshanke saattaa olla luonteeltaan pieni tai voi vaatia täydellisen konstruktioprojektin.

2. Tuoteprosessi

- Uuden tuotteen suunnittelu yksittäiseksi tai piensarjatuotteeksi: Tuotetta valmistetaan vain yksi tai muutamia kappaleita. Piensarjatuotteen ominaispiirre on pienempi valikoima valmistusmenetelmiä, kuten räätälöidyt mikropiirit, metallin prässäys ja muovin puristus, joka vaatii riittävän suuren sarjakoon työkalukustannuksien kuolettamiseksi. Tämä rajoittaa suunniteltavia komponentteja, ja yleensä pyritäänkin käyttämään standardiosia. Usein ensimmäinen tuote valmistetaan sekä prototyypiksi että lopulliseksi asiakkaalle toimitettavaksi tuotteeksi.
- Uuden tuotteen suunnittelu massatuotantoon: Uuden tuotteen suunnittelu-prosessi vaihtelee helposta monimutkaiseen (lelun suunnittelu verrattuna autosuunnitteluun). Tuotannon ja kokoonpanon suunnittelu muodostavat merkittävän osan suunnitteluprosessista.

2.3 Tuoterakenne

Tuoterakenteen avulla tuotteeseen tulevat osat, komponentit, kokoonpanot ja ohjelmistot saadaan koottua rakenteeksi, jolla tuotekokonaisuutta voidaan hallita. Tuoterakenteella on vaikutus sekä konstruktioprosessiin että tilaus-toimitusketjuun. Rakenteessa valitut konfigurointimallit mahdollistavat tuotteen hallinnan ja konfiguroitavuuden tuoteprosessin eri vaiheissa. Jos tuotteen rakenne on epäselvä eivätkä tuotteen konfigurointisäännöt noudata tiettyä logiikkaa, tuotetiedon hallinta vaikeutuu moninkertaisesti. Tuoterakenteessa pitää ottaa kantaa esimerkiksi tuotteen konfiguroinnin sääntöihin ja varianttien määrään, jotta lopputuote voidaan hallita systemaattisesti tietojärjestelmissä.

Näiden sääntöjen puuttuminen ja päätöksien tekemättä jättäminen tuntuu olevan tyypillistä, ja useissa yrityksissä ihmetelläänkin, miksi myyntimiehet myyvät tuotteita, joita yritys ei valmista. Tekemättömien päätösten vaikutuksia koko tilaus-toimitus ketjuun ei ole tunnistettu, eikä täysin ymmärretä sitä, millaisia ongelmia huonosti määritelty tuote aiheuttaa organisaation eri prosesseissa. Tämä korostuu erityisesti, kun perinteiset valmistavan teollisuuden yritykset ovat siirtymässä kohti palveluliiketoimintaa. Kyvyttömyys määritellä aineettomat tuotteet (kuten tieto ja huoltopalvelut, ohjelmistot) on johtanut siihen, ettei asiakkaita osata laskuttaa oikein eikä tuotteesta pystytä perimään oikeaa hintaa, kun tuotteen kustannusrakennetta ei pystytä määrittelemään tarkasti. (Kropsu-Vehkaperä 2010.)

2.4 Konseptien luonti ja luonnostelu

Luonnostelu on se osa konstruointia, jossa määritetään vaikutusrakenteeseen perustuva periaatteellinen ratkaisu. Siihen päästään oleellisten ongelmien abstrahoinnin ja toimintarakenteiden laatimisen avulla sekä selvittämällä sopivat vaikutusperiaatteet ja yhdistelemällä niitä vaikutusrakenteeksi. (Pahl & Beitz 1992.)

Monissa tapauksissa vaikutusrakennetta voidaan kuitenkin arvostella vasta sen jälkeen, kun se saa konkreettisemmän muodon. Tämä edellyttää täsmällisempää kuvaa tarvittavista materiaaleista ja komponenteista sekä useimmiten alustavaa peruspiirustusta (mallintamista) ja teknisten mahdollisuuksien huomioon ottamista. Yleensä vasta näin saadaan arvosteltavaksi kelpaava ratkaisuperiaate, jossa on otettu huomioon tavoitteenasettelu ja rajoittavat ehdot. Useat ratkaisumuunnelmät ovat tehtävästä riippuen mahdollisia.

Periaatteellinen ratkaisu voidaan esittää monella tavalla. Joissain tapauksissa vapaakätinen luonnos riittää tai karkeamittakaavainen piirustus (malli) voi olla tarpeen. Käytettäessä kiinteitä rakenne-elementtejä voi riittää pelkkä toimintarakenteen lohkokaavio, kytkentäkaava tai kulkukaavio.

Luonnosteluvaihe voidaan jakaa seuraaviin työvaiheisiin: informaation määrittely, luominen, arvostelu ja päättäminen. Nämä askeleet pitää käydä läpi sen vuoksi, että jo ennakolta varmistettaisiin parhaan ratkaisun löytyminen, sillä luonnoksen heikkouksia ei voida poistaa seuraavissa vaiheissa. Tämä toteamus ei kumoa sitä tosiasiaa, että tarkoituksenmukaisiksi osoittautuneiden periaatteiden ja niiden yhdistelmien mukaisten ratkaisuiden vaikeudet aiheutuvat tavallisimmin yksityiskohdista. (Pahl & Beitz 1992.)

2.5 Tuotteen suunnittelu, kehittäminen ja viimeistely

Kehittäminen on se osa konstruointia, jossa suunnitellaan tuotteen rakenne täydellisesti ja yksikäsitteisesti teknisten ja taloudellisten näkökohtien mukaan. Monessa tapauksessa joudutaan tekemään rinnakkain useampia alustavia muunnelmia. Kehittelyvaiheen asianmukaisen läpikäymisen jälkeen päädytään teknisen kokonpanorakenteen alustavasta ehdotuksesta arvosteluun, jolloin voidaan tehdä päätös lopullisen kokonaiskehittelyn rakennemuotoilusta tai palaamisesta kehittäelyyn. Lopullisessa kokonaiskehittelyssä on jo tarkistettu toiminnot, kestävyys, tilankäytön sopivuus jne., minkä ohessa viimeistään on osoitettava, että kustannuksia koskevat vaatimukset voidaan täyttää. (Pahl & Beitz 1992.)

2. Tuoteprosessi

Viimeistely on se osa konstruointia, jossa teknisen rakennelman kokoonpanorakennetta täydennetään ja se saadaan lopulliseen muotoonsa kaikkien yksittäisosien mitoitusta ja pinnanlaatua koskevilla määräyksillä, työainesten määrityksillä sekä valmistusmahdollisuuksien ja lopullisten kustannuksien tarkastuksilla. Viimeistelyssä laaditaan sitovat piirustukset ja muut asiakirjat suunnitelman aineellista toteuttamista varten. Konstruoinnin päävaiheita ei voida rajata tarkasti. Esimerkiksi luonnoksesta päätettäessä tarvitaan mittakaavaista tutkielmaa tavoitteena olevasta rakennemuotoilusta. Toisaalta kehittälyvaiheen alussa kokoonpanorakennetta voidaan hahmotella karkeasti vasta alustavien mittakaavaan tehtyjen luonnosten jälkeen. Edelleen sellaiset optimoinnit, jotka rajoittuvat yksittäisiin osa-alueisiin eivätkä aiheuta laajemmalti palautevaikutusta, voivat siirtyä kehittälyvaiheesta viimeistelyvaiheeseen. Tällaiset poikkeamat ovat tehtävänasettelusta ja tuotelajista riippuen hyvin mahdollisia ilman ristiriitaa esitellyn yleisen menetelmätavan kanssa. (Pahl & Beitz 1992.)

Monessa tapauksessa joudutaan tekemään prototyyppejä jo luonnosteluvaiheessa, varsinkin silloin, kun on selvittävä perustavaa laatua olevia kysymyksiä. Tätä käytäntöä noudatetaan hienomekaniikassa, elektroniikassa ja suursarjoja valmistavissa yrityksissä. Sen sijaan raskaiden koneiden prototyyppien rakentamisessa, mikäli ei ole kysymys detaljiongelmasta, tarvitaan usein täydellisiä viimeistelyvaiheesta saatavia tietoja, ennen kuin prototyypin rakentamiseen ja kokeiluun voidaan ryhtyä. (Pahl & Beitz 1992.)

2.6 Tuotanto ja jälkimarkkinointi (ylläpito)

PLM muodostaa yhteyden tuotannon ja tuotekehityksen välille. Muutosten hallintatyökalut mahdollistavat suunnittelijoille tuotannon informoinnin muutoksista osiin, komponentteihin, suunnitelmiin ja muutoksiin liittyvien uusien piirustuksien ja 3D-mallien toimittamiseen. Toisaalta tuotanto voi myös esittää muutoksia suunnitteluun niin, että tuotteen tuotannollisuus paranee. (Sääksvuori & Immonen 2008.)

PLM-järjestelmien käyttö tuotteen ylläpidossa on lisääntynyt voimakkaasti. Useat yritykset ovat rakentaneet kokonaan uuden liiketoiminnan jälkimarkkinoinnista, joka on noussut merkittäväksi liiketoiminnan osaksi. Esimerkiksi lentokoneiteollisuudessa yritysten kannattavuus ei enää perustu koneiden myyntiin vaan niiden kunnossapitoon (Lee ym. 2008). Toisaalta, kun tuotteet kehitetään nopeasti, uusia tuoteversioita tulee markkinoille jatkuvasti. Tämä asettaa suuren haasteen varaosamyynnille ja huoltopalveluille, varsinkin niissä yrityksissä,

jotka toimivat maailmanlaajuisilla markkinoilla. Dokumenttien, tuoterakenteen ja nimikkeiden hallinta on merkittävässä asemassa. Tieto tarvittavista varaosista, tuoteversioista ja toimitetuista tuotteista voidaan selvittää nopeasti ja ylläpitää helposti PLM-järjestelmässä. (Sääksvuori & Immonen 2008.)

2.7 Myynti ja markkinointi

PLM-järjestelmä soveltuu hyvin tilaaja-toimittaja prosessiin, sillä se mahdollistaa tuotteiden toimittamisen ja konfiguroinnin asiakkaan toiveiden mukaisesti. Myynnin, markkinoinnin ja valmistuksen tukena on nykyään erilaisia myynti- ja tuotekonfiguraattoreita. Näiden kahden konfiguraattorin ero voidaan määritellä lyhyesti niin, että myyntikonfiguraattoreita käytetään myynnin ja asiakkaan välisessä rajapinnassa ja tuotekonfiguraattori on puolestaan yrityksen sisäinen työkalu. Myyntikonfiguraattoreissa valinnat tehdään tuotteen ominaisuuksien perusteella, sillä asiakas on kiinnostunut ensisijaisesti niistä. Tuotekonfiguraattori taas määrittelee tuotteen rakenteen lähinnä valmistuksen näkökulmasta.

Nykypäivän monitekniset tuotteet sisältävät useita modulaarisia komponentteja, joita on muokattava asiakkaan tarpeiden mukaisesti. Yksi määritelmä konfiguroitavalle tuotteelle onkin, että se on muunneltavissa asiakaskohtaisesti (Peltonen ym. 2002). Moniteknisyyden, modulaarisuuden ja asiakkaiden erilaisten tarpeiden täyttämisen vaatii useita toisistaan poikkeavia tuotekonfiguraatioita. Tuotteeseen liitettävien moduulien ja komponenttien välille on tuotesuunnitteluvaiheessa määriteltävä ehtoja, joilla rajoitetaan sopimattomien komponenttien ja moduulien valintaa tuoterakenteeseen. Tuotekonfiguraatio on tarkka yksilöllinen listaus tuoterakenteen sisällöstä: se sisältää täydelliset komponenttiluettelot, sekä itse että ulkopuolella valmistettavat osat ja valmistusdokumentit (Peltonen ym. 2002). Valmistus- ja materiaalitietojen lisäksi se voi sisältää tietoja tuotteen kokoonpanosta, testauksesta ja pakkauksesta (Peltonen ym. 2002).

Myyntikonfiguraattori on tietotekninen sovellus, joka helpottaa myyntitapah- tumaa asiakkaan kanssa. Myyntikonfiguraattorin avulla asiakas voi joko itsenäisesti tai yhdessä myyjän kanssa määrittää valitsemansa tuotteen rakenteen, tai pikemmin sen ominaisuudet, ennalta määriteltyjen konfigurointiehtojen perusteella. Myyntikonfiguraattori tarjoaa asiakkaalle tehtyjen valintojen perusteella siihen sopivat muut vaihtoehtoiset ominaisuudet. Tietyt valinnat voivat esim. rajata toiset pois. Myyntikonfiguraattorin laadusta riippuen asiakkaalle voidaan ilmoittaa valintojen yhteydessä, miten ne vaikuttavat seuraaviin valintoihin. Olennaista on, että asiakkaalla ei tarvitse olla teknistä ymmärrystä siitä, miten

2. Tuoteprosessi

tehdyt valinnat vaikuttavat muihin mahdollisiin valintoihin. Asiakas voi siis aina valita vain toteutuskelpoisista vaihtoehtoista.

Myyntikonfiguraattorin olennainen piirre on välitön palaute asiakkaalle, eli asiakas näkee esimerkiksi valintojen pohjalta muodostuvan tuotteen hinnan ja toimitusajan. Mitä vuorovaikutteisempi myyntikonfiguraattori on, sitä enemmän asiakas ymmärtää valintojensa seurauksia. Myyntitapahtumasta tulee näin asiakkaalle myönteisempi kokemus. (Peltonen ym. 2002).

Myyntikonfiguraattorilla tehdyt valinnat siirretään joko manuaalisesti tai ne siirtyvät automaattisesti tuotekonfiguraattoriin, jonka avulla muodostetaan asiakkaan määrittelemän tuotteen yksilöllinen tuoterakenne. Tuotekonfiguraattorin ansiosta tuoterakenteen komponenttiluetteloiden, dokumentaatioiden ja spesifikaatioiden muodostaminen nopeutuu huomattavasti, koska tarvittava tieto on saatavissa helposti ja nopeasti. Tuloksena saadaan tuoteyksilö, joka täyttää asiakkaan vaatimukset, ja toisaalta myös toimittajan näkökulmasta tuote on looginen, toiminnallinen, mahdollinen ja tuoterakenteeltaan virheetön. Huolellinen myynti- ja tuotekonfiguraattoreiden rakentaminen ja muutosten hallinta estävät kalliiden virheiden syntymisen räätälöidyissä tuotteissa. (Sääksvuori & Immonen 2008).

3. Simulointipohjaisen suunnittelun tutkimus Suomessa

Ohjelmistosuunnitteluun on kehitetty ja sovellettu lukuisia erilaisia menetelmiä 1960-luvulta lähtien (Puolitaival 2008, Abrahamsson ym. 2002), mutta koneen-suunnittelijat ovat olleet huomattavasti konservatiivisempia pitäytyen 2000-luvulle asti pääosin 1970-luvulla kehitetyssä Pahl & Beitzin (1992) suunnittelu-prosessimallissa. Ohjelmistokehittäjien tavoitteena on ollut luoda kankeiden suunnitteluprosessien tilalle uusia, joustavampia menetelmiä, joiden avulla muuttuviin asiakasvaatimukseen voidaan reagoida paremmin (Puolitaival 2008, Naur & Randell 1968, Abrahamsson ym. 2002). Tarve ketterämmille suunnittelumenetelmille on kuitenkin lisääntynyt myös koneenrakennuksessa koneiden moniteknistymisen ja asiakaskohtaisen räätälöinnin myötä. Tietotekniikan ja simulointiohjelmistojen nopea kehittyminen parin viime vuosikymmenen aikana on luonut edellytykset joustavampien simulointipohjaisten suunnittelumenetelmien kehittämiseksi.

Tämä luku sisältää katsauksen tärkeimpiin TEKESin rahoittamiin teknologiaohjelmiin, joissa on tutkittu moniteknisten koneiden simulointipohjaista suunnittelua. TEKES on käynnistänyt 1990-luvun alusta lähtien useita teknologiaohjelmia, joiden osatavoitteena on ollut edistää mallinnuksen ja simuloinnin hyödyntämistä suomalaisessa teknologiateollisuudessa. Tällaisia tutkimusohjelmia ovat olleet Liikkuva työkone (TYÖ) 1993–1998, Tuotekehityksen tehostaminen valmistavassa teollisuudessa (RAPID) 1996–1999, Huomisen koneet ja järjestelmät (SMART) 1997–2000, Koneenrakennuksen teknologiaohjelma (MASINA) 2002–2007 sekä Mallinnus ja simulointi (MASI) 2005–2010. Tällä hetkellä yritysten valmiuksia simulointipohjaiseen suunnitteluun edistetään Digitaalinen tuoteprosessi -ohjelmassa (DTP), joka ajoittuu vuosille 2008–2012.

Teknologiaohjelmien vaikuttavuutta suomalaisessa teollisuudessa on arvioitu ohjelmien arviointiraporteissa. TYÖ-, SMART- ja RAPID-ohjelmien tiivistetty

3. Simulointipohjaisen suunnittelun tutkimus Suomessa

jälkiarviointi sekä MASINA-ohjelman väliarviointi ovat raportissa Kivikko (2004). SMART-ohjelman loppuraportin ja -arvioinnin ovat tehneet Kuuva ym. (2001). RAPID-ohjelmaa ovat arvioineet Eppinger & Synterä (2000) ja MASINA-ohjelmaa Syrjänen ym. (2008). MASI-ohjelman loppuarviointi ja DTP-ohjelman väliarviointi ovat raportissa Lehenkari ym. (2010).

Seuraavissa alaluvuissa esitellään tiivistetysti edellä mainittujen TEKESin teknologiaohjelmien tavoitteet ja tulokset simulointipohjaisen tuotesuunnittelun edistämisen näkökulmasta. Samalla arvioidaan teknologiaohjelmissa syntyneiden simulointipohjaisten suunnitteluprosessimallien hyödynnettävyyttä verkotuneessa ja rinnakkaisessa suunnittelussa. Arviointi tehdään seuraavilla kriteereillä:

1. kyky reagoida muutoksiin asiakastarpeissa ja markkinoissa
2. moniteknisuus
3. rinnakkainen suunnittelu
4. suunnittelutiedon hallinta.

Nämä arviointikriteerit valittiin, koska simuloinnille on määriteltävä selkeä rooli ja tavoitteet yritysten tuotekehitysprosessissa. Suunnitteluprosessin olisi lisäksi sovelluttava useita osajärjestelmiä sisältävien tuotteiden rinnakkaiseen suunnitteluun. Toisaalta moniteknisuus ja rinnakkainen suunnittelu edellyttävät ratkaisuja suunnittelutiedon hallintaan, koska suunnitteluun osallistuu yleensä eri alojen asiantuntijoita, joiden välillä tiedon olisi siirryttävä luotettavasti.

3.1 Liikkuva työkone (TYÖ) 1993–1998

Liikkuva työkone -teknologiaohjelma päätettiin käynnistää silloisen alan keskeisen toimijan Timberjack/Plustech Oy:n tekemän selvityksen perusteella alan merkityksestä ja kehittämistarpeista. TYÖ-ohjelman tavoitteena oli parantaa suomalaisen työkoneteollisuuden kilpailukykyä kehittämällä uusia teknologisia ratkaisuja koneiden energijärjestelmiin, ohjaukseen ja automaatioon sekä rakenteisiin ja materiaaleihin. Ohjelmassa suoritettut tutkimusprojektit olivat Liikkuvan työkoneen ympäristöhavainnointi, Luistonesto- ja voimansiirtojärjestelmä, Puomi 2000, Liikkuvien työkoneiden kevytrakenteet, Liikkuvien työkoneiden etäoperointi, Reaaliaikainen simulointi off-road-työkoneen kehittämisessä sekä Sähköinen voimansiirto liikkuvissa työkoneissa. Koska tutkimusprojektit olivat hyvin ongelmakeskeisiä, ne muodostuivat sisällöltään moniteknisiksi edellyttäen useiden osaamisalueiden integrointia. TYÖ-ohjelmassa haluttiinkin vah-

vistaa työkonealan verkostoitumista pk-yritykset mukaan lukien. (Kivikko 2004.)

Projektien tuloksena syntyi useita patenteihin johtaneita innovatiivisia tuote-parannuksia, ja moniteknisten tuotteiden suunnittelusta opittiin uutta. Vaikka ohjelma lisäsin yritysten teknologiaosaamista merkittävästi, ohjelmalla oli varsin vähän vaikutusta tuotekehitys- tai innovaatioprosesseihin – toisaalta ohjelmalle ei asetettu siihen tähtääviä tavoitteitakaan. TYÖ-ohjelmassa kuitenkin tunnistettiin prosessien kehittämistarpeet, koska ohjelmaan sisältynyt simulointiprojekti antoi kokemuksia virtuaalisuunnittelusta. Lisäksi huomattiin, kuinka hankalaa on organisoida moniteknisiä kehityshankkeita, ja ymmärrettiin, että älykkäiden ja moniteknisten tuotteiden suunnittelu edellyttää virtuaalisuunnittelun hyödyntämistä. (Kivikko 2004).

3.2 Tuotekehityksen tehostaminen valmistavassa teollisuudessa (RAPID) 1996–1999

RAPID-teknologiaohjelma jatkoi suomalaisen valmistavan teollisuuden kilpailukyvyyn nostamista TYÖ-ohjelmassa havaittujen kehittämistarpeiden pohjalta. RAPID-ohjelman tärkein tavoite oli tehostaa tuotekehitysprosesseja kokonaisvaltaisesti valmistavassa teollisuudessa. Tuotekehitysprosessien tehostamista tavoiteltiin kehittämällä liiketoimintaprosesseja, lisäämällä informaatiotekniikan hyödyntämistä, siirtämällä teknologiaa pk-yrityksiin sekä lisäämällä yhteistyötä yritysten, korkeakoulujen ja tutkimuslaitosten kesken. RAPID-ohjelmassa tutkimus kohdistettiin viidelle painopistealueelle (Eppinger & Synterä 2000, Kivikko 2004):

1. tuotekehitysprosessin tehostaminen
2. tuotekehitystä tukevan tietotekniikan aiempaa laajempi hyödyntäminen
3. tuotekehitystä tukevien prototyypiteknikoiden soveltaminen
4. tuotteiston hallinnan parantaminen
5. tuotetiedon hallinnan kehittäminen.

Erityisesti tutkimuksen toisella painopistealueella haluttiin siirtyä yrityksissä yleistyneestä CAD-suunnittelusta seuraavalle tasolle eli virtuaaliseen prototyypointiin ja simulointiin. Virtuaalisuunnittelumenetelmien kehittämistä pidettiin tärkeänä, koska virtuaalisimuloinnin avulla voitiin edistää konseptisuunnittelua ja siten nopeuttaa koko tuotekehitysprosessia. RAPID-ohjelman aikoihin virtuaalisuunnittelun tutkimukselle oli hyvät edellytykset, koska PC-työasemien las-

3. Simulointipohjaisen suunnittelun tutkimus Suomessa

kentateho oli riittävä 3D-työkalujen hyödyntämiseen. Ohjelmassa tehtiin myös ensimmäiset kokeilut kaupallisilla tuotetiedon hallintajärjestelmillä, mutta tavoitteita ei niiden osalta saavutettu johtuen pitkälti ohjelmistojen kehittymättömyydestä. (Kivikko 2004).

RAPID-ohjelman selkeästä tavoitteenasettelusta ja osaprojektien hyvistä tuloksista huolimatta ohjelmassa ei onnistuttu kehittämään uusia periaatteita tai menetelmiä, jotka olisivat tehostaneet tuotekehitysprosesseja merkittävästi tai muuttuneet käytännöiksi valmistavassa teollisuudessa. Saavutetut tulokset olivat pääasiassa projektikohtaisia ja hyödynnettävissä projekteihin osallistuvissa yrityksissä, mutta ei valmistavassa teollisuudessa laajemmin. (Eppinger & Synterä 2000, s. 27).

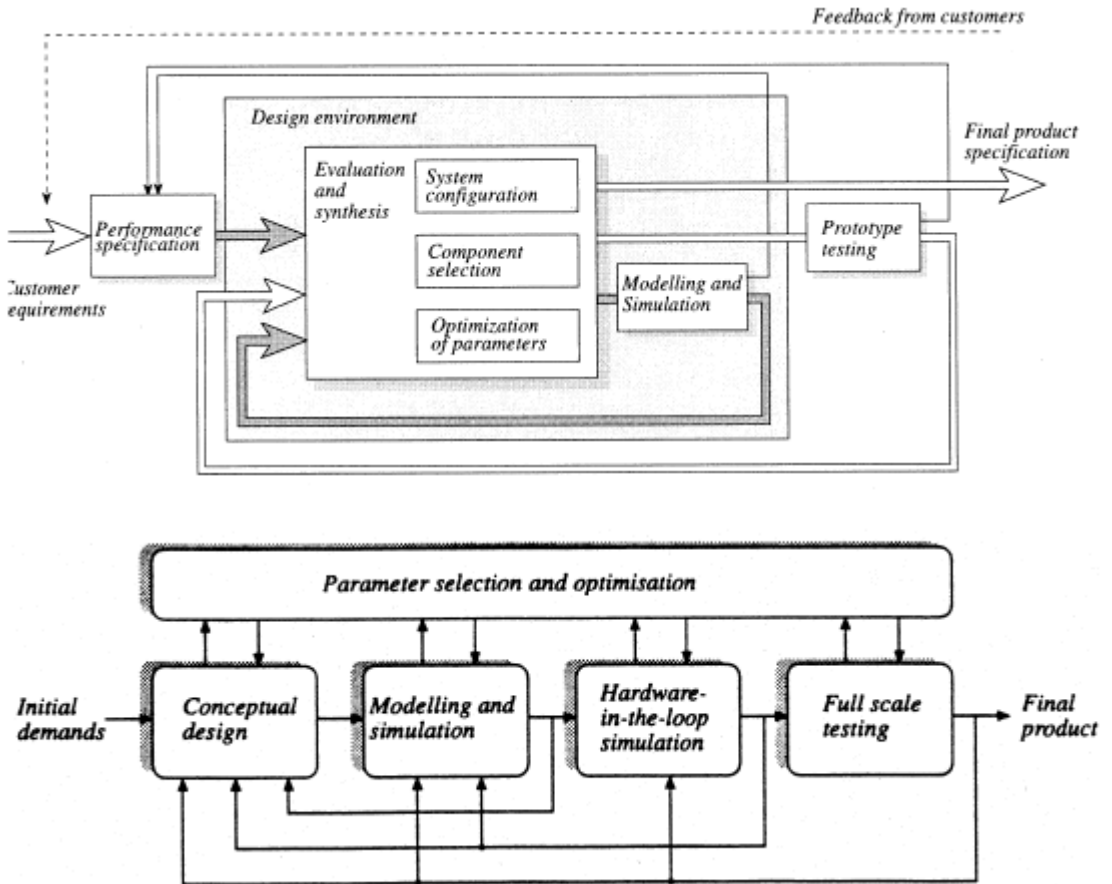
3.3 Huomisen koneet ja järjestelmät (SMART) 1997–2000

Tulevaisuusorientoituneen SMART-ohjelman tavoitteena oli helpottaa modernin tekniikan hyödyntämistä koneenrakennuksen ja sähkömekaniikan toimialoilla. Simulointityökalujen hyödyntämistä haluttiin edelleen edistää, ja ohjelman keskeisenä aiheena oli moniteknisten mekatronisten järjestelmien mallinnus ja simulointi (Kivikko 2004). Holopainen (2001) on koonnut yhteen SMART-ohjelman loppuseminaarissa esitetyt mallinnus- ja simulointitutkimukset.

SMART-ohjelman tulokset liittyivät ensisijaisesti uusiin mallinnusmenetelmiin ja -työkaluihin, mutta suunnitteluprosessin tarkastelu kokonaisuutena jäi vähemmälle huomiolle (Kivikko 2004). Loppuseminariaineiston (Holopainen 2001) perusteella voidaan todeta, että SMART-ohjelmassa koneita ja laitteita suunniteltiin tyypillisesti simulointiavusteisesti käyttäen simulointia erikoistyökaluna jonkin tuotteen yksityiskohdan analysoimisessa ja kehittämisessä. SMART-ohjelmassa saatiin silti myös kokemuksia simulointiperusteisesta suunnittelusta, jota käsitellään artikkeleissa Vessonen & Järviluoma (2001) sekä Ellman ym. (2001). Viimeksi mainitut tarkastelevat virtuaalisuunnittelun roolia tuotekehityksessä myös yleisemmällä tasolla ottaen huomioon asiakastarpeen ja -palautteen. He esittävät Palmbergin, Krusin ja Janssenin (1995) kehittämän virtuaalisuunnitteluprosessin, jossa hyödynnetään simulointia jo tuotekehityksen varhaisessa vaiheessa. (Kuva 2). Suunnitteluprosessi on kehitetty hydraulisille järjestelmille, jotka 90-luvulla olivat tyypillisesti moniteknisiä sisältäen mekaniikkaa, hydraulikkaa, antureita ja elektroniikka. Todennäköisesti osajärjestelmien yhteensovittamisen takia suunnitteluprosessi on myös korostuneesti iteraatiivinen ja jokaisesta vaiheesta on takaisinkytkentä edelliseen vaiheeseen. Palm-

3. Simulointipohjaisen suunnittelun tutkimus Suomessa

bergin ym. (1995) esittämää virtuaalisuunnittelun prosessimallia on kuitenkin hankala noudattaa käytännössä, koska se on esitetty liian karkealla tasolla. Esimerkiksi siihen ei oteta kantaa, kuinka eri osajärjestelmien (mekaniikka, elektroniikka, hydraulikka) suunnitteluun ja mallinnukseen tarvittavat ohjelmistot kytkeytyvät toisiinsa ja kuinka prosessin tuottamaa suunnittelutietoa hallitaan.



Kuva 2. Virtuaalisuunnitteluprosessi Palmbergin ym. (1995) ja Ellmanin ym. (2001) mukaan.

3.4 Koneenrakennuksen teknologiaohjelma (MASINA) 2002–2007

MASINA-ohjelman päätavoite oli parantaa suomalaisen koneenrakennuksen kansainvälistä kilpailukykyä sekä nykyaikaistaa koko toimialaa (TEKES 2008, Syrjänen ym. 2008). MASINA-ohjelman tutkimus kohdistettiin esiselvityksien ja visiotyöryhmän näkemyksen pohjalta neljään painoalueeseen (TEKES 2008):

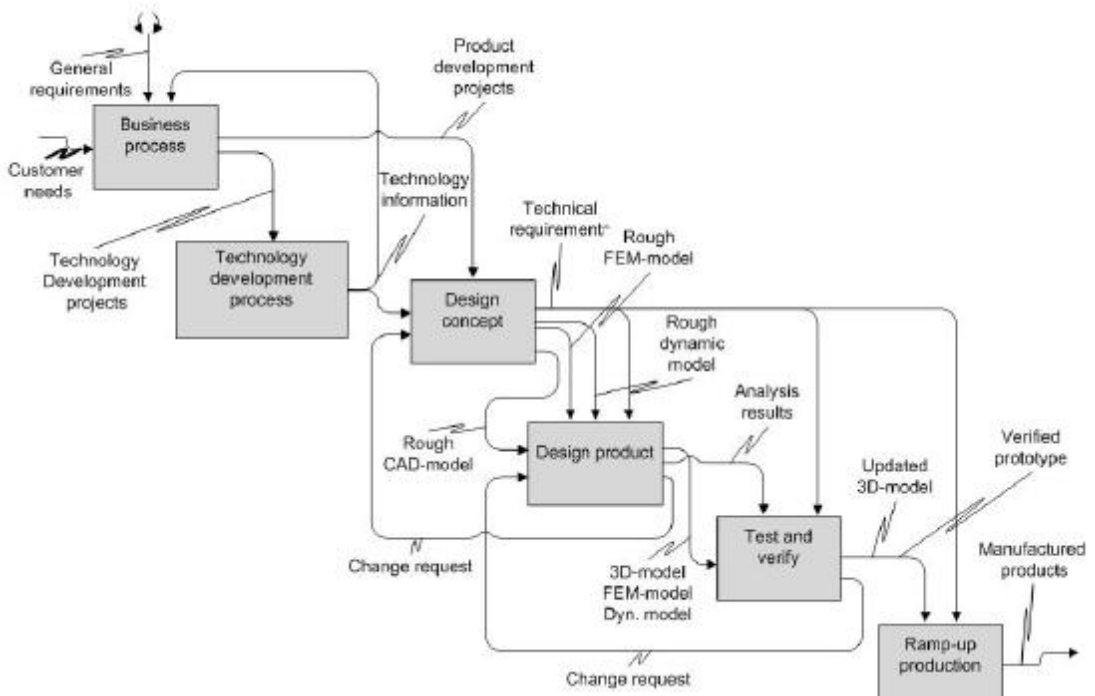
1. älykkäät koneet ja järjestelmät
2. edistykselliset rakenteet
3. elinkaaripalvelut ja -järjestelmät
4. kestävä kehitys ja käyttövarmuus.

MASINA-ohjelmaan osallistuneille yrityksille tehdyn palautekyselyn perusteella ohjelma lisäsi teknologiaosaamista, paransi kilpailukykyä sekä edisti yritysten ja tutkimuslaitosten verkostoitumista. Ohjelman vaikutukset palveluliiketoiminnan kehittämiseen jäivät kuitenkin vähäisiksi ja yritysten kilpailukykyä paransivat pääasiassa osaprojekteissa kehitetyt uudet tuotteet (Syrjänen ym. 2008).

Älykkäiden koneiden tuoteprosessia tutkittiin MASINA-ohjelmaan kuuluneessa KONEMASINA-projektissa vuosina 2002–2005. Projektin tavoitteena oli kehittää moniteknisten tuotteiden tuoteprosessia kohti rinnakkaista ja verkottunutta suunnittelua. Keskeisimmät tutkimusongelmat olivat simulointiin perustuvan suunnitteluprosessin vaikutukset tuotemallipohjaiseen tuotekehitykseen ja toisaalta verkottuneen tuotekehityksen toteutettavuus ja kustannustehokkuus (Lehtonen 2006).

KONEMASINA-projektissa kehitettiin kuusivaiheinen simulointiin perustuva tuotekehitysprosessi, joka otti huomioon yrityksissä vallinneet suunnittelukäytännöt (kuva 3). Ensimmäinen vaihe on liiketoimintaprosessi, jonka syötteenä ovat asiakastarve ja tuotteen yleiset vaatimukset. Toinen vaihe on teknologian kehitysvaihe, jossa tutkitaan tulevassa tuotteessa tarvittavia uusia teknologioita ja komponentteja. KONEMASINAN suunnitteluprosessin kahden ensimmäisen vaiheen toteuttaminen on periaatteeltaan samanlainen riippumatta siitä, onko suunnitteluprosessi simulointipohjainen vai ei. (Lehtonen 2006.)

3. Simulointipohjaisen suunnittelun tutkimus Suomessa



Kuva 3. Kuusiportainen simulointipohjainen tuotekehitysprosessi (Lehtonen 2006).

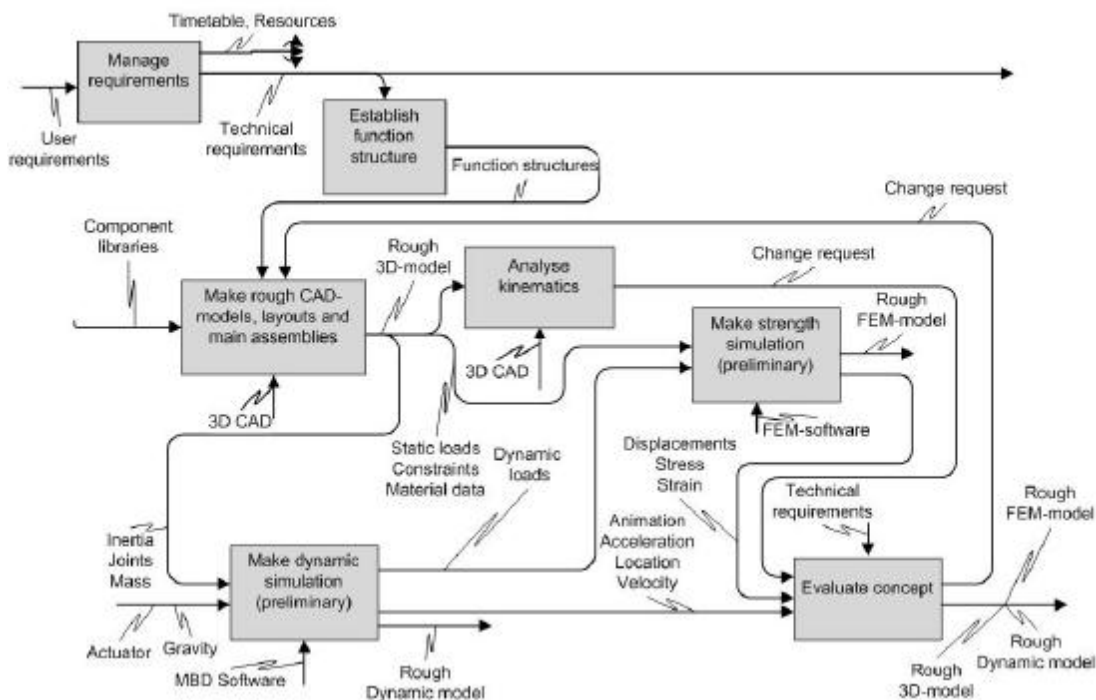
Mallinnus ja simulointi otetaan käyttöön suunnitteluprosessin kolmannessa vaiheessa, joka on tuotekonseptin suunnittelu (Design concept) (kuva 4). Konseptin suunnittelussa hyödynnetään teknologian kehittämissivaiheen tuloksia ja laaditaan yksityiskohtaisempi tekninen vaatimusmäärittely, jonka pohjalta tuotteen osista ja kokoonpanoista luodaan karkeat CAD-mallit. Karkeiden mallien avulla tehdään alustavat kinemaattiset ja dynaamiset simuloinnit sekä lujuustarkastelut, joiden perusteella malleja modifioidaan ja tarkennetaan edelleen (Lehtonen 2006). Karkeat simuloinnit ovat simulointipohjaisen suunnitteluprosessin tärkeä vaihe, jossa verrataan eri ratkaisuvaihtoehtoja, määritetään tuntemattomien parametriarvojen suuruusluokat ja luodaan perusta tuotteen yksityiskohtaiselle mallinnukselle (Paredis ym. 2001).

Tuotteen suunnitteluvaiheessa (Design product) suunnittelu ja mallinnus etenevät periaatteeltaan kuten konseptisuunnittelussa, mutta mallit ovat silloin huomattavasti yksityiskohtaisempia. Tuotteen suunnitteluvaiheessa voidaan lisäksi hyödyntää reaaliaikaisimulointia ihminen–kone-vuorovaikutuksen tutkimiseen ja tuotteen käytettävyyden kehittämiseen (Lehtonen 2006). Monitekni-

3. Simulointipohjaisen suunnittelun tutkimus Suomessa

ten koneiden reaaliaikasmulointia on käsitelty tarkemmin lähteissä (Schramm ym. 2005, Lehtonen 2006, Korkealaakso 2009).

Testaus- ja verifiointivaiheessa testataan ja todennetaan virtuaaliprototyypit ja rakennetaan ensimmäinen todellinen prototyyppi koko tuotteesta tai osajärjestelmästä. Testien tarkoitus on selvittää, vastaako tuotteen toiminta suunnittelua ja asiakkaan tarpeita. Testatuloksien perustella tuotteeseen tehdään tarvittavat muutokset ja virtuaaliprototyypit päivitetään vastaamaan todellisia prototyyppisiä (Lehtonen 2006, Paredis ym. 2001).



Kuva 4. Konseptin suunnitteluvaihe (Lehtonen 2006).

Tuotannon käynnistysvaiheessa simuloidaan tuotteen valmistus tavoitelluilla tuotantokoneilla. Tässä vaiheessa voidaan simuloida myös muiden tuotantokoneiden ja oheislaitteiden toimintaa, tehdassuunnittelua sekä materiaalivirtojen hallintaa. Tuotannon simuloinnin tavoitteena on varmistaa tuotannon joustava käynnistys ja poistaa siihen liittyvät ongelmat etukäteen (Lehtonen 2006). Simulointiin perustuva tuotantojärjestelmän suunnittelu on erityisen hyödyllistä asiakkaan toiveiden pohjalta räätälöitävien piensarjatuotteiden kohdalla, koska simuloinnin avulla voidaan minimoida tuotantojärjestelmän uudelleenkonfigu-

rontiin kuluva tuottamatonta aikaa sarjojen vaihdon välillä (Lalic ym. 2005) sekä optimoida raaka-aineiden käyttöä ja minimoida materiaalihukkaa (Heilala ym. 2008).

Yhteenvetona voidaan todeta, että KONEMASINAssa esitetty simulointipohjainen suunnitteluprosessi ohjaa hyödyntämään mallinnusta ja simulointia tuotekehityksessä varsin kokonaisvaltaisesti. Varsinkin ylimmän tason kuusiportainen prosessikaavio (kuva 3) on hyvin yleistetty, ja sen avulla voidaan vaiheistaa simulointilähtöinen tuotekehitys välietappihin. KONEMASINAssa ei kuitenkaan onnistuttu luomaan selkeää ja helposti noudatettavaa prosessimallia moniteknisten tuotteiden virtuaalisuunnittelulle. Esimerkiksi kuvan 4 mukaista suunnitteluprosessia on suhteellisen helppo noudattaa mekaniikkasuunnittelussa, mutta prosessimalli ei ohjaa eri osajärjestelmien rinnakkaista suunnittelua. KONE-MASINAssa esitetyn suunnitteluprosessin soveltaminen moniteknisten tuotteiden suunnittelussa on hankalaa pääasiassa seuraavista syistä:

- Esitetyt virtuaalisuunnittelun prosessimallit ottavat huomioon vain mekaniikkasuunnittelun, eivätkä ne sisällä muiden osajärjestelmien suunnittelua (esim. hydraulikka, elektroniikka, ohjelmisto).
- Prosessimallit eivät tue rinnakkaista ja verkottunutta suunnittelua.
- Suunnittelutiedon hallintaan ei ole esitetty ratkaisua.

3.5 Mallinnus ja simulointi (MASI) 2005–2010

MASI-ohjelma käynnistettiin tukemaan suomalaisen mallinnus- ja simulointitoimialan kehittymistä sekä parantamaan suomalaisen teollisuuden ja liiketoiminnan kilpailukykyä. Ohjelman tavoitteet olivat mallinnuksen ja simuloinnin laajempi hyödyntäminen teollisuudessa, mallinnus- ja simulointiprosessien innovointi sekä uusien kaupallisten mahdollisuuksien luominen. MASI-ohjelmassa simulointiosaamista pyrittiin myös levittämään suomalaisiin yrityksiin niiden koosta riippumatta, ja ohjelmaan osallistuikin iso joukko pk-yrityksiä, jotka joko seurasivat julkisia projekteja tai toteuttivat omia yrityskohtaisia projektejaan. (Holviala 2010.)

MASI-ohjelmassa toteutettiin 35 julkista tutkimusprojektia, joista suurimmasa osassa keskityttiin uusien mallinnusmenetelmien ja simulointityökalujen kehittämiseen. Koneenrakennuksen kannalta MASI-ohjelman julkiset tulokset näyttäisivät vastaavan SMART-ohjelman tuloksia, koska ohjelmassa kehitettiin pääasiassa uusia mallinnusmenetelmiä, mutta varsin vähän ratkaisuja niiden

3. Simulointipohjaisen suunnittelun tutkimus Suomessa

kustannustehokkaaseen hyödyntämiseen tuotekehityksessä. MASI-ohjelman vaikutukset palveluliiketoimintaan sekä mallinnuksen ja simuloinnin hyödyntämiseen pk-yrityksissä on myös arvioitu vähäisiksi (Lehenkari ym. 2010). Kooste MASI-ohjelman julkisista tutkimusprojekteista ja niiden tuloksista on lähteessä Holviala (2010).

MASI-ohjelman julkisessa osaprojektissa ”Simuloinnin ja suunnittelun uudet sovellustavat ja liiketoiminta” (SISU) tutkittiin simuloinnin käyttöönottoa ja hyödynnettävyyttä yritysten tuotekehityksen ja liiketoiminnan lähtökohdista. SISU-projektin tavoitteena oli kehittää uusia suunnittelumenetelmiä erityisesti pk-yrityksille sekä laatia suunnitelmia menetelmien teolliseen käyttöönottoon ja siten luoda edellytyksiä uudelle liiketoiminnalle. (Leppävuori, Olin, Valli ym. 2009).

SISU-projektissa kehitettiin SISUQ8-menetelmä, jonka tarkoitus oli konseptoida simulointityötä ja selventää sen merkitystä yrityksen liiketoiminnalle sekä kaikille simuloinnin osapuolille kuten asiakkaalle, käyttäjälle, johdolle, simuloijalle, kehittäjälle, jne. SISUQ8-menetelmässä simulointitoiminta jaetaan kahdeksaan vaiheeseen (Taulukko 1), joita edeltää perustavaa laatua oleva (nollatason) kysymys ongelmanratkaisun merkityksestä yrityksen liiketoiminnalle. Lisäksi jokaiseen vaiheeseen kohdistetaan tarkentavia lisäkysymyksiä koskien vaiheen määrittelyä (Miksi, Mitä, Milloin?), toteutustapaa (Miten?), edistymistä (Missä mennään?) ja toimivuutta (Kuinka toimii?). Vaiheistusta ja niihin kohdistuvista kysymyksistä muodostuvaa 8*4 matriisia voidaan edelleen käyttää apuvälineenä simulointiprojektin suunnittelussa ja hallinnassa. (Leppävuori ym. 2009).

Virtuaalisuunnittelun kannalta SISUQ8-menetelmän huono puoli on, että menetelmä on esitetty hyvin yleisellä tasolla. SISUQ8 onkin ensisijaisesti apuväline simulointiprojektien suunnitteluun ja hallintaan, mutta se ei ohjaa mallinnus- ja suunnitteluprosesseja riittävän tarkasti. Menetelmässä ei esimerkiksi esitetä ratkaisua (prosessimallia) moniteknisen tuotteen rinnakkaiseen mallinnukseen ja suunnitteluun. Menetelmä saattaa tosin auttaa tunnistamaan prosesseihin liittyvät ongelmat ja ohjata etsimään niihin ratkaisuja.

3. Simulointipohjaisen suunnittelun tutkimus Suomessa

Taulukko 1. Simulointitoiminnan vaiheistus SISUQ8-menetelmässä (Leppävuori ym. 2009).

	Vaiheet	Vaiheen lyhyt kuvaus
0	Vaikutus	Visio. Miten alla tarkemmin kuvatun ongelman ratkaisu auttaa liiketoiminnassa?
1	Ongelman määrittely	Strateginen tavoite. Ongelman määrittely.
2	Ratkaisu	Toiminnan tai ratkaisun suunnittelu, taktiikka. Projektisuunnitelma mallin laadintaan.
3	Konseptointi	Perusvalinnat valitulle taktiikalle. Laadittavan mallin sanallinen kuvaus
4	Mallin tiedot	Tarvikkeet konseptin toteuttamiseksi. Mallin rajoitukset, tarvittava data, tulosteiden alustava suunnittelu jne.
5	Mallien rakentaminen	Konseptin mukaisen mallin käytännön laatiminen toimimaan kohdan 4 tarvikkeilla. Mallin rakentaminen, vertailut ja kelpoistus.
6	Mallien soveltaminen	Varsinainen toiminta ongelman ratkaisemiseksi kohdan 5 mallia soveltaen tai jossain tapauksessa palaaminen edellisiin kohtiin. Mallin soveltaminen kohdissa 1–3 kuvatun ongelman ratkaisemiseksi, simuloinnit.
7	Tulosten esittäminen	Tulosten esittäminen sidosryhmien ymmärtämissä muodossa. Ratkaisun (tulosten) dokumentointi ja esittäminen.
8	Elinkaari	Tarvitaanko mallia jatkossa? Mallin ylläpito ja kehittäminen jatkossa.

3.6 Digitaalinen tuoteprosessi (DTP) 2008–2012

Vuosina 2008–2012 suoritettavan TEKESin Digitaalinen tuoteprosessi -ohjelman tavoitteena on vahvistaa suomalaisten yritysten kilpailukykyä, kasvumahdollisuuksia sekä tuotteen elinkaaren hallintaan ja palveluihin liittyvää osaamista. Ohjelman tavoitteet pyritään saavuttamaan erityisesti tietotekniikan soveltamisen tasoa nostamalla (TEKES 2011). DTP-ohjelma on jaettu kolmeen painopistealueeseen:

3. Simulointipohjaisen suunnittelun tutkimus Suomessa

1. prosessit ja johtaminen
2. implementointi ja osaamiset
3. uudet työkalut ja standardisointi.

Simulointilähtöisen tuotekehityksen jalkauttamiseksi yrityksiin tuotekehitysprosessin olisi otettava huomioon painopistealueilta seuraavia asioita: systemaattinen asiakastarpeiden määrittäminen ja hallinta, elinkaaren hallinnan tarpeet, moniteknisyyden tukeminen ja hallinta työkaluissa, järjestelmien yhteensopivuus sekä verkostomainen toiminta (Lehenkari ym. 2010).

MoniDigi-projektin lisäksi tuotetiedon hallintaa ja simulointipohjaisia suunnittelumenetelmiä tutkitaan seuraavissa Digitaalinen tuoteprosessi -ohjelman julkisissa projekteissa:

- Simulaatiomallien elinkaarenhallinta -projekti – Simulation Life Cycle Management (SLM) – keskittyy moniteknistien tuotteiden simulointimallien hallintaan ja kehittää uutta tiedonhallintamenetelmää. Aalto-korkeakoulusäätiö.
- PLM – Dynamic and Digital Product Structure -projektin tavoitteena on määrittää elementit dynaamiselle ja digitaaliselle tuoterakenteelle, jota voidaan hyödyntää läpi tuotteen elinkaaren. Oulun yliopisto.
- Tietokantakeskeinen koneenohjausjärjestelmien suunnittelu -projektissa testataan koneenohjausjärjestelmien suunnittelutiedon tietokantakeskeistä hallintaa ja jakamista suunnitteluryhmien ja alihankkijoiden kesken. Teknologian tutkimuskeskus VTT.
- Laskennalliset mallit osana tuotteen elinkaarta -projektissa tutkitaan semanttisen tietomallinnuksen soveltuvuutta teollisen mittakaavan suunnittelu- ja tuoteprosessin tiedonhallintaan sekä laskennallisten mallien integrointiin. Projektin toisena laajana tutkimusalueena on tuotteen elinkaaren simulointipohjainen hallinta. Teknologian tutkimuskeskus VTT.

4. Nykytila ja kehittämismahdollisuudet pk-yrityksissä

4.1 Tuotesuunnittelu

Pienissä ja keskisuurissa yrityksissä tuotteiden suunnittelu ja valmistus edellyttävät yhä enemmän moniteknistä tuoteosaamista. Suunnitteluun, valmistukseen ja tuotantoon liittyvät kehittämisen yhteistyöhankkeet on osattava organisoida uudella tavalla. Käytännössä uusien teknologioiden soveltamisen esteenä pk-yrityksissä on osaavan henkilöstön sekä kehittämisvälineiden ja ohjelmistojen puute. Uusien teknologioiden soveltamista ja käyttöönottoa voidaan tehostaa yrityksissä verkottuneella toimintatavalla.

Moniteknisen tuotteen kustannustehokas suunnittelu vaatii kehittyneitä suunnitteluohjelmistoja, jotka ovat usein liian kalliita erityisesti pk-yrityksille. Hajautettu, verkottunut tuotekehitys tarjoaa kustannustehokkaamman tavan hyödyntää kalliita suunnitteluohjelmistoja. Pk-yritysten tuotesuunnittelun tehostamiseksi tarvitaan kaupallisia toimijoita, jotka tarjoavat yrityksille esim. virtuaalisuunnitteluohjelmien verkkolisenssiaikaa tai suunnittelupalveluja. Tällä hetkellä moniteknisen tuotteen virtuaalisuunnitteluun liittyviä palveluja tarjoavat pk-yrityksille lähinnä tutkimuslaitokset ja yliopistot, jotka ovat hankkineet kaupalliseen käyttöön tarkoitetut ohjelmistolisenssit. Tällöin erityisen tärkeää on tutkimus- ja kehittämispalveluja tarjoavien tahojen sekä yritysten vuorovaikutus ja yhteistyö.

4.2 Tuotetieto ja tuotteen elinkaaren hallinta

Monitekninen tuote vaatii paljon osaamista, ja tuotteen räätälöinti asiakkaiden tarpeisiin edellyttää uusia ratkaisuja piensarjatuotteiden suunnitteluun sekä tuotetiedon ja valmistusprosessin hallintaan. Asiakaskeskeinen toimintatapa lyhentää tuotteiden elinkaarta, jolloin uudet tuotevariaatiot lisääntyvät. Eri tuotevariaatiot aiheuttavat tuotetiedon määrän nopean kasvun. Tuotetiedon hallinta vaatiikin yritysten kasvaessa ja kansainvälistyessä tehokkaita järjestelmiä ja toimintatapoja. Tällä hetkellä on paljon pk-yrityksiä, joilla ei ole käytössä tuotetiedon hallintajärjestelmiä, vaan tietoa hallitaan projekti- tai asiakaskohtaisesti esim. Microsoft Windows -työkaluilla. Tästä seuraa suuria ongelmia tiedon päivittämisessä ja jakamisessa tuoteprosessin eri vaiheissa. Yhtenä visiona voisi olla tuotteen suunnittelun ja valmistuksen kattavan yritysverkoston laajuinen tuotetiedon hallintajärjestelmä, joka sisältää kullekin verkoston yritykselle olennaisen tuotetiedon. Myös tuotteen elinkaaren hallinnan kannalta on tärkeää nähdä reaaliajassa, missä tuote on käytössä ja onko siihen tehty muutoksia tai korjauksia.

Useat kaupalliset järjestelmätoimittajat tarjoavat ratkaisuja tuotetiedon hallintaan. Tuotetiedon hallintajärjestelmät (PDM) ovat kehittyneet suunnittelutiedon hallintajärjestelmistä (EDM), jotka ovat useimmiten lähtöisin tietokoneavusteisen suunnitteluohjelmistojen (CAD) dokumenttien hallintajärjestelmistä. Tänä päivänä PDM-järjestelmien pohjalta on kehitetty tuotteen elinkaaren hallintajärjestelmiä (PLM), jotka ovat PDM-järjestelmiä laajempia ja kokonaisvaltaisempia. Rajapinta tuotannon ohjaukseen hämärtyy yhä enemmän, ja se on määriteltävä aina yritysکوhtaisesti.

Mainituista järjestelmistä aiheutuvat kustannukset muodostuvat pääosin

1. käyttölisenssien hankintahinnasta
2. vuotuisista ylläpito- ja käyttäjätukimaksuista
3. käyttöönottoprojektin kustannuksista
4. koulutuskustannuksista.

Lisenssit myydään yleensä käyttäjäkohtaisesti, joten niistä aiheutuvat kulut ovat suhteessa käyttäjien lukumäärään, samoin kuin koulutusmaksut. Suuruusluokaltaan hankintakulut ovat useista tuhansista euroista useisiin kymmeneen tuhansiin euroihin. Ylläpitomaksut ovat vuosittain noin 15–25 % lisenssin hankintahinnasta. Käyttöönottoprojektin kustannukset ja kesto riippuvat monesta eri tekijästä, kuten yrityksen koosta ja olemassa olevan tuotetiedon määrästä sekä laadusta. Kustannukset ovat suuruusluokaltaan kymmenistä tuhansista euroista satoihin

tuhansiin euroihin ja kesto viikoista useisiin kuukausiin. Koulutuskustannukset ovat tyypillisesti joitain tuhansia euroja. Pk-yrityksissä kynnyskysymyksenä PDM-/PLM-järjestelmien käyttöönotolle ovat paitsi kustannukset myös järjestelmän valinta. Tarjoajia ja erityyppisiä järjestelmiä on lukuisia, ja eri järjestelmien ominaisuudet ja laajuus voidaan räätälöidä yrityskohtaisesti, joten yrityksillä on paljon valinnanvaraa. Koska tällaisten järjestelmien käyttöönotto vaatii tyypillisesti suuria muutoksia yritysten toimintatapoihin, hankinta on harkittava erittäin perusteellisesti, mikä yhdistettynä suureen tarjontaan johtaa työläaseen ja kalliiseen valintaprosessiin. Valintaprojektia varten on suositeltavaa varata riittävät resurssit, jotta hankintapäätökselle tarvittavaa tietoa saadaan riittävästi. Monet yritykset ovat tänä päivänä tunnistaneet tarpeen tuotetiedon järjestelmälliselle hallinnalle, mutta jo valintaprosessiin liittyvät resurssitarpeet ja monien yritysten tiukka taloudellinen tila ja aikataulut hidastavat PDM-/PLM-järjestelmien laajempaa käyttöä.

4.3 Digitaalinen osavalmistus ja tuotanto

Tuotteiden valmistus pk-yrityksissä on tyypillisesti piensarjatuotantoa, joka tuo haasteita tuotetiedon, osavalmistuksen ja kokoonpanon kustannustehokkaaseen hallintaan. Moniteknisissä tuotteissa tarvitaan tietotekniikan, tietoliikennetekniikan ja elektroniikan soveltamista ja sulauttamista tuotteisiin. Langaton tiedonsiirtotekniikka ja ohjaustekniikan miniatyrisointi mahdollistavat ohjauksjärjestelmien integroimisen yhä pienempiin ja halvempiin tuotteisiin ja komponentteihin. Tuotteiden tietosisältö lisääntyy jatkuvasti mm. automaation, tietotekniikan, etähallinnan ja käytönaikaisen diagnostiikan yleistyessä, jolloin tuotetiedon sekä tuotantoprosessin hallintaan tarvitaan uusia työkaluja.

Tuotteiden valmistuksen kansainvälistyessä myös pk-yrityksiltä edellytetään yhä nopeampia ja laadukkaampia keinoja valmistaa tuotteita. Tuotteiden lyhenneet elinkaaret ja tuotteiden asiakaskohtaisen räätälöintimallin käyttöönotto ovat johtaneet yritykset tilanteeseen, jossa paineet valmistusaikojen lyhentämiseksi kasvavat koko ajan. Monitekniset tuotteet ovat tavanomaista monimutkaisempia mm. osavalmistusta ja tuotteen kokoonpanoa ajatellen. Tuotesuunnittelun ja valmistuksen integroinnin avulla tilaus-toimitusprosessin läpäisyäikää on kyetty lyhentämään merkittävästi (esim. koneistettavat kappaleet). Automaattinen työstösuunnittelu ja simulointi ovat nopeuttaneet tietokoneavusteista valmistusta ja parantaneet tuotteiden laatua merkittävästi. Tuotesuunnittelun ja valmistuksen integroinnin suurin etu on siinä, että tuotteen kaikki informaatio on

4. Nykytila ja kehittämismahdollisuudet pk-yrityksissä

siirtynyt papereiden sijasta digitaaliseen muotoon. Tämä mahdollistaa niin suunnittelun kuin valmistuksenkin globalisoinnin. Eri toiminnot voidaan hajauttaa ja verkottaa, sillä tuotemalli voidaan siirtää hetkessä toiselle puolelle maapalloa. Samalla siirtyy myös tuotteen valmistukseen tarvittava informaatio.

4.4 Haastattelututkimukset

MoniDigi-projektin aikana on tehty erilaisia haastattelututkimuksia, joiden avulla on pyritty saamaan tietoa pk-yritysten nykytilanteesta. Tutkimuksia on tehty Suomessa, Saksassa ja Etelä-Koreassa. Suomessa on haastateltu konepajasektorin yritysten lisäksi rakennussektorilla toimivia yrityksiä. Rakennussektorin tuotanto voidaan rinnastaa hyvin myös piensarjatuotantoon: esimerkiksi kahta täysin samanlaista siltaa tai tietä tullaan tuskin koskaan tekemään. Haastattelujen avulla selvitettiin näiden toimialojen välisiä eroja ja synergioita.

Tavoitteena oli selvittää, millä tasolla pk-yritysten digitaaliseen tuoteprosessiin liittyvät toiminnot ovat ja miten tiedonsiirto yhteistyöverkostossa on järjestetty. Yrityksille tehtiin kysely, jonka avulla pystyttiin selvittämään mahdollisimman hyvin tarvittavat tiedot. Kyselyä varten laadittiin lomake, jonka tuli olla niin selkeä ja yksiselitteinen, että sitä voitiin käyttää sellaisenaan lähettämällä lomake sähköpostilla haastateltavalle yritykselle. Osaan yrityksistä suoritettiin vierailuja, joissa lomaketta käytettiin keskustelun runkona. Tällä varmistettiin, että kaikilta osapuolilta kysyttiin samat perusasiat. Yrityskäyntien yhteydessä esitettiin tarkentavia kysymyksiä, joiden avulla saatiin hyvä kuva yrityksen toiminnasta ja yhteistyöverkostoista.

Lomakkeet lähetettiin yrityksille sähköisenä, mikäli ei sovittu alusta alkaen yritysvierailusta. Palautettujen kyselylomakkeiden tiedot kerättiin yhteen ja analysoitiin.

Kyselyn aihealueet olivat:

1. käytössä olevat ohjelmistot
2. yritysten digitaaliseen tuoteprosessiin liittyvät prosessit
3. yhteistyöverkoston kartoitus
4. yritysten johdon ja tuotetiedon hallinnasta vastaavien henkilöiden näkemykset digitaalisesta tuoteprosessista ja siihen liittyvät yrityksen kehitystarpeet.

4.5 Kyselyn tulokset

Nykytilan kartoitus antoi kuvan siitä, millä tasolla digitaalista tuoteprosessia pk-yrityksissä ja rakennussektorilla nykyään sovelletaan. Laajempia johtopäätöksiä suomalaisten tai kansainvälisten yritysten tilanteesta ei tämän kyselyn pohjalta voitu tehdä suhteellisen suppean otannan vuoksi. Tietynlaisia asioita tuli kuitenkin hyvin esille, ja niiden voitiin olettaa pätevän laajemminkin. Näistä esimerkiksi voidaan mainita se, että suunnittelussa käytetään jo hyvin laajasti CAD-suunnitteluohjelmistoja, jotka tuottavat tuotetiedon digitaalisessa muodossa. Näitä ohjelmistoja käytetään mekaniikkasuunnittelun lisäksi myös esimerkiksi sähkö-, laitos- tai rakennussuunnittelussa. Näin ollen digitaalisen tuoteprosessin lähtöedellytykset ovat olemassa sekä konepaja- että rakennussektorilla. Saksassa voimakkaasti esillä olevassa auto-, mutta myös lento- ja avaruusteollisuudessa toimii hyvin suuri määrä erikokoisia yrityksiä. Varsinkin isompien yritysten osalta voidaan olettaa, että digitaalinen tuoteprosessi on huomattavasti pidemmällä kuin tähän kyselyyn osallistuneilla pk-yrityksillä. Tämä ilmenee esimerkiksi kehittyneiden suunnittelujärjestelmien toimittajien referenssiluetteloista, kuten esimerkiksi Dessault Systèmes, PTC tai Siemens PLM Software. Osa Saksan kyselyyn osallistuneista pienistä yrityksistä toimivat esimerkiksi autoteollisuuden alihankkijoina, ja he kertoivat muun muassa digitaaliseen tiedonsiirtoon liittyvistä tiukoista turvamääräyksistä ja päämiehensä PLM-järjestelmän käytöstä.

Yleisesti ottaen valtaosa tuotetiedosta hallitaan sähköisessä muodossa. Joko se on tuotettu valmiiksi digitaaliseen muotoon tai vanhat paperidokumentit on siirretty sähköiseen muotoon esimerkiksi skannaamalla. Kyselyssä selvisi kuitenkin, että suuressa osassa pk-yrityksistä tämä digitaalinen tuotetieto hallitaan ilman varsinaisia siihen tarkoitettuja järjestelmiä. Yleinen tapa hallita vaikkapa yrityksen lähiverkossa olevalla kovalevyllä oleva tietoa on esim. Windows resurssienhallinnan kansiorakenne. Jokainen yritys on kehittänyt oman tapansa tallentaa tuotetietoa esim. projekti- tai asiakaskohtaisiin kansioihin. Menetelmä on hyvin yksinkertainen, eikä se vaadi lisäohjelmistoja ja on lisäksi yleisesti tunnettu. Haittoina voidaan mainita, että käyttäjillä on oltava selkeät ohjeet, miten ja mihin tietyt asiat tallennetaan tai miten versiointi hallitaan. Jokaisen käyttäjän vastuulla on se, että annettuja ohjeita noudetaan, jotta tuotetieto on myös myöhemmässä vaiheessa vielä kaikkien löydettävissä. Tämä korostuu puutteellisten hakutoimintojen myötä, sillä järjestelmä ei kuitenkaan ole tietokantapohjainen. Tietokantapohjaisten järjestelmien etuna on muun muassa, että tieto voidaan hakea erilaisten ennalta määritettyjen parametrien avulla. Kansioihin tallennettu-

4. Nykytila ja kehittämismahdollisuudet pk-yrityksissä

jen tiedostojen välille ei pystytä luomaan ainakaan kovin monimutkaisia relaatioita, mitä vaikeuttaa tuotetiedon ylläpitoa jonkun tiedon muuttuessa. Jotkin kone suunnitteluyrityksistä hyödynsivät CAD-järjestelmien omia keveitä dokumenttien hallintasovelluksia. Etenkin rakennussektorilla on hyvin yleisesti käytössä eri suunnitteluohjelmistojen omia tiedonhallinta- ja synkronointisovelluksia serveripohjaisten projektipankkien lisäksi.

Johtopäätöksenä voidaan todeta, että pk-yrityksissä olisi digitaalisen tuoteprosessin näkökulmasta paljon parannettavaa, mutta käytännön rajoituksina ovat PDM-/PLM-järjestelmien hinta, käytettävyys ja monimutkaisuus. Hinta ja hyötynäkökohdat on ratkaistava yrityskohtaisesti, mutta myös järjestelmätoimittajien sekä -kehittäjien on tarjottava tulevaisuudessa pk-yrityksille suunnattuja parannuskeinoja tähän ongelmaan. Perinteisesti kaupalliset järjestelmät on kehitetty etenkin suurille yrityksille ja globaaleille konserneille.

Varsinkin pienten yritysten osalta on havaittavissa, että jo hankittujen järjestelmien ominaisuuksia laajennetaan mieluummin kattamaan muita osa-alueita kuin ostettaisiin siihen tarkoitettu uusi järjestelmä. Tähän vaikuttavat kustannukset, käyttökokemus ja osaaminen, mutta myös se tosiasia, ettei yrityksellekään aina ole selvää, missä vaiheessa uusi järjestelmä kannattaa hankkia tai minkälaisia muita järjestelmiä ylipäättänsä on olemassa. Jos nykyisen järjestelmätoimittajan valikoimasta ei löydy uudelle osa-alueelle soveltuvaa tuotetta, voi usein käydä niin, että järjestelmätoimittaja suosittelee konfiguroimaan olemassa olevaa sovellusta. Tästä syystä yrityksen on aika ajoin hyvä tarkastella kokonaistilannetta prosessien, toimintamallien ja käytettyjen järjestelmien osalta. On selvää, ettei yhtä ainoaa ratkaisua ole olemassa, vaan jokaisen yrityksen on tehtävä päätöksensä vallitsevan tilanteen mukaisesti.

Vaikka suunnittelussa tuotetietoa synnytetään ja hallitaan digitaalisesti, se usein muutetaan valmistusta varten käytännössä piirustukseksi, josta koneistaja lukee tuotetiedon ja hyödyntää sitä sitten työssään. Ideaaliprosessin kannalta tämä ei ole toivottava välivaihe, joten sen poistamiseen on syytä panostaa. Osittain tämä vaihe on jo digitalisoitu CAM-teknologian avulla, mutta käyttöastetta voitaisiin nostaa vielä huomattavasti. Sukupolvenvaihdos ja uudenlainen lähestyminen tietotekniikkaan sekä laitteiden ja järjestelmien kehitys edesauttavat tätä prosessia. Haastattelujen pohjalta voidaan todeta, että etenkin Saksassa yritysten mahdollisuudet hyödyntää digitaalista tuotetietoa ovat varsin hyvällä tasolla. Valmistus rajoittuu tämän kyselyn osalta lähinnä lastuavaan työstöön, koska otannassa ei ollut mukana yrityksiä, jotka käyttävät esimerkiksi hitsausrobotteja tai levytyökeskuksia. Todettakoon tässä yhteydessä, että myös näiden valmistus-

prosessien osalta olisi teknisesti mahdollista hyödyntää digitaalista tuotetietoa, mihin digitaalisessa tuoteprosessissa olisikin pyrittävä. Myös rakennussektorilla valmistusvaiheessa käytetään vielä paperille tulostettuja piirustuksia, koska eri työvaiheisiin kuuluu paljon käsin tehtäviä vaiheita. Konepajasektorin CAM-teknologiaan rinnastettava työvaihe rakentamisen digitalisoitumisen osalta on massansiirto, jossa rakennettavan kohteen alle jäävä maa-aines on vaihdettava tai sen ominaisuuksia on parannettava erilaisin vahvistusmenetelmin. Nykypäivänä tässä työkonevetoisessa prosessivaiheessa käytetään hyvin paljon digitaalista suunnittelutietoa hyödyksi erilaisissa koneohjausmalleissa.

Tiedonsiirrossa niin konepaja- kuin rakennussektorilla turvaudutaan suurilta osin sähköpostiin ja liitetiedostoihin. Tietoturvan ja liitetiedostojen versioinnin kannalta menetelmä ei ole paras mahdollinen, mutta sähköpostin yleinen ja helppo käyttö puoltavat menetelmää. Osittain tiedonsiirtoon on jo ryhdytty käyttämään PLM-järjestelmiäkin, mutta käyttöaste on vielä hyvin alhainen niin Suomessa, Saksassa kuin Etelä-Koreassa. Suojattua tiedonsiirtoa vaativat tyypillisesti isot yritykset tai yritykset, jotka toimivat erikoisteollisuudessa. Esimerkkinä voidaan tässä mainita kilpa-autoilu (kuten Formula 1), ydin- ja puolustusvälineiteollisuus tai avaruusteollisuus.

Yksi tiedonsiirron ongelmakohta on tietojen siirtoon soveltuvan yleispätevän formaatin puute. Uusia protokollia on kehitetty jo vuosikymmeniä, mutta tehtävä on hyvin haastavaa. Teknisten haasteiden, kuten ohjelmistojen kirjo ja niiden toisistaan eroavien tietorakenteiden, lisäksi törmätään myös järjestelmävalmistajien markkinapoliittisiin kysymyksiin. Monet suuret ohjelmistotekijät laajentavat omia järjestelmiä ja pyrkivät tarjoamaan asiakkailleen kokonaisvaltaisia ratkaisuja. On hyvin todennäköistä, että kyseiset valmistajat kehittävät integroituja ratkaisuja lähinnä omien ohjelmistojensa osalta, jotta he pystyisivät sitouttamaan asiakkaitaan paremmin. Asiakkaat ovat tällöin vahvasti riippuvaisia toimittajistaan, mikä ei ole kovin toivottavaa esimerkiksi hintakilpailun kannalta. Puolueettomat tahot kehittävät kuitenkin neutraaleja siirtoformaatteja ja paineet valmistajiin päin tarjota parempia tiedonsiirtomahdollisuuksia kasvavat todennäköisesti jatkuvasti. Onhan valmistajankin kannalta kynnys päästä toisen valmistajan tai asiakkaan osajärjestelmätoimittajaksi matalampi, jos uusi järjestelmä integroituu olemassa olevaan järjestelmään saumattomasti.

Yleisesti voidaan todeta, että digitaalinen tuoteprosessi on asia, joka kiinnostaa laajalti ja josta halutaan lisätietoa – kiinnostusta löytyi yleisesti kaikista maista, joissa kyselyt tehtiin. Käytännön rajoitukset esimerkiksi perinteisessä ajattelutavassa ja toimintamalleissa kuitenkin jarruttavat digitaalisen tuotepro-

4. Nykytila ja kehittämismahdollisuudet pk-yrityksissä

sessin leviämistä. Toinen yleinen ongelma on järjestelmien hinta ja käytettävyys sekä se tosiasia, ettei investoinnin hyöty tai takaisinmakuaika ole selkeästi mitattavissa. Näin ollen taloudellisia perusteluja on vaikea käyttää hankintaesityksen tukena. On törmätty tilanteisiin, joissa esimerkiksi suunnittelija teknisestä näkökulmasta puoltaa järjestelmän hankintaa ja näkee sen hyödyllisenä, mutta yrityksen johto on investoinnin hyödyistä eri mieltä. Varsinkin PLM-järjestelmän hankinta on enemmän kuin vain tietokoneohjelmiston osto. Jotta järjestelmästä ja sen tarjoamista mahdollisuuksista saataisiin suurin mahdollinen hyöty, on yritysjohdon oltava päätöksen takana ja valmis muokkaamaan toimintamalleja ja prosesseja niin, että niitä voidaan tehostaa järjestelmän avulla. Alkupanostus on tyypillisesti suuri sekä ajallisesti että taloudellisesti, mutta hyöty saadaan maksimoitua, kun järjestelmä on tehokkaassa käytössä.

5. Tuotetiedon hallinta

5.1 Tuotetieto

Tuotetietoa on kaikki se tieto, joka syntyy tuotteen määrittelystä tuotteen käytöstä poistamiseen asti. Tuotetieto voidaan Sääksvuoren ja Immosen (2002) mukaan jakaa kolmeen ryhmään: tuotteen määrittely- ja elinkaaritietoon sekä tuotetietoa kuvaavaan metatietoon. (Sääksvuori & Immonen 2002). Esimerkiksi tuotesuunnittelussa, valmistuksessa, markkinoinnissa ja huoltotoiminnassa syntyy paljon informaatiota. Tämä tuotetieto on oltava hallittua ja saatavana liiketoiminnan jokaisessa prosessissa. Tähän liittyy suuria haasteita, varsinkin, kun asiakokonaisuutta voidaan lähestyä monelta eri kantilta, kuten tuotesuunnittelun, valmistuksen tai yleisesti liiketoiminnan näkökulmasta. Jokainen osapuoli tarvitsee erityyppistä tuotetietoa omiin prosesseihinsa ja ryhmittelee sitä omien tarpeidensa mukaisesti. Ideaalitapauksessa tuotetieto hallitaan niin, että kaikki tieto on kaikille osapuolille saatavilla niiden tarvitsemassa muodossa. Käytännössä tämä on kuitenkin hyvin vaikea toteuttaa. Suurin ongelma lienee siinä, miten yleensä erittäin suuren tietomäärän erilaiset sisäiset riippuvuudet hallitaan ja ylläpidetään, kun tehdään muutoksia. Lukuisat eri tahot muokkaavat tätä tuotetietoa eri vaiheissa ja eri näkökulmasta. Jotta tällaista kokonaisuutta pystytään hallitsemaan, tietotekniset ratkaisut ovat miltei välttämättömiä.

5.2 Tuotetiedon hallinta

Tuotetietoa hallitaan keskitetysti niin, että eri osapuolet pystyvät hallitsemaan heille olennaista tuotetietoa sekä tuotteeseen liittyviä prosesseja. Tiedon tallentamisen lisäksi tähän liittyy hallittu tiedon jakaminen yrityksessä ja sen verkostoissa sekä käyttöoikeuksien hallinta, jossa määritellään oikeudet katsella tai muokata tuotetietoa. Etenkin verkostomaisessa toimintaympäristössä luotettava

5. Tuotetiedon hallinta

tuotetiedon hallinta on yksi menestyksen avaintekijä, sillä se mahdollistaa nopean, turvallisen ja reaaliaikaisen tuotetiedon siirron ja saatavuuden verkoston eri osapuolille.

Tuotetiedon hallinta voidaan jakaa esimerkiksi seuraaviin pääalueisiin (Peltonen ym. 2002):

- nimikkeiden hallinta
- dokumenttien hallinta
- tuoterakenteiden hallinta
- muutosten hallinta.

Perinteisesti tuotetiedon hallinnan pääpaino on ollut tuoteprosessin alkupäässä eli suunnittelutiedon hallinnassa. Tästä seuraa, että monet järjestelmät tukevat erinäköisiä versiointi-, tarkastus- ja hyväksymiskäytäntöjä. (Peltonen ym. 2002.) Sen käyttö on kuitenkin laajentunut kattamaan koko tuotekehitystä ja sittemmin tuotteen koko elinkaaren, jolloin puhutaan tuotetiedon elinkaaren hallinnasta. Eri yhteyksissä käytetyt termit ja etenkin lyhenteet ovat osittain hyvin sekavia, sillä toisinaan tuotetiedon hallinta (PDM) käsittää yhtä lailla tuotteen elinkaaren hallintaa, jolloin voisi paremminkin puhua PLM:stä. On siten aina syytä selvittää, mitä milläkin termeillä tarkasti ottaen käsitetään. Lisäksi on aina määriteltävä, käsittääkö termi tuote tiettyä tuotesarjaa vai yhtä tuoteyksilöä. Vuonna 1995 Kenneth McIntosh (1995, käänös Sääksvuori & Immonen 2002) määritteli tuotetiedon hallintaa seuraavasti:

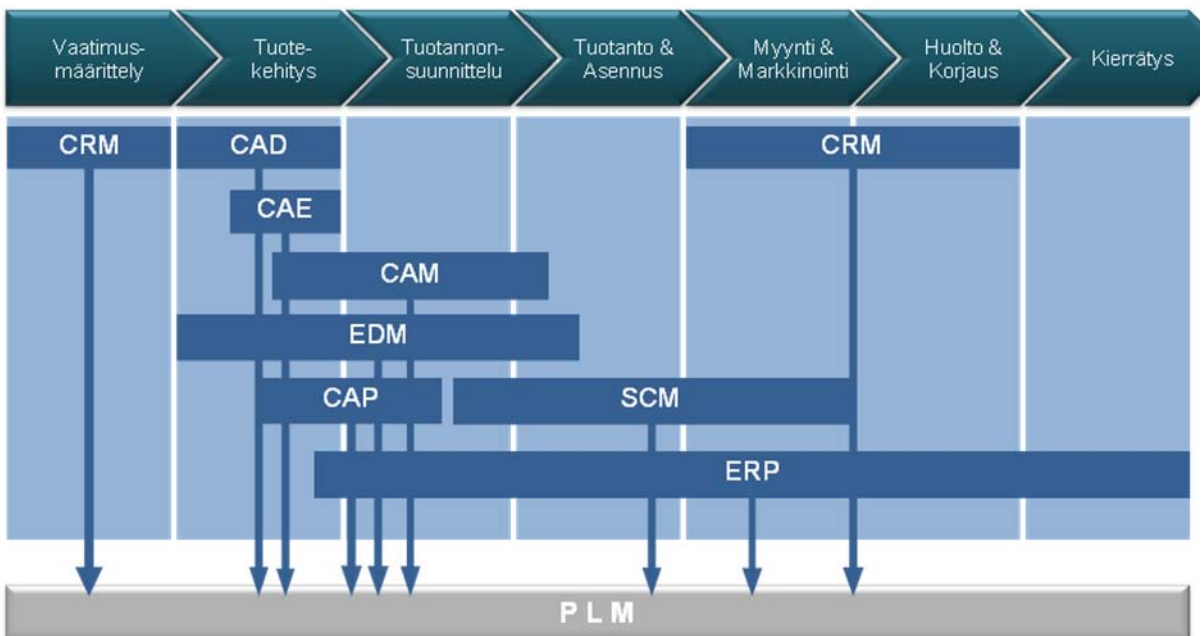
”Tuotetiedon hallinta on systemaattinen tapa suunnitella, hallita, ohjata ja valvoa kaikkea sitä tietoa, jota tarvitaan tuotteen dokumentoimiseksi, tuotteen kehittämis-, suunnittelu-, valmistus-, testausprosessin ja käytön aikana, tuotteen koko elinkaaren ajan.”

Tähän vielä tänä päivänä pätevään määritelmään on syytä lisätä myös tiedonvaihto ja -siirto, joka on osa tehokasta tuotetiedon hallintaa ja jolle nykyaikaiset järjestelmät tarjoavat hyviä mahdollisuuksia. Voidaan todeta, että yllä mainittu määritelmä sopii myös tuotteen elinkaarihallinta-termille, jota nykypäivänä käytetään enemmän.”

5.3 Tuoteprosessiin liittyvät tietojärjestelmät

Tuotetiedon ja tuoteprosessien tehokkaaseen hallintaan ja valvontaan on ajan saatossa kehitetty lukuisia järjestelmiä liiketoiminnan eri osa-alueille (kuva 5).

Näitä järjestelmiä on itse tuotetta ja sen kehitystä sekä valmistusta varten (kuten EDM ja ERP), mutta myös hallitsemaan liiketoiminnallisia prosesseja ja tietoja (kuten CRM ja SCM). Nämä ryhmittelyt ovat karkealla tasolla, ja selkeät rajapinnat on jokaisessa yritystapauksessa määriteltävä yksilöllisesti, jotta eri järjestelmien roolit ovat yksiselitteisesti käyttäjien tiedossa. Kyseisiä järjestelmiä voi olla siten hyvin monta, ja niiden avulla hallitaan tuotetietoa, prosesseja sekä muuta liiketoiminnan kannalta tärkeää informaatiota. Tyypillisesti näitä järjestelmiä on yrityksissä otettu käyttöön vaiheittain yrityksen liiketoiminnan kehityksessä ja kasvaessa. Aluksi on hankittu esimerkiksi järjestelmä tuotannon ohjausta, materiaalivirtojen ja varaston hallintaa varten. Seuraavaksi on saattanut syntyä tarve hallita asiakkuuksia ja taloutta paremmin, mihin on hankittu joko uusi järjestelmä tai laajennettu olemassa olevaa. Tällä hetkellä voi todeta, että monet yritykset käyttävät erityyppisiä järjestelmiä, joiden keskinäinen integrointi on heikkoa tai sitä ei ole olemassa. Tämä tarkoittaa, että tiedonsiirto järjestelmien välillä on manuaalista, vaivalloista ja erittäin virhealtista.



Kuva 5. Tuotteen elinkaaren hallintaan liittyvät tietojärjestelmät.

Jos tarkastellaan valmistavaa teollisuutta, voidaan todeta, että yritykset käyttävät vähintään tuotannonohjausjärjestelmää. Mikäli kyseessä on suhteellisen pieniko-

5. Tuotetiedon hallinta

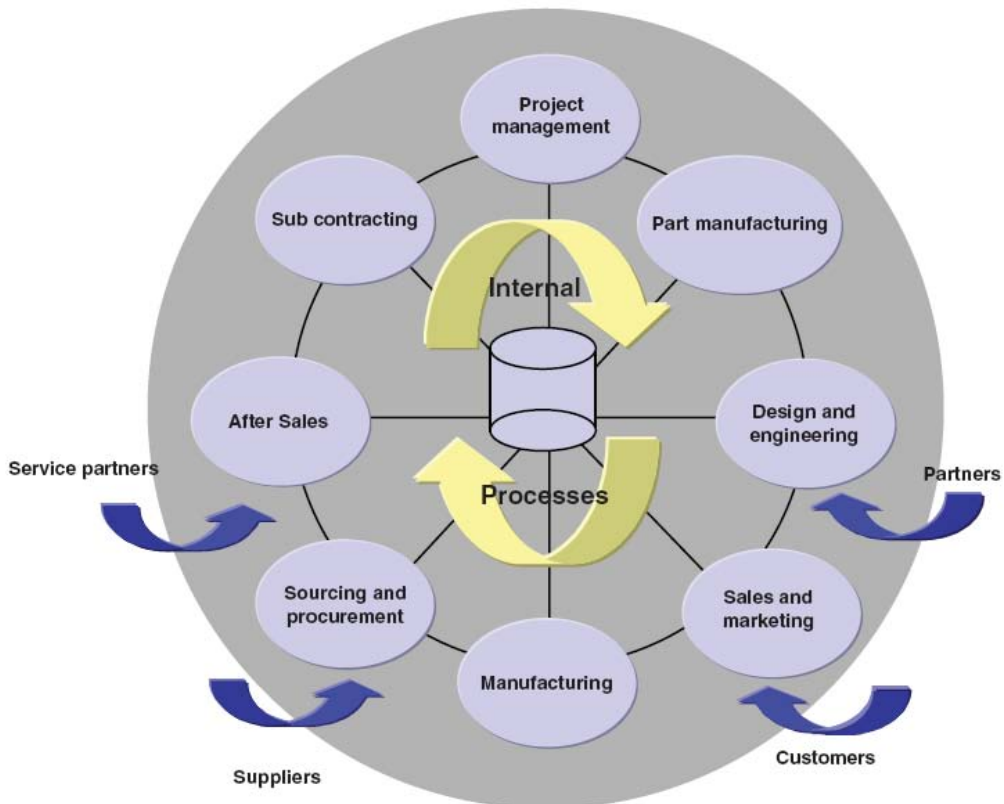
koinen yritys, samalla järjestelmällä hoidetaan monia muitakin osa-alueita, kuten taloutta ja asiakkuuden hallintaa. Joskus samaa järjestelmää sovelletaan jopa vähäisen suunnittelutiedon hallintaan, vaikka se ei siihen suoraan soveltuisikaan. Rajalliset resurssit sekä usein myös tiedon puute ovat pääsyyinä tähän. Isoimmilla yrityksillä on käytössä tyypillisesti useita eri järjestelmiä.

CAD-suunnitteluohjelmistoja käyttävien yritysten lähestymistapa tuotetiedon hallintaan poikkeaa yrityksistä, jotka eivät käytä CAD-ohjelmistoja. Pienet yritykset hallitsevat tuote- ja suunnittelutietoa esimerkiksi Microsoft Windowsin resurssienhallinnan avulla noudattaen yrityksen omia käytäntöjä. Mikäli keskitetty tiedonhallinta otetaan käyttöön, on kyse joko suppeasta suunnittelutietojen hallintajärjestelmästä (EDM) tai laajemmasta tuotetiedon hallintajärjestelmästä (PDM). EDM- järjestelmillä hallitaan suunnitteluun liittyvää tietoa (kuten 3D-mallit, piirustukset ja niihin liittyvät oheisdokumentit). Windowsin resurssienhallintaan verrattuna nämä järjestelmät tarjoavat monesti valmiudet rinnakkaiseen suunnitteluun ja muutosten hallintaan. Mallien editointi on kuitenkin mahdollista vain yhdelle käyttäjälle kerrallaan, mikä estää samanaikaiset muutokset. Lisäksi dokumenttien versioinnin hallinta varmistaa sen, että käytössä on aina uusin versio. Samalla tallentuu muutoshistoria, josta voidaan tarvittaessa jäljittää tehdyt muutokset.

Suunnittelutiedon hallinnan ongelmana on eri CAD-järjestelmien tietorakenteiden riippuvuuksien säilyttäminen. Tämä tekee muutosten hallinnan erittäin haastavaksi, mikäli muutokset joudutaan tekemään tiedostonimiin manuaalisesti. Jos muutoksia tehdään esimerkiksi tietyn CAD-osan tiedostonimeen, CAD-ohjelmistoihin integroidut EDM-järjestelmät päivittävät uuden nimen kaikkiin relaatioihin, jotka liittyvät kyseiseen osaan. Näin CAD-kokoonpanot ovat muutoksesta huolimatta käyttökelpoisia, kun relaatiot niihin kuuluviin osatiedostoihin säilyvät. EDM-järjestelmät hoitavat tarvittavat relaatiopäivitykset automaattisesti myös tiedoston tallennusosoitteen muuttuessa.

PDM-järjestelmät ovat käytännössä laajennettuja EDM-järjestelmiä. Niissä hallinnoidaan suunnittelutiedon lisäksi nimikkeitä, tuoterakenteita sekä muita asioita kuten tuotteen toimituskokonaisuuksia ja niiden rakenteita ("konekortti"). Jotkut PDM-järjestelmät sisältävät myös valmiita prosesseja esimerkiksi muutostenhallintaan sekä erilaisia hyväksymisprosesseja nimikkeen vapauttamiseksi tuotantoon. PLM-järjestelmät ovat edellä mainittuja laajempia järjestelmiä, joilla hallitaan PDM-järjestelmien toimintojen lisäksi tuotteen koko elinkaaren aikana syntyvää tietoa ja muutoksia. PLM-järjestelmällä hallitaan dokumenttien, nimikkeiden ja tuoterakenteiden lisäksi myös analyysi- ja testituloksia, komponenttien

ympäristötietoa, laatua, suunnitteluvaatimuksia, muutoksia, valmistusprosesseja, tuotteen suoritus tietoja, toimittajia, yrityksen erilaisia prosesseja sekä objektien, kuten nimikkeiden ja dokumenttien, elinkaarta (esimerkiksi: ”Luonnos”, ”Hyväksytty”, ”Tuotannossa”, ”Pois käytöstä”). Ideaalitapauksessa PLM-järjestelmä on laaja kokonaisuus, jossa hallitaan yrityksen tuotetieto keskeisesti ja johon muut järjestelmät (esim. ERP, CRM) on hyvin integroitu (ks. myös luku 6.8). Kuva 6 osoittaa, miten keskeinen ja kokonaisvaltainen PLM-järjestelmä ideaalitapauksessa on. (Sääksvuori & Immonen 2008)



Kuva 6. PLM-järjestelmä muodostaa laajan kokonaisuuden ja tukee näin erilaisia prosesseja sekä toimii keskeisenä tietopankkina. (Sääksvuori & Immonen 2008, s.14)

PLM ja Digitaalinen tuoteprosessi

Digitaalisen tuoteprosessin kannalta tuotetiedon ja etenkin tuotteen elinkaaren hallintajärjestelmät ovat erityisen tärkeitä. Tavoitteena on hallita kaikki tuotetie-

5. Tuotetiedon hallinta

to digitaalisesti koko tuoteprosessin läpi. Yritysverkostossa tiedonsiirto ja tuotetiedon reaaliaikainen saatavuus ovat menestymisen ehto, sillä väärinkäsitykset, vanhentuneet tiedostoversiot ja muut vastaavat seikat voivat aiheuttaa suuria ylimääräisiä kustannuksia. Pk-yritysten muodostamassa verkostossa on kuitenkin sovittava siitä, mikä osapuoli hallinnoi järjestelmää ja keskitettyä tuotetietoa. Tämä ei aina ole aivan yksiselitteinen ja vaatii tapauskohtaista tarkastelua. Järjestelmään liittyviä kustannuksia on myös tarkasteltava yhteistyöverkoston näkökulmasta. Muita vaikeuksia voi aiheutua muiden järjestelmien huonosta integroitavuudesta PLM-järjestelmiin. Integrointi saattaa vaatia manuaalisia välivaiheita, joita digitaalisessa tuoteprosessissa pyritään minimoimaan ja ideaalitapauksessa eliminoimaan kokonaan.

Henkilöstön ohjeistuksen merkitys

Kun yritys haluaa tehostaa toimintansa optimoimalla prosessejaan, on ensiarvoisen tärkeää, että henkilöstö on sitoutunut kehittämiseen ja ennen kaikkea tietoinen siitä, miten prosessit hoidetaan tehokkaasti. Yrityksen kannalta on olennaista, että koko henkilöstö toimii samojen periaatteiden ja käytäntöjen mukaisesti. Tämän vuoksi muutoksista tiedottaminen ja ohjeistuksen laatiminen on erityisen tärkeää. On syytä miettiä, mitä viestintäkanavia pitkin henkilöstöä informoidaan ja miten ohjeistuksia annetaan. Tämä riippuu vahvasti yrityksen toimialasta sekä koosta ja on siten selvitettävä tapauskohtaisesti. Samalla on varmistettava tiedonkulun kaksisuuntaisuutta, jotta ohjeistuksen ja tiedottamisen ohella henkilöstöltä saadaan kerättyä myös mahdolliset palautteet ja kehitysehdotukset.

Tuoteprosesseja ja tietojärjestelmiä varten on syytä laatia selkeitä ja yksityiskohtaisia ohjeita, jotka ovat kaikkien käyttäjien ulottuvilla. Sähköisessä muodossa olevat ohjeet pystytään päivittämään helpommin ja nopeammin kuin paperille painetut. Tällaisia ohjeistuksia on syytä laatia myös silloin, kun käytössä ei ole esim. PLM-järjestelmää ja tuotetieto hallitaan vaikkapa verkkolevyllä kansiorakenteen avulla. Silloin käyttäjillä tulee olla yhdenmukaiset tavat nimetä tiedostoja ja merkitä versiointitunnukset niihin. Tietojärjestelmät hoitavat tämän puolen omaa logiikkaa käyttäen, mutta koska verkkolevyille tiedostot voidaan tallentaa millä nimillä tahansa, tiedon löytäminen ja uudelleen käyttäminen on haasteellista. Monesti ajaudutaan siihen tilanteeseen, että tiedostot hallitaankin osittain paikallisesti omilla työasemilla, mikä vaikeuttaa verkostomaista toimintaa ja tehokkaasta tuotetiedon käyttöä. (Sääksvuori & Immonen 2002, s.18)

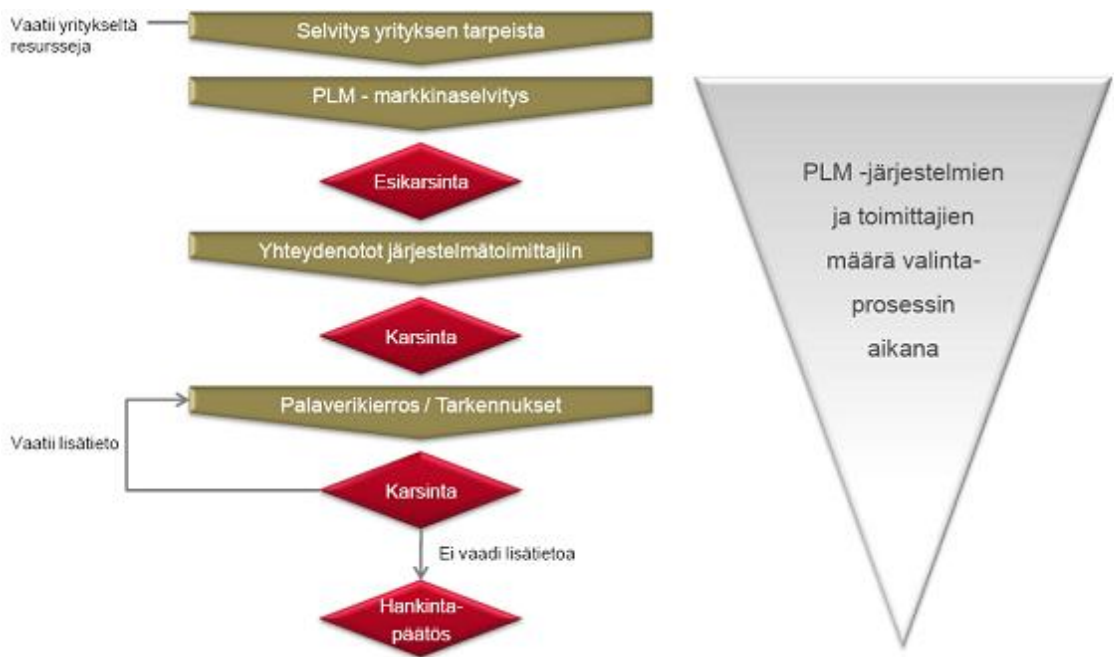
5.4 Tuotetiedon hallintajärjestelmän valintaprosessi

Tuotetiedon hallintajärjestelmiin ja niiden valintaan liittyvä vaikeus on, että jokainen järjestelmä lähestyy tuotetiedonhallintaa hieman eri näkökulmasta ja tarjoaa siksi jonkin verran erilaisia ominaisuuksia ja ratkaisuja. Myös järjestelmien peruslogiikka ja arkkitehtuuri voivat poiketa toisistaan paljonkin. Lisäksi tutkimuksissa tuli esille, että valintaa ei voida tehdä puhtaasti objektiivisin perustein. Koska järjestelmien lähestymistavat ovat erilaisia, kaikkia ominaisuuksia ei voida arvioida suoraan, vaan joudutaan turvautumaan subjektiiviseen arviointiin. Tällöin internetin, referenssien sekä myyjän kautta muodostunut mielikuva vaikuttaa päätökseen. Tässä vaiheessa on hyvä korostaa, että PLM-järjestelmän hankinnassa itse tietojärjestelmän ominaisuudet kokonaisuudessaan ovat ratkaisevia tekijöitä, mutta lähes yhtä tärkeitä ovat järjestelmätoimittajan kyky toteuttaa käyttöötoprojekti kustannustehokkaasti tarjota sekä hyvä tuotetuksi järjestelmän käyttöaikana. Tutkimuksessa kyseinen problematiikka tuli hyvin esille, koska projektin resurssit PLM-järjestelmän hankintaan olivat varsin rajalliset ja koulutus sekä käyttöönotto jäivät suunniteltua vähemmälle. Käyttöönottoprojektin edetessä kattavammasta neuvonnasta ja koulutuksesta olisi ollut hyötyä.

Vastaavia kokemuksia on myös projektiin osallistuneilla yrityksillä. Eräässä yrityksessä järjestelmä oli todettu käyttöönottovaiheessa hyväksi, mutta järjestelmätoimittajan tuki oli riittämätöntä, mikä kasvatti yrityksen omaa työmäärää arvioitua suuremmaksi. Toisessa yrityksessä käyttöönottoprojektia puolestaan kehuttiin onnistuneimmaksi IT-projektiksi, mitä yrityksessä koskaan oli tehty. Kyseinen yritys osti koko käyttöönottoprojektin PLM-järjestelmän toimittajalta, joka vastasi projektin läpiviennistä. Nämä esimerkit osoittavat sen, että kiinnostavan toimittajan toimintatavoista ja luotettavuudesta kannattaa ottaa mahdollisimman hyvin selvää esimerkiksi muilta yrityksiltä, jotka jo käyttävät saman toimittajan järjestelmiä.

PLM-järjestelmän käyttöönotto ja vaihto ovat sen verran isoja projekteja, että yhteistyö järjestelmätoimittajan kanssa tulee todennäköisesti olemaan pitkäaikaista. Tiedonkeruuvaiheessa on lisäksi syytä selvittää mahdollinen tarve hankkia tietokantasovellus. PLM-järjestelmät ovat lähes poikkeuksetta tietokantapohjaisia. Tietyt tietokantasovellukset ovat varsin kalliita ja nostavat siten hankintakustannuksia entisestään. Jotkut PLM-järjestelmät taas mahdollistavat ilmaisen tietokantaohjelmiston käytön, joka on pk-yrityksille usein riittävä. PLM-järjestelmän valintaprosessi on esitetty yleisellä tasolla kuvassa 7.

5. Tuotetiedon hallinta



Kuva 7. PLM-järjestelmän valintaprosessi yleisellä tasolla.

Sääksvuori & Immonen (2008, s. 73) ovat listanneet yleisellä tasolla seuraavat seikat, jotka PLM-järjestelmän hankintaa suunnittelevan yrityksen olisi otettava huomioon:

1. Kyseisen PLM-järjestelmän markkinat/toimiala, järjestelmän kehityssuunnitelma sekä järjestelmäkehittäjän tulevaisuuden visiot.
2. Järjestelmän toiminnalliset sekä tekniset ominaisuudet: sen rajaukset, arkkitehtuuri ja skaalattavuus.
3. Integraation taso; mahdollisuus käyttää vakioteknologioita, kuten xml:ää.
4. Käyttöönottoon tarvittava työmäärä ja aika; oma vs. ulkopuolinen työ.
5. Ylläpidon helppous.
6. Helppokäyttöisyys ja käyttömukavuus.
7. Saatavana oleva tuki ja muut toimittajapalvelut, kuten käyttäjäkonferenssit.
8. Kokonaishinta: työ, lisenssit, ylläpito, uudet kolmannen osapuolen ohjelmistot, käyttökoulutus, laitteisto.

9. Olemassa olevat käyttökohteet; referenssikäynnit järjestelmää käyttävissä organisaatioissa.
10. Järjestelmän tuki erikielisille nimikkeiden ja dokumenttien hallinnalle.
11. Käyttöliittymän kieliversiot.

MoniDigi-projektissa PLM-järjestelmän valintaprosessi käynnistettiin kattavalla tietohaulla. Aluksi yritettiin löytää mahdollisimman paljon tietoa PLM-järjestelmistä ja niiden referensseistä. Tietoa etsittiin internetistä, artikkeleista ja muista julkaisuista sekä haastattelemalla projektiin osallistuvia yrityksiä. Samalla mietittiin, mitkä ovat projektin tarpeet ja vaatimukset PLM-järjestelmälle. Työkaluna käytettiin matriisia, johon kirjattiin ylös, miten eri järjestelmät vastaavat asetettuja vaatimuksia. Itse valintakriteerit kirjattiin tiedonhankinnan edessä taulukkomuotoon ja ne painotettiin tärkeysasteen mukaisesti. Vaatimukset on esitetty taulukossa 2.

On tärkeää, että yritys harkitsee tarkkaan, mitkä PLM-järjestelmän ominaisuudet ovat hankintahetkellä tärkeitä ja mitkä ovat tulevaisuudessa todennäköisesti esille tulevat tarpeet. Ei ole suositeltavaa hankkia kerralla laaja järjestelmä, vaan on parempi edetä todellisten tarpeiden ja resurssien mukaisesti. Näin käyttöönottoprojekti sekä järjestelmän hankinta hallitaan paremmin ja henkilökunnan koulutustarpeet pysyvät kohtuullisina. Kun yrityksen tarpeet on selvitetty, matriisin käyttö tiedonhakuvaiheessa mahdollistaa sen, että järjestelmistä saatujen tietojen puutteet ovat hyvin havaittavissa. On selvää, että eri toimittajat kertovat internet-sivuillaan laajalti järjestelmiensä hyvistä ominaisuuksista, mutta niiden heikot kohdat ja puutteet eivät tule yhtä hyvin esille. Kun eri järjestelmien tiedot kirjataan matriisimuotoon, voidaan helpommin täsmentää puuttuvat tiedot.

Tutkimuksessa tuli esille järjestelmätoimittajien tapa kertoa tuotteistaan hyvin eri tavoilla, mikä vaikeutti järjestelmien keskinäistä vertailua. Mitä tarkemmin tiedonhaku ja mahdolliset tarkentavat kyselyt tehdään, sitä paremmin järjestelmän hankintaa pystytään perustelemaan yrityksessä niille tahoille ja päätöksen tekijöille, jotka eivät ole osallistuneet aktiivisesti valintaprosessiin.

Tiedonkeruuvaiheen jälkeen MoniDigi-projektissa otettiin yhteyttä kuuteen eri toimittajaan. Yksi toimittajista totesi ensimmäisen yhteydenoton aikana, ettei heidän järjestelmänsä sovellu projektin käyttöön. Viiden muun järjestelmätoimittajan kanssa sovittiin erilliset palaverit, joiden tavoitteena oli perehtyä tarkemmin kuhunkin järjestelmään ja tarkentaa puuttuvat tiedot valintamatriisia varten. Tavoitteena oli myös tarkentaa hankittavan järjestelmän ominaisuudet ja toimituksen sisältö siten, että järjestelmätoimittaja pystyy tekemään tarjouksen

5. Tuotetiedon hallinta

järjestelmästä sekä sen käyttöönotosta ja koulutuksesta. Tarjoukset arvioitiin taulukossa 2 olevien kriteerien mukaisesti ja järjestelmien sisältöä tarkennettiin vielä tarvittaessa.

Lopuksi päätettiin hankkia Dassault Systèmesin ENOVIA V6. Järjestelmä valittiin kaikkien tässä julkaisussa esitettyjen tietojen ja arviointien mukaisesti. Valintaan vaikuttivat merkittävästi tiedot ja vaikutteet, jotka valintaryhmä sai järjestelmätoimittajien esityksistä. Projektiryhmä arvioi keskenään eri vaihtoehtojen sopivuutta sekä projektin että pk-yritysten näkökulmasta. Voidaan todeta, että valinnassa ei ollut kysymys pelkästään hankintahinnasta, vaan tärkeimpiä valintaperusteita olivat myös järjestelmän ominaisuudet ja toiminnallisuus. Järjestelmän hankintahinnan yläraja määräytyi hankintaan varatun rahoituksen mukaisesti.

Taulukko 2. MoniDigi-projektissa käytetyt kriteerit PLM-järjestelmän valinnassa.

	Vaatus	Paino- tus	Selitys
1	Helppokäyttöisyys	3	Käyttöliittymän on oltava mahdollisimman yksinkertainen ja selkeä, jotta perusasiat voi oppia nopeasti.
2	Out-of-the-box-asennus	3	Asennuksen ja konfiguroinnin on oltava nopeaa ja yksinkertaista ilman suurempaa räätälöintiä, jotta käyttöönottohinta pysyy alhaisena ja ohjelmiston päivitettävyyden paranee.
3	Käyttäjän räätälöintimahdollisuus	3	Ylläpitohenkilöllä on oltava mahdollisuus muokata rajoitetusti käyttöliittymää ilman koodausta.
4	Toimitusaika	2	Järjestelmällä on oltava lyhyt toimitusaika, jotta asennus ja koulutus saadaan ajoissa.
5	Laajennettavuus	2	Sekä käyttäjämäärää että järjestelmän toiminnallisuutta on voitava laajentaa. Pk-yritykset voivat ottaa aluksi käyttöön kevyen version, jota voidaan laajentaa tarvittaessa.
6	Tuki yritysverkostokäytölle	3	On oltava mahdollisuus toimia yritysverkostossa; lisensointipolitiikka ei saa olla esteenä ja tiedonsiirtomahdollisuuksien tulee olla monipuoliset. Internetin hyödyntäminen on ehdoton vaatimus.

7	Soveltuvuus kansainväliseen toimintaan	2	Eri kielille ja aikavyöhykkeille on oltava tuki. Tarvitaan hajautetun toiminnan tukeva järjestelmäarkkitehtuuri, joka on nopea ja joustava ja luo mahdollisuuden rinnakkaiseen toimintaan samojen tuotteiden ympärillä.
8	Selainpohjainen järjestelmä	3	Ei vaadita konekohtaisia asennuksia, vaan ainoastaan päätelaite Internet-liittymällä (tietokone, älypuhelin, jne.).
9	Myyntikonfiguraattori	3	On oltava mahdollisuus tehdä selainpohjainen myyntikonfiguraattori, jolla tuoterakenne voidaan määritellä ja joka on suorassa yhteydessä järjestelmään.
10	Tuotekonfiguraattori	3	On oltava tuotekonfiguraattori, jolla määritellään yksilöllinen tuoterakenne, jota voidaan hyödyntää valmistuksessa/kokoonpanossa.
11	CAD-integraatio CATIA V5	3	On oltava, sillä Oulun yliopiston konekniikan osastolla on käytössä kyseinen CAD-ohjelmisto.
12	CAD-integraatio muihin CAD-ohjelmistoihin	3	On oltava integraatiomahdollisuus mahdollisimman moniin CAD-ohjelmistoihin, koska otetaan huomioon pk-yritykset, joilla voi olla erilaisia suunnitteluohjelmistoja. Myös yritysverkostossa voi olla erilaisia ohjelmistoja käytössä.
13	Integraatio mm. sähkö-/hydrauliikkasuunnitteluohjelmistoihin	2	Olisi toivottavaa, että integraatio löytyy myös muihin front-end-ohjelmistoihin kuin CAD-ohjelmistoihin, jotta järjestelmää voidaan hyödyntää tuotesuunnittelun eri vaiheissa (simulointi, analyysi, hydrauliikka- ja sähkösuunnittelu).
14	Toimiva rajapinta mm. ERP:hen	1	Tämän projektin kannalta ei kovin tärkeä vaatimus, mutta yrityksille kyllä; nimikkeet, tuoterakenteet jne. on oltava siirrettävissä automaattisesti ja yksikäsitteisesti toiminnanohjaus- ja muihin business-järjestelmiin (back-end-integration).

5. Tuotetiedon hallinta

15	Kustannuslaskenta	1	Kustannukset on saatava jo suunnitteluvaiheessa esille, jotta suunnittelija näkee päätöksensä vaikutukset tuotekustannuksiin.
16	Yksinkertainen ohjelmistopäivitettyvyys	1	On oltava yksinkertainen ja mahdollisuuksien mukaan yhdestä paikasta suoritettava toimenpide, joka ei vaadi uudelleenkonfigurointia ja räätälöintiä.
17	Hinta	3	Pk-yritysten pienistä taloudellisista ja henkilöresursseista johtuen lisenssi-, ylläpito- ja käyttöönottokustannusten on oltava mahdollisimman pienet.
18	Turvallisuus /tietoturva	1	Järjestelmän on oltava turvallinen myös yhteistyökäytössä: tietoturva, virustentorjunta, vakoiluohjelmat jne. Käyttöoikeuksien määrittelyn tulee olla selkeä ja toimiva (riittävän tarkka, mutta ei liian yksityiskohtainen).
19	Turvallisuus/toimivuus	1	Järjestelmää ei saa liian helposti sekaisin (esim. ei voi luoda useita samoja nimikkeitä).
20	Suunnitteluprosessin tuki (toimintamalli määritelty)	1	Esim. versionhallinta, taustamateriaalin hallinta; businessprosessien toimintamallien tulee olla implementoitu järjestelmään, jotta pk-yritys saa tehokkaan ja toimivan toimintamallin käyttöönsä.
21	Tuki ilmaiselle tietokannalle	1	On oltava tuki ilmaisille tietokantaohjelmistoille, ettei tarvitse hankkia erillisiä lisenssejä, jotka aiheuttavat lisäkustannuksia.
22	Kotimaisuus	1	Järjestelmän tulee olla mielellään kotimainen, jolla tuetaan tuotteen käytettävyyttä Suomessa.
23	Koulutuksen laajuus suhteessa lisenssikustannuksiin	1	Koulutuksen laajuuden ja hinnan on oltava sopivassa suhteessa lisenssihankintoihin ja ylläpitomaksuihin nähden. Järjestelmän tehokkaan käyttöönoton kannalta tarvitaan mahdollisimman paljon koulutusta.
24	Tekninen dokumentaatio	3	On oltava riittävän yksityiskohtainen ja selkeä dokumentaatio, joka mah-

			dollistaa itseopiskelun ja omatoimisen ongelmaratkaisun.
25	Lisensointimalli projektinjälkeiselle opetuskäytölle	1	On oltava sopiva lisensointimalli, jolla yliopisto voi halutessaan ottaa järjestelmän projektin jälkeen opetuskäyttöön.
26	Arvio ylläpitomaksuista projektin jälkeiselle käytölle	1	Olisi hyvä, mikäli toimittaja voisi antaa arvion lisenssimaksuille projektin jälkeen. Tarkkuudeltaan riittäisi esim. se, että järjestelmää voidaan käyttää jatkossa suunnilleen samansuuruisilla ylläpitomaksuilla.
27	Toimittajan referenssit	2	On oltava uskottavuutta tuovia referenssejä sekä tunnettujen suuryritysten että pk-yritysten osalta. On oltava koko skaalaa edustettuna.
28	Selkeä ja yksinkertainen tuoterakenne	1	On oltava selkeä tuoterakenne. Muutama moduulin sisältävä tuoteskaala helpottaa asiakasta ymmärtämään, mitä hän tarvitsee ja mitä missäkin moduulissa on. Liian tarkka jaottelu ja laaja valikoima sekoittavat ja lisäävät epävarmuutta.
29	Järjestelmän soveltuvuus pk-yrityksille	3	On oltava ominaisuuksia, jotka tukevat pk-yritysten toimintamalleja ja työkaluja. Kevyt ja edullinen järjestelmävariantti käyttöönottokonseptilla on plussaa.
30	Järjestelmän soveltuvuus suur-yrityksille	1	Kasvuyritykset huomioon ottaen järjestelmän on kyettävä palvelemaan myös globaaleja isoja yrityksiä, ettei kasvuyritykselle tule tilannetta, jossa joudutaan vaihtamaan koko järjestelmä.
31	Järjestelmän globaali käyttö yrityksissä	2	Otetaan huomioon, miten yleisesti järjestelmä on yritysmaailmassa käytössä. Otetaan huomioon myös järjestelmäkehitykseen käytetyt resurssit ja rahasummat sekä järjestelmän tulevaisuuden kehitysnäkymät.

6. Digitaalinen tuoteprosessi – uusi konsepti

Tuoteprosessia digitalisoimalla voidaan tehostaa teknologiayritysten tuotekehitystä ja tuotantoa sekä parantaa yritysverkoston yhteistoimintaa. Saumattomasti toimiva yritysverkosto on tärkeä erityisesti monitekniisiä tuotteita valmistaville yrityksille, koska tuotteiden suunnittelu edellyttää osaamista usealla tekniikan osa-alueella. Digitaalisen tuotetiedon yleistyessä mallinnuksesta ja simuloinnista on tullut yhä luontevampi osa suunnitteluprosessia. Simulointimallien hallintaan ja simulointituloksien jakamiseen voidaan käyttää samoja järjestelmiä, joilla muutoinkin hallitaan tuotemalliin perustuvaa tuotekehitystä. Suojattujen verkko-yhteyksien kautta tietoa voidaan jakaa turvallisesti yhteistyöyrityksille, joilla on käyttöoikeus vain rajattuun, yrityksen vastuualuetta vastaavaan osaan tuotetietoa.

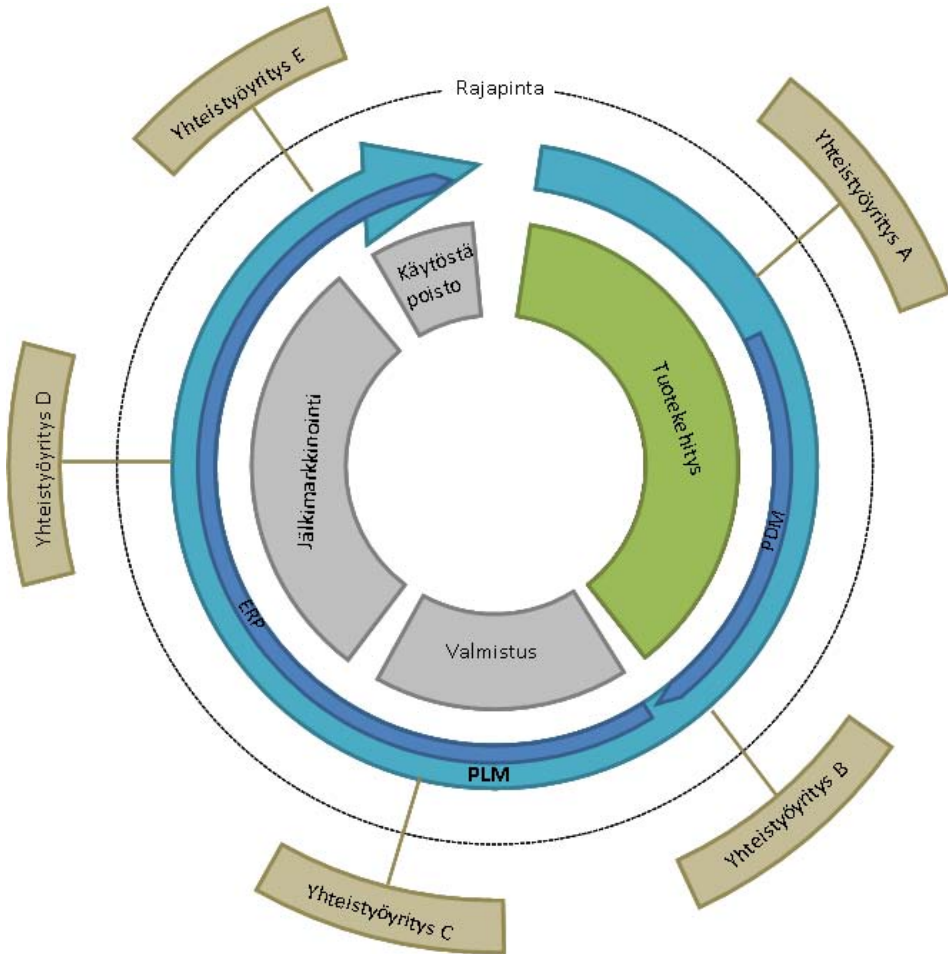
Tuotteen elinkaaren ensimmäinen vaihe on tuotekehitys, joka yleensä käynnistyy, kun markkinoilla havaitaan merkittävä asiakastarve tai kysyntä uudelle tuotteelle. Tuotekehityksen alkuvaiheessa selvitetään asiakastarpeet ja markkinat, tuotteen yleiset vaatimukset sekä tuotteen toteuttamiseen tarvittavat teknologiat. Jos tuote on teknologisesti toteutettavissa ja tuotekehitys arvioidaan markkinoihin nähden kannattavaksi, tuote suunnitellaan, toteutetaan ja testataan. Digitaalisessa tuoteprosessissa suunnittelu- ja testausvaiheissa hyödynnetään mahdollisimman paljon virtuaalisuunnittelun työkaluja, ja todellisia prototyyppejä rakennetaan vain siinä tapauksessa, jos virtuaaliprototyyppi arvioidaan liian epäluotettavaksi. Tuotekehitysvaihe katsotaan päättyneeksi, kun tuotteesta on olemassa ensimmäinen kaupallinen versio, joka täyttää sekä asiakkaan että standardien ja normien tuotteelle asettamat vaatimukset.

Digitaalisessa tuoteprosessissa tuotteen valmistusvaihe aloitetaan tuotteen valmistamiseen käytettävän tuotantojärjestelmän tietokoneavusteisella suunnittelulla ja simuloinnilla. Tuotantojärjestelmän virtuaalisuunnittelun tavoitteena on

varmistaa mahdollisimman kustannustehokas valmistus lyhentämällä tuotannon ylösajovaihetta sekä optimoimalla virtuaalisen tehdasmallin materiaalivirtoja (Lehtonen 2006, Heilala ym. 2008). Valmistuksen virtuaalisuunnitteluvaiheessa voidaan esimerkiksi generoida valmiiksi koneistettavien kappaleiden konekohtaiset työstöradat sekä optimoida kiinnityksien ja työkalujen vaihdot ilman tuotannossa olevien tuotantokoneiden pysäyttämistä. Kun tuotteen valmistusprosessi on suunniteltu etukäteen, uusi tuote voidaan valmistaa tilausperusteisesti ja mahdollisesti asiakaskohtaisesti räätälöitynä mahdollisimman kustannustehokkaasti.

Tuotteen elinkaaren hallintajärjestelmä ja tietoverkkoyhteydet ovat verkottuneen yritystoiminnan ja rinnakkaisen tuotekehityksen teknologinen perusta. PLM-järjestelmässä hallitaan ja pidetään tuoterakenne ajan tasalla. Tietoverkkoyhteydet takaavat sen, että suunnittelusta vastaavat yhteistyöyritykset saavat niiden vastuualueeseen kuuluvan tuotetiedon reaaliajassa.

6. Digitaalinen tuoteprosessi – uusi konsepti



Kuva 8. Tuotteen elinkaaren päävaiheet ja niiden hallintaan liittyvät tietojärjestelmät.

Kuvassa 8 on esitetty tuotteen elinkaaren päävaiheet ja niiden hallintaan liittyvät tietojärjestelmät. Tuotteen elinkaari on kuvattu ympyräksi, koska tuotteen ensimmäisen elinkaaren päättyessä osa tuotteesta voidaan kierrättää tai käyttää uudestaan tai koko tuote voidaan päivittää (modernisoida) ensimmäisen käyttötarkoituksen jälkeen toisenlaiseen tehtävään. Olemassa olevan koneen modernisointiin saattavat johtaa myös kunnossapidolliset syyt tai tiukentuneet ympäristö- ja turvallisuusvaatimukset (Malm & Hämäläinen 2006). Ympyrä kuvaa osaltaan myös suunnittelutiedon kierrättämistä, koska käytännössä uuden tuotteen suunnittelu aloitetaan täysin puhtaalta pöydältä vain harvoin. Usein kehitysprosessien aluksi kartoitetaan, voidaanko uuden tuotteen yhteydessä käyttää

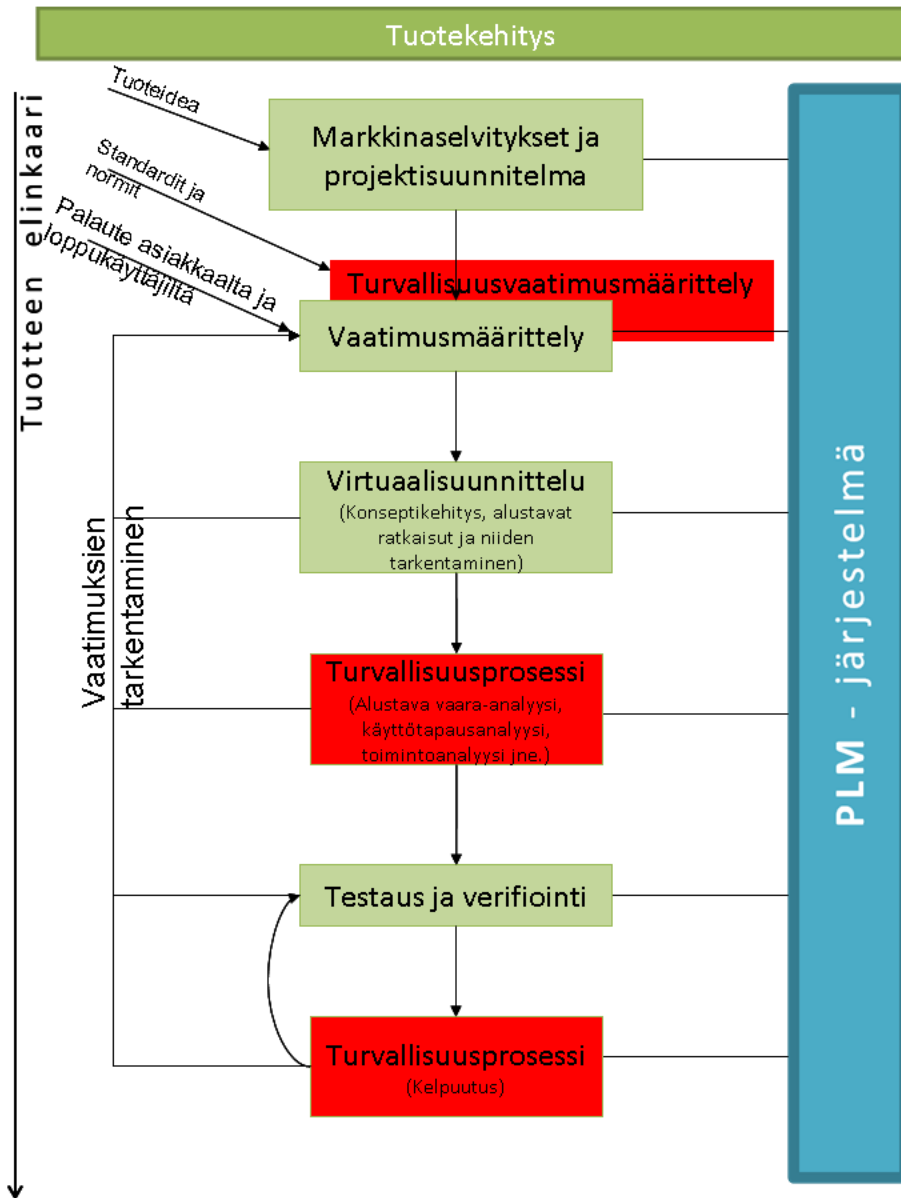
jotain jo valmista osaa tai kokonaisuutta. Näin ollen ympyrä kuvaa nykyaikaisen tuotteen elinkaarta paremmin kuin lineaarisesti etenevä päävaihejana. Kuvassa 8 esitetty PDM-järjestelmä ei tyypillisesti ole erillinen PLM-järjestelmästä, vaan mikäli yrityksessä on käytössä PLM-järjestelmä, se sisältää myös PDM-järjestelmän. Kuvan tarkoitus on havainnollistaa, mihin kohtaan tuotteen elinkaarta PDM-järjestelmä suunnilleen sijoittuu. Kehitetyssä konseptissa PLM-järjestelmä on kuitenkin olennaisessa roolissa, sillä tuotetietoa on hallittava koko elinkaaren ajan.

Kuvassa 8 on esitetty kaikki elinkaaren vaiheet, mutta MoniDigi-projektissa on keskitytty lähinnä tuotekehitysvaiheen virtuaalisuunnitteluun, virtuaaliseen testaukseen ja verifiointiin sekä PLM-järjestelmän käyttöön. Elinkaarimallin muiden vaiheiden esitykset perustuvat pitkälti lähdekirjallisuuteen. PLM-järjestelmän ominaisuuksista ja mahdollisuuksista pyritään kertomaan tässä julkaisussa aina niissä kohdin, missä se on tarkoituksenmukaista.

6.1 Tuotekehitys

Tuotekehitysvaiheessa asiakkaan määrittelemistä vaatimuksista, piirteistä, tarpeista tai ongelmista suunnitellaan ja toteutetaan vaatimukset mahdollisimman hyvin täyttävä ja kustannustehokas tuote. Asiakastarpeen lisäksi suurimman osan tuotteen vaatimuksista muodostavat standardit ja normit, kuten esimerkiksi koneasetus ja -direktiivi sekä turvallisuusstandardit (Hietikko ym. 2009). Kuvassa 9 tuotekehitys on esitetty iteratiivisena prosessina, jonka viisi päävaihetta ovat markkinaselvitys ja projektisuunnitelma, vaatimusmäärittely (sisältäen turvallisuusvaatimukset), virtuaalisuunnittelu, turvallisuusprosessi sekä testaus ja verifiointi. Vaiheet seuraavat ja ovat riippuvaisia toisistaan: seuraava vaihe voi käynnistyä vasta edellisen vaiheen valmistuttua tai päästyä siihen pisteeseen, että tarvittavat lähtötiedot seuraavan vaiheen suunnittelulle ovat olemassa.

6. Digitaalinen tuotekehitys – uusi konsepti



Kuva 9. Tuotekehityksen vaiheet.

Markkinaselvitykset ja projektisuunnitelma sekä vaatimusmäärittelyt tuottavat arvokasta lähtötietoa. Näiden valmistelevien vaiheiden jälkeen tuotekehitysprojektin tavoitteen pitäisi olla mahdollisimman selvä kaikille osapuolille, kuten

suunnittelijoille, mallintajille, myynnille, johdolle, asiakkaalle jne. (Leppävuori ym. 2009). Kun tuotekehityksellä on selkeä tavoite ja se on kaikkien osapuolien tiedossa, virtuaalisuunnittelu on helpompi kohdistaa olennaiseen. Virtuaalisuunnittelun työkaluilla voidaan esimerkiksi varmistaa jo konseptisuunnittelussa, että tuote vastaa asiakkaan tarpeita, ja korjata suurin osa karkeista suunnitteluvirheistä, ennen kuin rakennetaan kalliita fyysisiä prototyyppkejä. Virtuaaliprototyypit ovat oiva vuorovaikutusväline asiakkaan ja suunnittelun välillä.

Virtuaalisuunnittelu, turvallisuusprosessi sekä testaus ja verifiointi ovat kaikki iteratiivisia prosesseja. Prosessien iteraatiokierroksien lukumäärää ei ole rajattu. Uuden tuotteen vaatimuksia sekä turvallisuutta arvioidaan ja kehitetään jatkuvasti koko tuotekehitysprojektin ajan. Jatkuva riskinarviointi ja katselmukset ovat tärkeitä, jotta tuotteen suunnittelussa ja toteutuksessa tulee otettua huomioon standardien ja normien tuotteelle asettamat vaatimukset. Suunnittelu, simuloinnit, riskienarvioinnit ja katselmukset ovat lähes hyödyttömiä, jos niissä syntyy tietoa ei hallita kunnolla. Eri vaiheissa syntyvä tieto on myöhemmin parhaiten hyödynnettävissä, kun sen riippuvuudet tuotteen osaan tai ominaisuuteen voidaan esittää tarkasti. Esimerkiksi puutteet turvallisuusprosessissa voivat pahimmassa tapauksessa johtaa siihen, että yritys toteuttaa toiminnallisesti täysin moitteettoman tuotteen, joka ei kuitenkaan täytä sille asetettuja turvallisuusvaatimuksia eikä siten ole kelvoinen markkinoille. Lähes valmiin tuotteen muuttaminen turvallisuusvaatimusten mukaiseksi jälkikäteen saattaa tulla erittäin kalliiksi. Jotta kyseinen tilanne estettäisiin ja välttyttäisiin kalliilta muutostöiltä, turvallisuusprosessin on oltava luonteeltaan samalla tavalla iteratiivinen kuin suunnitteluprosessikin: turvallisuusvaatimukset tarkentuvat suunnittelun edetessä sitä mukaa, kun suunnittelutieto lisääntyy (Hietikko ym. 2009).

Monitekniisyys asettaa lisähaasteen tuotekehitykselle, koska menestyksekkäs suunnittelu vaatii tuotteesta riippuen osaamista useilta tekniikan osa-alueilta, kuten mekaniikkasuunnittelusta, tietotekniikasta, elektroniikasta, hydraulikasta ja pneumatiikasta. Työkoneiden tapauksessa suunnittelijoiden on lisäksi tunnettava koneen työprosessi, käyttöympäristö ja koneeseen liittyvät turvallisuusmääräykset. Suunnittelun moniulotteisuuden takia suunnitteluprosessi joudutaan yleensä hajauttamaan eri tiimeihin, jotka ratkaisevat osaongelmia ja suunnittelevat osajärjestelmiä. Suunnittelun hajauttamisen tavoite on edistää rinnakkaista suunnittelua sekä parantaa laatua jakamalla suunnitteluvastuuta niille, jotka hallitsevat teknisen osa-alueen parhaiten. Hajauttaminen lisää kuitenkin tarvetta tehokkaille viestintäratkaisuille varsinkin silloin, jos tiimit ovat maantieteellisesti erillään tai työskentelevät eri organisaatioissa. Tällaisessa verkottuneessa tuo-

6. Digitaalinen tuoteprosessi – uusi konsepti

tekehityksessä sujuva tiedonkulku suunnittelutiimien välillä on ensiarvoisen tärkeää.

Moniteknisten tuotteiden tuotekehitystä hankaloittaa myös se tosiasia, että useita teknologioita sisältävien tuotteiden prototyyppi on huomattavan kallista ja riskialtista. Monitekninen prototyyppi vaatii toimiakseen lukuisia yhteensopivia komponentteja, kuten rakenneosia, antureita, toimilaitteita, I/O-kortteja, laskeinta-alustoja ja ohjelmistoa. Kaikkien näiden komponenttien suunnittelu ja valmistus sekä valinta ja hankinta vievät aikaa ja rahaa lisäten tuotekehityksen kustannuksia. Koska valmistuskustannukset prototyyppiä kohden ovat korkeat, todellisten prototyyppien määrää tuotekehityksen aikana joudutaan minimoimaan. Prototyyppien vähäinen rakentaminen on toisaalta myös riski, koska tiimien suunnittelutyön ja osajärjestelmien yhteensopivuudesta ei ole mitään takeita, ennen kuin ne yhdistetään ja testataan kokonaisuutena. Hajautetun suunnittelun laatu, osajärjestelmien yhteensopivuus ja tuotteen oikea toiminta olisi kyettävä varmistamaan kustannustehokkaasti koko tuotekehitysprojektin ajan.

Nykyaikaisessa globaalissa kilpailussa menestyminen edellyttää tuotteilta teknisen yliveraisuuden lisäksi myös hyvää muotoilua. Muotoilun avulla on mahdollista parantaa tuotteiden visualisuutta sekä ottaa huomioon käyttäjän ja ympäristön näkökulma. Silti konepajateollisuuden tuotekehitys pyörii enimmäkseen teknisten ominaisuuksien ympärillä, ja teolliset muotoilijat otetaan tyypillisesti mukaan vasta tuotekehityksen loppuvaiheessa, jolloin muotoilun parantamiseksi voidaan tehdä varsin vähän – esimerkiksi viime hetken kulmanpyöristyksiä tai värin valintaa. Toisaalta muotoilijoita arvostellaan siitä, etteivät he ymmärrä esimerkiksi valmistustekniikan rajoituksia. Muotoilijoiden ja insinöörien välistä yhteistyötä olisi mahdollista tehostaa, jos muotoilua käytettäisiin yhtenä rinnakkaisen suunnittelun resurssina tuotekehityksen alusta pitäen. (Ekman & Säski-lahti 2002.) Nykyaikaiset virtuaalisuunnittelutyökalut antavat rinnakkaiseen suunnitteluun aikaisempaa paremman mahdollisuuden, koska muotoilijat ja insinöörit voivat hyödyntää työssään samoja digitaalisia malleja.

6.1.1 Markkinaselvitys ja projektisuunnitelma

Asiakastarvetta voidaan pitää nykypäivänä tuotekehitysprosessin yleisimpänä lähtökohtana, jonka tarkoitus on varmistaa, että yrityksen tuotekehitys perustuu olemassa olevaan asiakastarpeeseen (LUT 2011). Siinä asiakas, markkinointi tai myynti määrittelee ongelman, jonka kehitettävä tuote ratkaisee. Asiakastarve pitää sisällään ongelmanasettelun lisäksi tuotteen ominaisuuksia, jotka tarkenta-

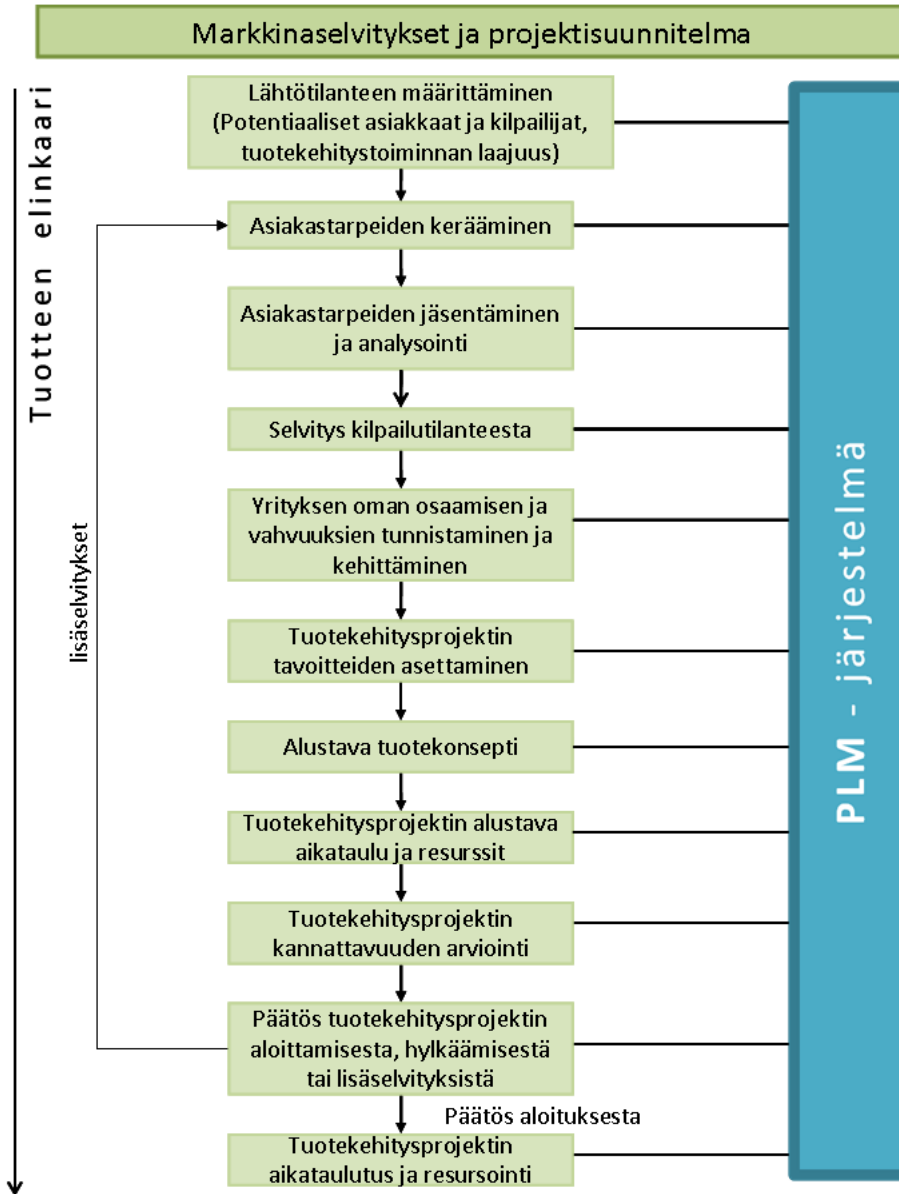
vat tuotteelle asetettavia vaatimuksia. Toisin sanoen asiakastarve on dokumentti, johon kerätään kehitettävän tuotteen alustavat vaatimukset, joihin myös tähdätään. Dokumenttiin kirjataan myös tiedot tuotteen toimintaympäristöstä ja määryksistä sekä standardeista, jotka tuotteen tulee täyttää.

Kun tuotekehitysprosessia suunnitteleva yritys tekee asiakastarvekartoituksen, se voidaan jakaa karkeasti kuuteen päävaiheeseen (LUT 2011):

1. lähtötilanteen määrittäminen
2. tiedon kerääminen asiakkaan tarpeista
3. asiakastiedon jäsentäminen ja analysointi
4. tiedon kerääminen yrityksen kilpailutilanteesta
5. tuotetta koskevien tavoitteiden asettaminen
6. tavoitteiden saavuttamiseen ohjaaminen.

Kuvassa 10 on esitetty markkinaselvityksen kulkukaavio asiakastarpeen keräämisestä tuotekehitysprosessin käynnistyspäätökseen yksityiskohtaisemmin kuin lähteessä (LUT 2011).

6. Digitaalinen tuoteprosessi – uusi konsepti



Kuva 10. Markkinaselvitykset ja projektisuunnitelma.

Yrityksen yleinen lähtötilanne on tarpeellista tuntea ennen asiakstarvekartoituksen tekemistä. Lähtötilannetta määritettäessä vastauksia voidaan hakea seuraaviin kysymyksiin: Keitä ovat asiakkaat ja kilpailijat ja mikä on yrityksen valtti

mahdollisessa kilpailutilanteessa? Kuinka laajasta kehitystoiminnasta on kysymys? Mitä asiakastarpeista ja kilpailijoista tiedetään ennakkoon sekä miten puuttuvaa tietoa lähdetään keräämään? (LUT 2011.) Oman lähtötilanteen tunteminen auttaa päätöksenteossa, kun pitää päättää, aloitetaanko tuotekehitysprojekti.

Tiedon kerääminen asiakkailta, markkinoinnilta tai muilta sidosryhmiltä on periaatteessa kahden henkilön välistä tiedonvaihtoa (LUT 2011). Tiedonvaihdon aikana osa esille tulevista vaatimuksista on vähemmän ja osa enemmän tärkeitä. Näiden tarpeiden välinen ero voi olla huomattavakin, ja siksi asiakastarpeita pitää pystyä tarkentamaan iterointikierröksittäin. Vaatimusmäärittelyn lopulla alkuperäiset asiakastarpeet ovat kehittyneet todennettaviksi suunnittelutehtäviksi. (Bray 2002.)

Asiakastarpeiden jäsentäminen ja analysointi on järkevää jakaa analyysihin tuotesuunnittelun osa-alueiden (mekaniikka, tietotekniikka, elektroniikka, hydrauliiikka ja pneumatiikka) mukaan ja tehdä alueittain karkeita luonnoksia. Analyysien hajauttamisella saadaan laajempi näkökanta asiakastarpeisiin ja asiantuntevat analysoijat jokaiselta suunnittelun osa-alueelta. Kilpailun kiristyessä tilanteen arviointi ja oman yrityksen sijoittaminen muiden kilpailijoiden kanssa samalle viivalle on hankalaa, mutta on tärkeää varmistua siitä, että kehitettävän tuotteen uutuusarvo tai markkinoita kiinnostavat ominaisuudet eivät ole samalla alueella toimivien kilpailijoiden tuotteissa.

Kun tuotekehitysprosessin lähtöarvot on kerätty, kehitysprojektille määritetään raja- ja tavoitearvot, joita noudattamalla päästään ohjatusti suunnittelutyön pariin. Lähtö- ja raja-arvoissa kaikille suunnittelutehtäville luodaan tavoitteet ja rajat siihen, missä suunnittelua tehdään. Esimerkiksi simulointien osalta määritellään, mitä osakokonaisuuksia simuloidaan ja mitkä ovat simulointien tavoitteet. Tavoitearvoja määriteltäessä asiakkaan tarpeet on syytä pitää mielessä, koska asiakkaan vaatimuksiin nähden liian korkeat tavoitteet eivät ole tarpeellisia. Ilman tavoitteiden ja rajojen asettamista kehitysprojekti voi pahimmassa tapauksessa joutua väärille urille eivätkä tulokset näin ollen enää vastaa asiakastarvetta. Kehitystyön ohjaaminen on oleellista tuloksellisessa suunnittelussa ja tavoitteiden saavuttamisessa. Tavoitteiden saavuttamiseksi kehitystyötä tulee seurata ja ohjata systemaattisesti. Saavutetuilla tuloksilla motivoidaan henkilöstöä ja niitä käytetään markkinointimateriaalina. On tärkeää kerätä palautetta toiminnasta, ottaa oppia sekä hyvästä että huonosta palautteesta ja huolehtia toimivasta dokumentoinnista (LUT 2011). Kun kaikki edellä mainitut kohdat on käyty tarkasti läpi, voidaan puhua tuotekehityksen kohteena olevan tuotteen ensimmäisestä konseptista.

6. Digitaalinen tuoteprosessi – uusi konsepti

Konseptikehityksen jälkeen tehdään aikataulu ja varataan resurssit. Aikataulu riippuu pitkälti tavoitteista, ja tietysti myös toisin päin. Esimerkiksi mitä nopeammassa aikataulussa tuotekehitysprosessin pitäisi mennä läpi, sitä enemmän joudutaan luopumaan korkeista tavoitteista. Luodun konseptin, aikataulun, tavoitteiden ja markkinaselvityksien avulla voidaan tarkastella kehitysprosessin kannattavuutta niin yrityksen kuin kilpailutilanteenkin kannalta.

Määrittelyjen, analyysien ja laskelmien tarkoituksena on varmistua tuotteen valmistamisen kannattavuudesta. Vaiheen tuloksena syntyvän dokumentin perusteella pitäisi pystyä päättämään, käynnistetäänkö tuotekehitysprosessi, tehdäänkö vielä lisäselvityksiä vai hylätäänkö se. Tuotekehitysprosessin käynnistyspäätöksen lisäksi vaiheen tuloksena syntyy dokumentti, johon on koottu asiakastarpeet (alustavat vaatimukset) ja niiden toteutuskelpoiset ratkaisumenetelmät sekä alustavat kustannukset ja resurssitarpeet.

Markkinaselvityksen ja projektisuunnitelman teon yhteydessä tuotteen vaatimukset voidaan kirjata PLM-järjestelmään omana dokumenttinaan tai järjestelmän tarjoamiin mahdollisuuksien mukaan jopa erillisinä omina vaatimusobjekteina, joihin voidaan liittää esimerkiksi vaatimusmäärittelyssä, tuotesuunnittelussa ja testauksessa syntyvää tuotetietoa. PLM-järjestelmässä olevien vaatimusobjektien etu on, että tuotteen vaatimukset saadaan esille aina siinä kohdassa elinkaarta, missä niillä on merkitystä. Näin huomataan nopeasti, mikäli tuotteen ominaisuudet ovat mahdollisesti ristiriidassa alkuperäisten vaatimusten kanssa. Tässä vaiheessa PLM-järjestelmään kannattaa luoda alustava tuotelinja tai tuote, johon kaikki siihen liittyvät dokumentit tallennetaan ennen varsinaista tuoterakenteen avaamista. Tuotteeseen voidaan liittää myös erilaisia aikataulutuksia, projektin merkkipaaluja sekä vaatimuksia ja ominaisuuksia, joita voidaan käyttää myöhemmin esimerkiksi tuotevariaatioita määriteltäessä.

6.1.2 Vaatimusmäärittely

Tuotteen vaatimusmäärittelyä on tutkittu paljon ja siitä löytyy myös paljon kirjallisuutta. Esimerkiksi kirjassa *The Requirements Engineering Handbook* on lueteltu kattava lista alan teoksia, joihin lukijaa kehoitetaan tutustumaan (Young 2004). Suomessa tutkimuksia vaatimusmäärittelyn aihepiiristä ovat tehneet esimerkiksi VTT:llä Parviainen ym. (2003) ja Pöyhönen & Hukki (2004). Näiden lisäksi aihepiiristä järjestetään vuosittain IEEE:n hallinnoima konferenssi, jossa alan tutkijat kokoontuvat yhteen ja esittävät tutkimustensa tuloksia. Springer-Link julkaisee lisäksi vielä *Requirements Engineering Journalia*, joka ilmestyy

neljä kertaa vuodessa. Tässä alaluvussa on esitetty kuvaus vaatimusmäärittelystä, joka perustuu alan laajaan kirjallisuuteen.

Vaatimusmäärittely on tuotekehitysprojektin onnistumisen kannalta erittäin ratkaisevaa, koska sillä on oleellinen rooli tuotteen onnistumisessa. Puutteellisesti määrittely tuote aiheuttaa epävarmuutta suunnitteluprosessissa, koska eri teknologioista vastaavat suunnitteluryhmät eivät tarkalleen tiedä työnsä tavoitetta. Suunnittelun epävarmuus johtaa myös helposti turhaan suunnittelutyöhön, koska puutteellisia teknisiä ratkaisuja joudutaan korjaamaan useampaan kertaan. Hyvin määritelty tuote puolestaan voidaan suunnitella ja toteuttaa pienemmällä määrällä suunnittelun iteraatiokierroksia, mikä tarkoittaa säästöä tuotekehitykseen kuluvassa ajassa ja kustannuksissa. Vaatimusmäärittely suoritetaan vähintään kerran tuotekehitysprojektin alussa, mutta vaatimuksia on täydennettävä ja tarkennettava myös suunnitteluprosessin aikana (Hietikko ym. 2003).

Vaatimusmäärittelyssä listataan tuotteen tavoite, oleellimmat toiminnot ja komponentit, teknologiset rajoitukset sekä ympäristön ja loppukäyttäjän vaatimukset. Vaatimusmäärittely on usein myynnin ja markkinoinnin, asiakkaan, työympäristön, standardien ja määräysten sekä tuotannon vaatimusten yhteensovittamista. Selkeästi dokumentoidut vaatimuslistat ja käyttötilannekuvaukset määrittelevät lähtöarvot virtuaalisuunnittelun eri osa-alueille ja edistävät siten rinnakkaista suunnittelua. (Ahola ym. 2011a).

Hyvässä vaatimuslistassa vaatimukset on priorisoitu ja ryhmitelty tärkeysjärjestyksen mukaan, jolloin ristiriitaiset tai päällekkäiset vaatimukset voidaan tunnistaa ja ottaa huomioon tuotteiden suunnittelussa (Abrahamsson ym. 2002, Parviainen ym. 2003). Mahdollisimman suuri osa moniteknisen tuotteen avainvaatimuksista pitäisikin tunnistaa ja sopia jo kehitysprojektin alkuvaiheissa.

Simulointilähtöisessä tuotekehitysprosessissa vaatimusmäärittelyn osuutta ei voida korostaa liikaa, koska selkeät tuotevaatimukset helpottavat monella tapaa virtuaalisuunnitteluvaiheen toteuttamista. Kun tuotteen tärkeimmät toiminnalliset vaatimukset ovat tiedossa ja kohdennettuina selkeiksi suunnittelutehtäviksi, on helpompi määrittellä myös niiden ratkaisemisessa käytettävien virtuaalimallien vaatimukset. Virtuaalimallien vaatimusten perusteella on edelleen helpompi valita sopivimmat mallinnusmenetelmät ja -työkalut, jotka osaltaan vaikuttavat tuotekehitysprojektin kestoon ja kustannuksiin. Virtuaalimallien vaatimusmäärittelyssä voidaan noudattaa esimerkiksi SISUQ8-menetelmää (Leppävuori ym. 2009).

Vaatimusmäärittelyä ei tehdä pelkästään suunnittelua varten. Vaatimusmäärittely toimii myös koko organisaation yhteisenä dokumenttina, jonka tarkoituks-

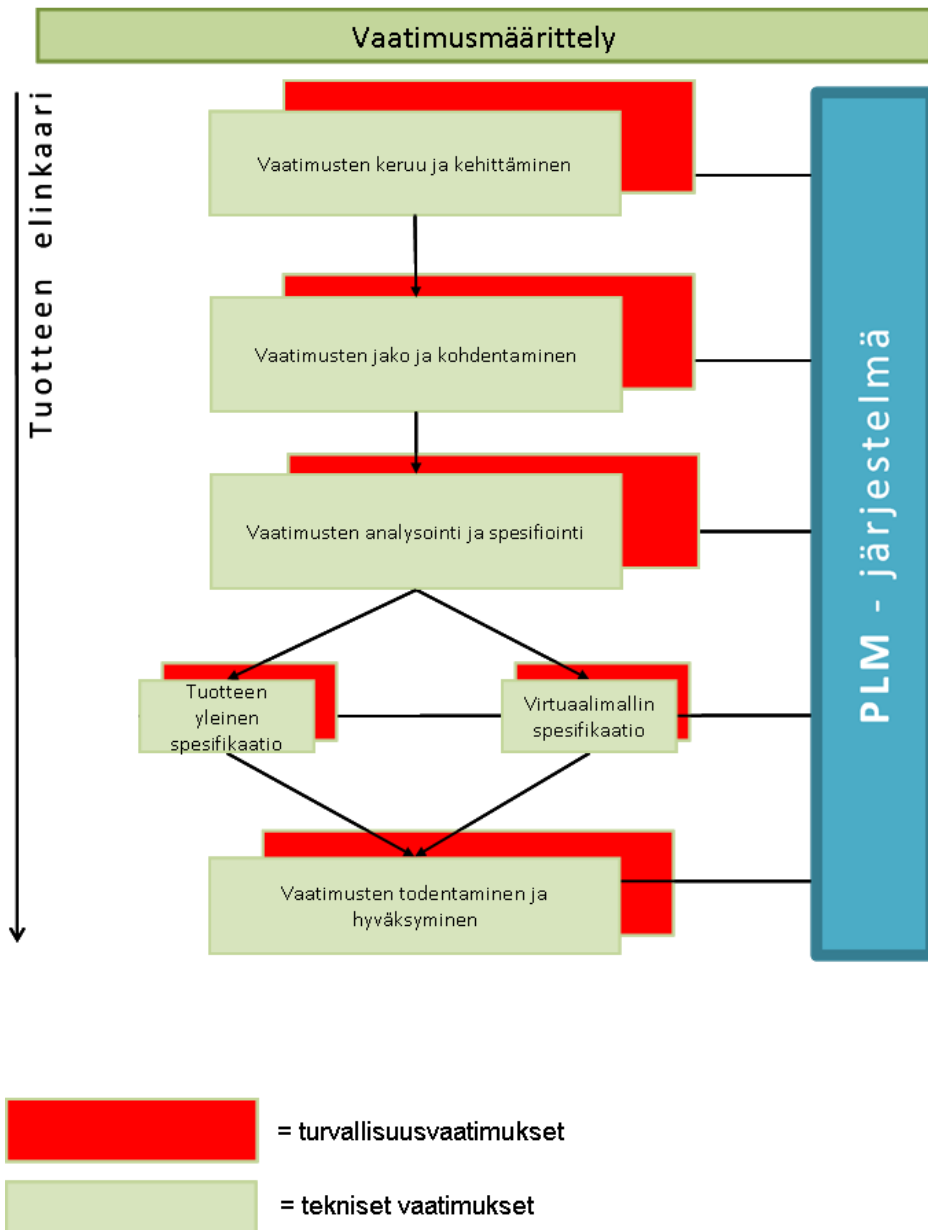
6. Digitaalinen tuoteprosessi – uusi konsepti

na on yhtenäistää eri suunnittelijoiden ja osastojen näkemystä siitä, mitä ollaan suunnittelemassa, sekä toimia myynnin ja markkinoinnin pohja-aineistona ja luoda edellytykset kannattavalle kehitysprosessille. (Vuori 2009.)

Young (2004) esittää kirjassaan 30 kohdan taulukon vaatimusmäärittelyn vaiheista, Bray (2002) pelkistää vaiheet puolestaan viiteen. Kuvassa 11 esitettävä vaatimusmäärittelyn kaavio on koottu edellä mainittujen lisäksi lähteistä Parviainen ym. (2003) ja Hietikko ym. (2009). Kaavioon on kerätty vaatimusmäärittelyn ydinkohdat moniteknisen tuotteen vaatimusmäärittelyä ajatellen.

Vaatimusmäärittely voidaan jakaa kahteen osaan, vaatimusten käsittelyyn ja vaatimusten hallintaan (Hietikko ym. 2003, Young 2004). Vaatimusten käsittely koostuu neljästä pääkohdasta, joissa keskitytään asiakastarpeiden pohjalta tuotteen suunnittelutehtävien määrittelemiseen ja tarkentamiseen. Vaatimusten hallinta on vaatimusten käsittelyn rinnalla alusta loppuun asti etenevä tukiprosessi, joka sisältää mm. dokumentoinnin, muutosten hallinnan ja jäljitettävyyden (Sommerville & Sawyer 1997). Vaatimusten käsittelyn pääkohdat ovat:

- vaatimusten keruu ja kehittäminen
- vaatimusten luokittelu, kohdentaminen ja resursointi
- vaatimusten analysointi ja spesifiointi
- vaatimusten todentaminen ja hyväksyminen.



Kuva 11. Vaativuusmäärittelyn vaiheet.

Johdonmukaisen dokumentoinnin avulla tiedetään, mitä on tehty, mitä on vielä tekemättä ja minkälaisia tuloksia on saavutettu. Muutosten hallinta ja jäljitettävyys ovat tärkeitä, koska tuotekehitysprosessin aikana tulee varmasti vaiheita,

6. Digitaalinen tuoteprosessi – uusi konsepti

joissa pitää selvittää ja tietää esimerkiksi mihin kaikkialle uusi muutos vaikuttaa. Nykyisissä PLM-järjestelmissä on hyvät työkalut dokumentointiin, vaatimusten kehittämiseen ja hallintaan. ENOVIA V6:ssa on esimerkiksi vaatimusmäärittelyyn käytössä erillinen Requirements Central -osio, joka mahdollistaa muun muassa vaatimusten hallinnan, elinkaaren seurannan ja jäljitettävyyden (Dassault Systemes 2011a).

Vaatimusten keruu ja kehittäminen

Asiakastarve- ja markkinaselvityksen tuloksien avulla tuotteelle määritetään vaatimukset (Parviainen ym. 2003). Onnistuneen vaatimusmäärittelyn olennaisen osan muodostavat asiakaskeskustelut, koska usein asiakkaan kirjaama tai kuvaama vaatimus ei ole riittävän tarkka ja yksityiskohtainen, jotta sen perusteella voitaisiin luoda kunnollisia suunnittelutehtäviä, ja tällöin vaatimusta joudutaan tarkentamaan (Vuori 2009). Vaatimuksia kerätessä ja tarkennettaessa on muistettava käyttäjän tarpeet eli ei määritellä tuotteeseen sellaisia ominaisuuksia, jotka eivät lisää tuotteen arvoa asiakkaalle. Muutenkin asiakkaan pitäminen mukana tuotekehityksessä auttaa suunnittelun kohdentamisessa, ja asiakkaalle muodostuu parempi kuva siitä, mitä hän on ostamassa.

Vaatimusten määrittelyn päätehtävä on muuttaa asiakastarpeen tuloksena saadut ominaisuudet toiminnallisiksi vaatimuksiksi (Parviainen ym. 2003). Vaatimukset voivat olla joko toiminnallisia, kuten ulottuvuus, nostokyky ja nopeus, tai ei-toiminnallisia, kuten luotettavuus, käytettävyys ja suorituskyky. Ei-toiminnalliset vaatimukset pitää pystyä kuitenkin mittaamaan ja todentamaan. Vaatimuksia kehitettäessä on tärkeää tarkastella niitä useista eri näkökulmista (asiakas, loppukäyttäjä, ympäristö sekä myynti ja markkinointi). Eri näkökulmien avulla vaatimusmäärittelystä saadaan mahdollisimman kattava.

Turvallisuusvaatimukset ovat osa nykyaikaista tuotteiden vaatimusmäärittelyä, koska lopullisen sertifikaatin tuotteelle antaa jokin suomalainen luotettu tahon. Jos tuotteen turvallisuusvaatimukset eivät täyty, ei sertifikaattiakaan voida myöntää. Turvallisuusvaatimusten keräämiseen ja hallintaan soveltuvat samat menetelmät ja työkalut kuin tuotteen muidenkin vaatimusten (Hietikko ym. 2009). Turvallisuusprosessi esitetään luvussa 6.1.3

Lyhyesti kuvattuna vaatimusten keruun ja kehittämisen vaiheet ovat seuraavat (Parviainen ym. 2003):

- vaatimusten muodostaminen asiakastarpeiden pohjalta
- tuotteen toiminnallisten vaatimusten luominen
- tuotteen toiminnallisen konseptin ja tilannekuvauksien luominen.

Vaatimusten luokittelu, kohdentaminen ja resursointi

Vaatimusten luokittelun ja kohdentamisen tarkoitus on varmistaa, että kaikki tunnetut vaatimukset on jaettu erillisiksi suunnittelutehtäviksi (Parviainen ym. 2003). Vaatimukset voidaan priorisoida välttämättömiin, hyödyllisiin ja mahdollisiin tai käytön kannalta kriittisiin ja ei-kriittisiin (Pöyhönen & Hukki 2004). Välttämättömät ja kriittiset vaatimukset voivat tulla esimerkiksi standardeista tai turvallisuusmääräyksistä, hyödylliset ja ei-kriittiset vaatimukset voivat olla esimerkiksi toimintaa auttavia ominaisuuksia ja mahdolliset vaatimukset esimerkiksi käyttömukavuuteen vaikuttavia.

Moniteknisen tuotteen kohdalla rakenne voidaan jakaa myös tuotesuunnittelun osa-alueiden mukaan, muun muassa mekaniikkaan, ohjelmistoihin ja toimilaitteisiin. Vaatimukset saadaan jaettua hyvin yksityiskohtaisesti jakamalla moniteknisen tuotteen vaatimukset ensiksi suunnittelun osa-alueiden mukaan ja sitten luokittelemalla vaatimukset kriittisyyden mukaan. Toisaalta monitekniselle tuotteelle asetettu ominaisuus voi vaikuttaa samaan aikaan useampaan suunnittelun osa-alueeseen, mikä taas monimutkaistaa rakenteen läpinäkyvyyttä. Ominaisuuksien jakaminen auttaa kuitenkin suunnittelutehtävien hajautusta omille osastoilleen ja näin ollen voidaan kohdistaa työpanosta enemmän välttämättömiin osa-alueisiin (Hietikko ym. 2009). Oleellinen osa moniteknisen tuotteen suunnittelua ovat myös osa-alueiden rajapinnat, jotka tulee käsitellä ja määritellä tarkasti (tSoft 2007). Moniteknisessä tuotteessa rajapintoja ovat esimerkiksi mekaniikka – toimilaitteet – anturit – ohjelmisto.

Suunnittelutehtävien kohdentamiseen liittyvät oleellisesti myös resurssit. Vaatimusmäärittelyssä resurssien kartoituksen tehtävänä on määrittää asiakastarvekartoitusta tarkemmin, mitä tuotekehitysprosessin vaiheista voidaan tehdä itse ja mitä joudutaan ostamaan alihankintana. Käytännössä tuotekehitysprosessin mikä tahansa vaihe voidaan ostaa alihankintana, kunhan vaatimukset alihankinnan tehtävistä määritellään riittävän selkeästi. Alihankkijoiden ja ulkopuolisten resurssien käytössä oleellista on tarvittavan tiedonsiirron ja sen turvallisuuden varmistaminen. Yksi ratkaisu toimivaan yhteydenpitoon on PLM-järjestelmän käyttöoikeuden antaminen alihankkijalle. Alihankkijan käyttöoikeudet voidaan

6. Digitaalinen tuoteprosessi – uusi konsepti

rajata järjestelmässä niin, että alihankkijalla on pääsyoikeus vain tarvittaviin tiedostoihin, kuten valmistuspiirustuksiin tai yksittäisen osakokonaisuuden 3D-malleihin. PLM-järjestelmiin voidaan luoda useita erilaisia automaattisia toimintoja, kuten hyväksymis- ja ilmoitusketjuja. Esimerkiksi ilmoitusketjun avulla alihankkijalle voidaan lähettää tieto siitä, että tarvittavat tiedostot ovat noudettavissa PLM-järjestelmästä, ja sama toisin päin, kun alihankkija on oman osuutensa toteuttanut.

Tiedonsiirtoformaatti pitää määrittää jokaisen alihankkijan kanssa erikseen, riippuen siitä, millä alalla alihankkija toimii ja minkälaisia ohjelmistoja tällä on käytössään. Tiedostoformaattien yhtenäistäminen poistaisi useita tiedostonsiirrosta syntyviä ongelmia, mutta formaattien yhtenäistäminen on nykypäivän ohjelmistoviidakossa mahdotonta useiden ohjelmistovalmistajien ja käyttöjärjestelmien takia. Tiedonsiirtoformaatin oikeellisuudesta PLM-järjestelmään voidaan määrittää esimerkiksi yksinkertainen tarkistustehtävä, jonka avulla ylimääräisestä tiedonsiirrosta päästään eroon.

Prosessit ovat myös osa tuotteen määrittelyä. Prosessien määrittelyn kautta saadaan käsitys tuotekehitysprosessiin kuuluvista vaiheista, tehtävistä ja tekijöistä. Esitetystä tuotekehityskonseptista suunnittelun hallintatyökaluna toimii PLM-järjestelmä, jonne prosessi voidaan kuvata. Mallinnettuun prosessiin voidaan liittää erilaisia toimintoja, kuten hyväksymisketjuja, aikataulutusta, suunnittelijoiden tehtäviä ja paljon muuta. Prosessi sisältää siis kaikki tuotekehityksen vaiheet ja näin ollen myös tuotteen ominaisuudet ja niitä vastaavat tuoterakenteet, suunnitelmat, tulokset jne. PLM-järjestelmiin voidaan rakentaa myös integraatiot ulkopuolisista järjestelmistä. Tutkimusprojektin aikana käytössä olleeseen ENOVIA V6:een oli rakennettu suora liityntä Dassault Systèmesin Catia-suunnittelutyökalusta. Liitännän avulla Catiassa tehdyn suunnitelman rakenne voidaan ladata suoraan kokonaisuudessaan PLM-järjestelmään. Lisätietoja löytyy tarvittaessa Dassault Systèmesin internetsivuilta (Dassault Systèmes 2011b).

Vaatimusten analysointi ja spesifiointi

Vaatimusten analysoinnin ja spesifioinnin tarkoituksena on jäsentää tehtävälueen sisältö (Bray 2002) suunnittelijoiden ymmärtämissä tehtäviksi. Yksittäisten vaatimusten ja suunnittelutehtävien välisten riippuvuuksien mallintaminen selvittää ja auttaa ymmärtämään vaatimukset (tSoft 2007). Vaiheen tuloksena on dokumentti, jonka perusteella suunnittelija pystyy suunnittelemaan tuotteen.

Vaatimuksia analysoitaessa on hyvä miettiä myös tuotteen konfiguroitavuutta, jos tuotteen vaatimuslista on pitkä ja erilaisia toteutettavia ominaisuuksia on paljon. Toisin sanottuna on syytä miettiä, mikä on tuotteen ominaisuuksien taso, joka kannattaa ja on mahdollista toteuttaa kustannustehokkaasti yhteen ja samaan tuotteeseen. Konfigurointi tarkoittaa tässä yhteydessä, että generisestä tuoterakenteesta, joka sisältää useita toisiaan poissulkevia vaihtoehtoja, valitaan halutut ominaisuudet ja muodostetaan näin tuoteyksilö. Tehdyt valinnat vaikuttavat toisiinsa tarjolle tuleviin valintavaihtoehtoihin tuotekonfiguraattorissa ennalta määriteltyjen ehtojen mukaisesti, mikä takaa, että lopullinen tuoterakenne on myös toteutettavissa. Näin muodostetut tuoteyksilöt voivat poiketa toisistaan ja ovat tuotevariaatioita, joilla on sama perusrakenne, mutta variaatiot eroavat toisistaan kuitenkin joiltain oleellisilta osiltaan. Asiakas on myös voinut konfiguroida tuotteen myyntikonfiguraattorissa valitsemalla tuotteeseen liitettävät ominaisuudet ennalta määriteltyjen ominaisuuksien välisten ehtojen mukaisesti. Nämä ehdot voivat olla muun muassa ominaisuuksia sitovia tai ominaisuuksia pois rajaavia.

Ominaisuuksia sitova ehto määrittelee esimerkiksi sen, että jos autossa on kahden litran moottori, siihen kuuluu automaattisesti ilmastointi. Ominaisuuksia pois rajaava ehto taas voidaan ilmaista niin, että jos autossa on 1,5-litrainen moottori, siihen ei ole saatavilla automaattivaihteistoa. Ominaisuuksien välisten riippuvuuksien hallinta voi olla PLM-järjestelmään liitetty ominaisuus. Tästä aiheesta on kirjoitettu myös luvussa 2 kohdassa ”Myynti ja markkinointi”.

MoniDigi-projektissa toteutettiin myyntikonfiguraattori, jolla hankittavan varsiston käyttötarkoitusta voitiin muuttaa valitsemalla varsiston päähän sijoitettavat työkalut määriteltyjen ehtojen mukaisesti. Raportti myyntikonfiguraattorin toteuttamisesta on liitteessä A. Nykyään myyntikonfiguraattoreita käytetään yleisesti myyntityön tukena. Ohjelmistotalot tarjoavat paljon omia konfiguraattoreitaan tuotteiden konfigurointiin ja visualisointiin. Yksi hyvä esimerkki myyntikonfiguraattorista on autovalmistajien käyttämät ”rakenna itse autosi” -sovellukset internetissä, joissa käyttäjä voi rakentaa auton moottorin valinnasta lisälaitteiden valintaan sovellukseen määriteltyjen ehtojen mukaisesti.

Vaatimusten todentaminen ja hyväksyminen

Vaatimusten todentaminen on vaihe, jossa käydään systemaattisesti läpi, mikä tuotteen tekninen vaatimus vastaa mitäkin asiakastarpeen ominaisuutta. Vaatimusten todentamisesta tehdään dokumentti, josta riippuvuussuhteet selviävät

6. Digitaalinen tuoteprosessi – uusi konsepti

(Parviainen ym. 2003). Vaatimusmäärittelyn viimeisessä vaiheessa määrittelylle haetaan hyväksyntä. Hyväksynnän antaa viime kädessä asiakas.

6.1.3 Turvallisuusprosessi

Konevalmistaja on päävastuussa uuden tuotteen turvallisuudesta. Koneenvalmistajan tehtävät tuoteturvallisuuden varmistamiseksi on määritelty koneasetuksessa (Työsuojeluhallinto 2008) seuraavasti:

- arvioida turvallisuusriskit
- selvittää konetta koskevat turvallisuusvaatimukset
- suunnitella ja rakentaa kone olennaisten turvallisuusvaatimusten mukaisesti
- laatia käyttöohjeet ja tehdä koneeseen tarvittavat merkinnät
- laatia tekninen tiedosto
- tehdä vaatimustenmukaisuus vakuutus
- kiinnittää koneeseen CE-merkintä.

Vaikka tyyppihyväksyntäviranomainen tarkastaisi ja hyväksyisi tuotteen, se ei poista tai siirrä konevalmistajan vastuuta koneturvallisuudesta. Vastaavasti jo käytössä olevien koneiden modernisointiin sovelletaan käyttöasetusta (Malm & Hämäläinen 2006).

Turvallisuusprosessi etenee tuotekehitysprojektin kanssa rinnakkain, ja tavoitteena on varmistaa, että tuote täyttää sille asetettavat turvallisuusvaatimukset (kuva 12). Turvallisuusprosessin malliksi on valittu KOTOTU (Koneiden ohjausjärjestelmien Toiminnallinen Turvallisuus) -projektin referenssimalli, joka noudattaa riskinarvioinnin standardia SFS-EN ISO 14121, riskin pienentämisen periaatteiden standardia SFS-EN ISO 12100, ohjausjärjestelmien turvallisuusstandardia SFS-EN ISO 13849 ja ohjausjärjestelmien kelpuutusstandardia SFS-EN ISO 13849. (Tiusanen ym. 2007, 2008, Hietikko ym.2009 mukaan.). Turvallisuussuunnitelman malli on vastaavasti lainattu standardista IEC 62061, koska se ei sisällä edellä mainittuihin standardeihin (Hietikko ym. 2009).

Kuvassa 12 esitetystä turvallisuusprosessin mallista on pääasiassa kolme riskianalysikierrosta (Hietikko ym. 2009):

1. alustava vaara-analyysi (tehdään koko koneelle kattaen kaikki mahdolliset vaarat)
2. käyttötapa-analyysi (ihmisen tekemistä virheistä aiheutuvat vaarat)

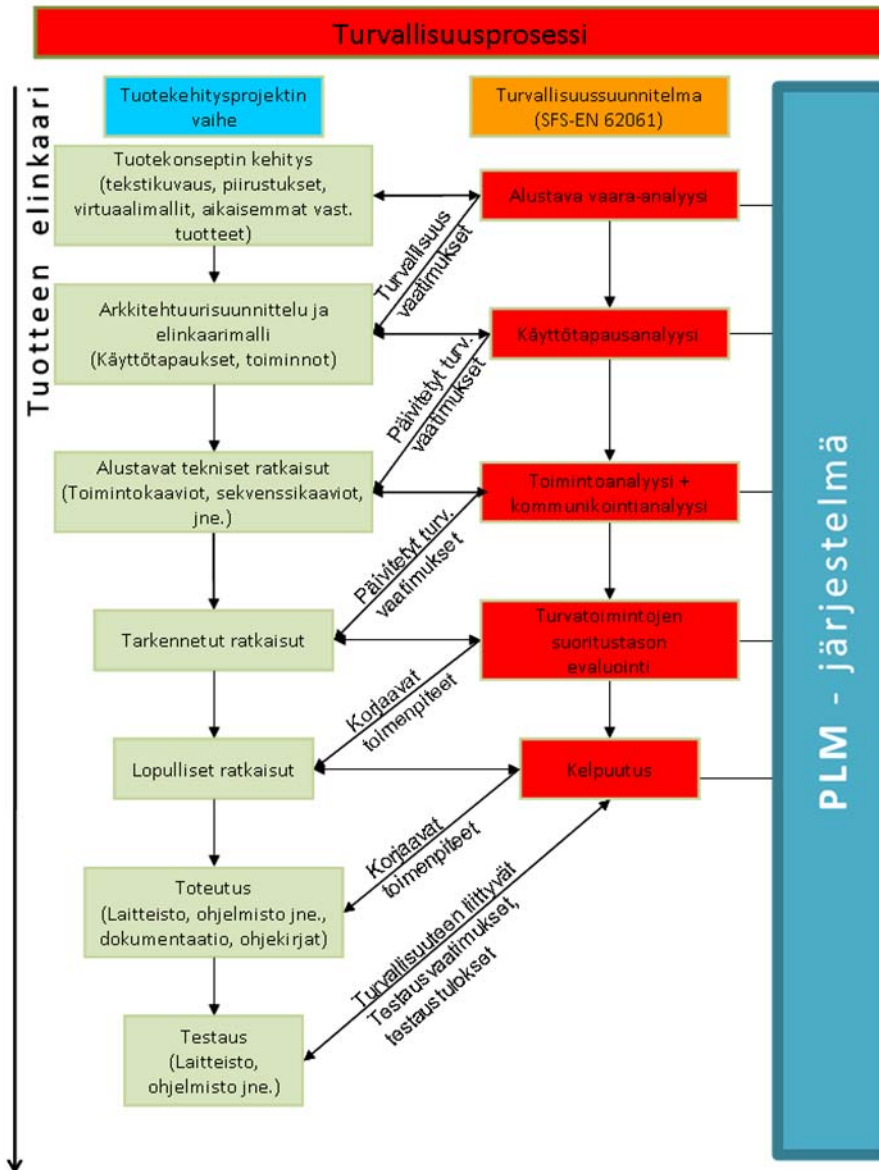
3. toimintoanalyysi (vika-vaikutusanalyysi koneen käyttö- ja turvatoiminoille).

Nämä kolme analyysivaihetta riittävät tyypillisen koneenohjausjärjestelmän kattavaan ja sopivin väliajoin tarkentuvaan riskianalyysiin. Pk-yritysten näkökulmasta KOTOTU-referenssimallin etu on, että kolme riskianalyysikierrosta ei vielä vaadi kohtuuttomasti työaikaa (Hietikko ym. 2009). Koska turvallisuusprosessi on sinänsä todella laaja aihealue, tässä ei käsitellä sitä yksityiskohtaisemmin, vaan lukijaa pyydetään tutustumaan mainittuihin lähteisiin ja standardeihin.

Yrityksen turvallisuusinsinööri voi tehdä tuotekehitysprojektin aikana vaadittavat riskianalyysit. Sen sijaan järjestelmän kelpuuttavien henkilöiden tulisi olla riippumattomia koneen tai laitteen suunnittelusta, mutta heidän ei tarvitse olla muusta organisaatiosta (Hietikko ym. 2009). Pk-yrityksissä ei kuitenkaan välttämättä ole riittävästi asiantuntemusta tai resursseja kaikkien asetuksien ja standardien perinpohjaiseen selvittämiseen ja turvallisuusprosessin toteuttamiseen niiden mukaisesti. Tällaisessa tilanteessa pk-yrityksen ainoa vaihtoehto on teettää tarvittavat riskianalyysit alihankkijalla. Suomessa tuoteturvallisuuspalveluja tarjoavat mm. VTT Expert Services Oy, Comatec Oy ja Noritek Oy.

Jos turvallisuusprosessi hallitaan yrityksen PLM-järjestelmässä, se mahdollistaisi pk-yrityksille turvallisuusprosessin joustavan alihankinnan. PLM-järjestelmään voitaisiin tarpeen mukaan avata pääsyoikeudet turvallisuusprosessin asiantuntijaorganisaatiolle, joka suorittaisi tuotteelle turvallisuussuunnitelman mukaisen riskianalyysin. Tällä tavalla riskianalyysit ja kelpuutusprosessit voitaisiin tarpeen mukaan alihankkia samaan tapaan kuin tuotteen suunnittelu, valmistus tai jälkimarkkinointikin.

6. Digitaalinen tuotekehitys – uusi konsepti



Kuva 12. Turvallisuusprosessi etenee tuotekehitysprojektin kanssa rinnakkain (mukailtu Hietikko ym. 2009).

6.1.4 Virtuaalisuunnittelu

Virtuaalisuunnittelun tavoitteena on nopeuttaa tuotekehitysprosessia, pienentää tuotekehityksen kustannuksia ja riskejä sekä parantaa suunnittelun laatua. Nämä tavoitteet saavutetaan parhaiten, jos kaikki suunnittelutiimit ja sidosryhmät (asiakas, loppukäyttäjä, myynti, tuotanto jne.) osallistuvat aktiivisesti virtuaalisuunnitteluun tuotekehityksen alusta alkaen. Luonnosteluvaiheessa kolmiulotteisten virtuaaliprototyyppien vahvuus on siinä, että ne havainnollistavat tuotteen toiminnallisuutta, jolloin puutteet suunnittelussa tai tuotteen vaatimuksissa ovat helposti havaittavissa. Kokemuksen mukaan virtuaalimallin demonstraatio herättää sidosryhmissä aina spontaania ja vilkasta keskustelua, joka on tärkeää vuorovaikutteisessa ja asiakassuuntautuneessa tuotekehityksessä. Virtuaalisuunnittelu pitäisikin nähdä koko tuotekehitysjärjestelmän eikä vain erillisen simulointiosaston asiana, joka silloin tällöin tarkistaa simuloimalla tuotteen tai suunnittelun yksityiskohtia.

Tällä hetkellä simulointi ja mallinnusohjelmistojen hankkiminen ja käyttö on kohtuuttoman kallista pk-yrityksille. Virtuaalisuunnittelun käyttöönoton riskejä ja kustannuksia voitaisiin pienentää, jos yrityksillä olisi mahdollisuus ostaa eri virtuaalisuunnitteluohjelmistojen käyttöaikaa tarpeen mukaan. Ehdotetussa toimintamallissa yrityksillä olisi ohjelmistoihin niin sanottu verkkolisenssi, joka vaatisi verkkoyhteyden lisenssipalvelimelle. Toistaiseksi Suomesta puuttuu simulointiohjelmistojen verkkolisenssejä myyvä palvelun tarjoaja. Tämäkin toimintamalli vaatisi vielä sen, että yrityksissä olisi simulointiohjelmistojen käyttöön perehtyneitä ja koulutettuja henkilöitä.

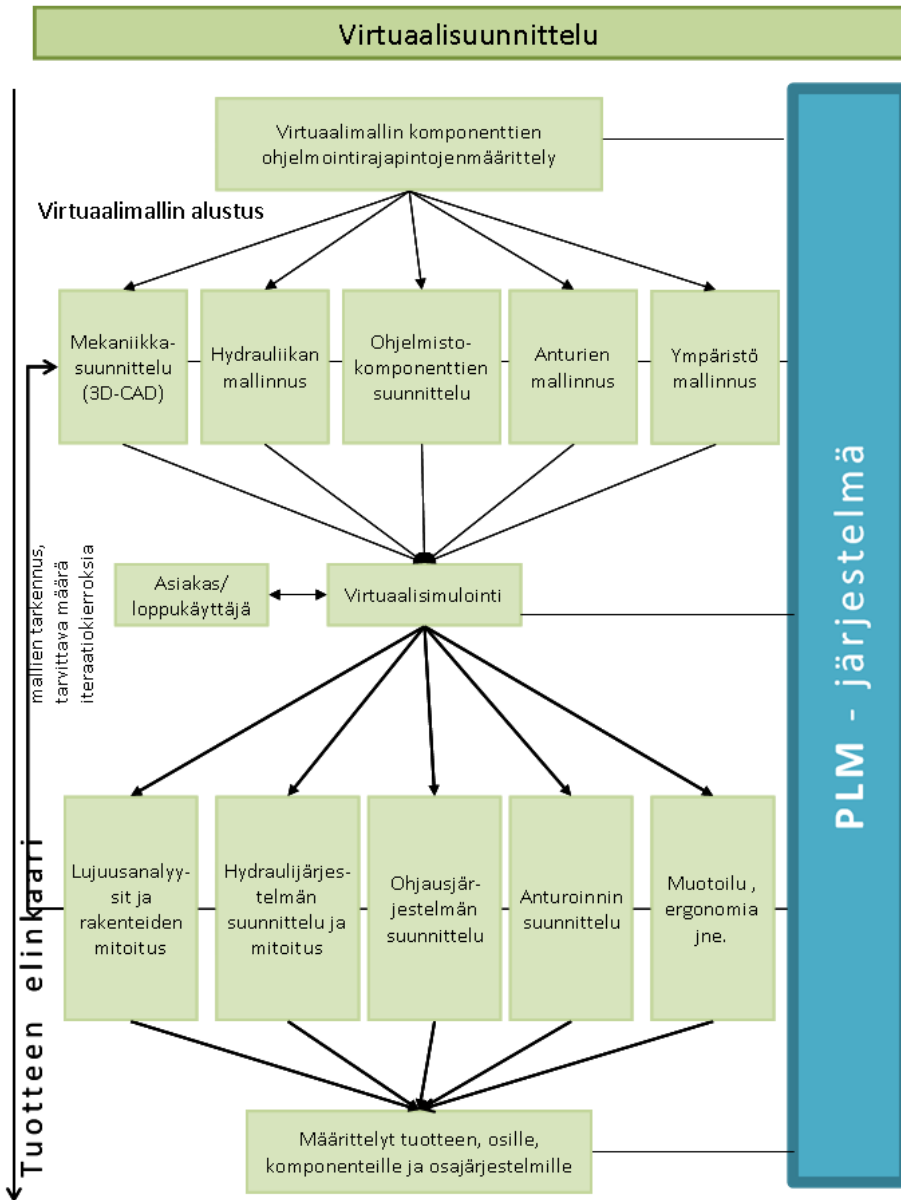
Modulaarinen mallinnus tukee rinnakkaista suunnittelua

Simulointipohjaisessa suunnitteluprosessissa mallinnus on mukana tuotekehityksen alusta alkaen, ja iteraatiokierroksien edetessä tuotteen malli lähestyy kohti teknisesti toteutettavissa olevaa ratkaisua (Ahola ym. 2011b). Moniteknisen tuotteen modulaarinen virtuaalimalli muodostuu suunnittelun eri osa-alueita vastaavista komponenteista: mekaaninen malli, hydraulikkamalli, anturimallit, ohjelmistokomponentit ja ympäristömallit, jota havainnollistetaan kuvassa 13. (Ahola ym. 2011a). Virtuaalimallin komponenttien alustus voidaan tehdä rinnakkain, koska parhaan lopputuloksen saavuttamiseksi osamallit kannattaa toteuttaa itsenäisinä kokonaisuuksina. Käytännön mallinnusprojekteissa modulaarinen mallinnus on todettu joustavaksi ja helpommin hallittavaksi kuin jos teh-

6. Digitaalinen tuoteprosessi – uusi konsepti

täisiin yksi, mutta huomattavasti monimutkaisempi malli (Lehtonen 2006, Kortelainen 2006, Paredis ym. 2001). Simulointipalveluja tarjoavalle yritykselle modulaarinen mallinnus mahdollistaa myös sen, että koko virtuaalimallia ei tarvitse rakentaa yrityksen omilla resursseilla, vaan osa mallinnuksesta voidaan tarpeen mukaan hankkia palveluna yrityksen ulkopuolelta. Mallinnustietoa hallitaan PLM-järjestelmässä, ja tarvittaessa eri mallinnusosapuolet vaihtavat tietoa PLM-järjestelmän kautta.

Ennen virtuaalisimulointia osamallit yhdistetään tuotteen kokonaissimulointimalliksi. Tässä esitetty tuotteen kokonaissimulointimalli on monikappalesysteemimalli (MBS), johon on yhdistetty hydrauliiikka-, ohjelmisto- ja anturimalleja ja tarpeen mukaan. Osamallien yhteensopivuus varmistetaan käyttämällä yhteensopivia mallinnusohjelmistoja ja määrittelemällä osamallien formaatti ja ohjelmointirajapinnat ennen osamallien toteuttamista. Mallin kokoamisen jälkeen simuloidaan tuotteen tai sen osajärjestelmän toiminnallisuutta. Tuotteen suunnittelusta vastaava yritys voi strategiansa mukaan päättää, hankkiiko se simulointipalvelua alihankintana, vai tekeekö se virtuaalimallin kokoamisen ja simuloinnit omilla resursseillaan.



Kuva 13. Moniteknisen tuotteen rinnakkaisen virtuaalisuunnittelun vaiheet.

Ennen kuin simulointituloksia voidaan hyödyntää tuotteen yksityiskohtien suunnittelussa, tarvitaan arvio virtuaalimallin luotettavuudesta, eli malli täytyy verifioida. Mallin verifiointi perustuu vaatimusmäärittelyssä listattuihin virtuaali-

6. Digitaalinen tuoteprosessi – uusi konsepti

mallin vaatimuksiin, joissa on mainittu kaikki suunnitteluongelmat, joihin mallin avulla etsitään ratkaisua. Verifioinnin tavoite on varmistaa, että virtuaalimalli vastaa tuotekehityksen tavoitteita ja että malli on riittävän hyvä (tarkka) suunnitteluongelmien ratkaisemiseen. Koska moniteknisen virtuaalimallin luotettavuuden arviointi on erittäin haastavaa ja vaatii laaja-alaista osaamista, mallit on suositeltavaa verifioida mallintajan ja eri osa-alueiden asiantuntijoiden yhteistyönä. Verifioinnin ja testauksen jälkeen mallien rajoitukset ja puutteet tulee dokumentoida, jotta malleja käytetään oikeaan tarkoitukseen myös myöhemmin.

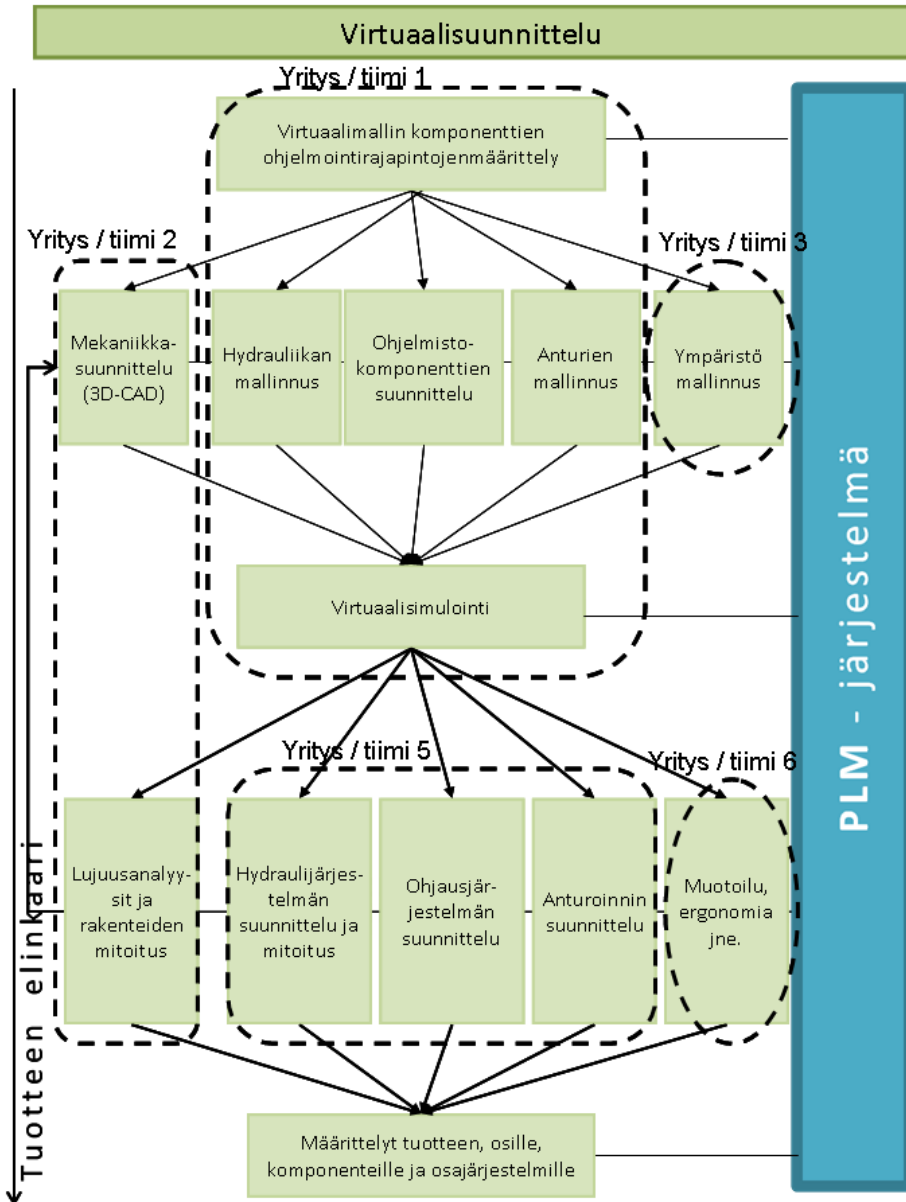
Mallin verifioimisen jälkeen simulointituloksia hyödynnetään mm. rakenteiden ja toimilaitteiden mitoituksessa sekä komponenttivalintojen lähtötietoina. Toisaalta simuloinnilla voidaan testata laitteistoriippumattomia ohjelmistokomponentteja sekä arvioida mittausjärjestelmän tarkkuutta. Virtuaalisuunnittelun päätavoite on saada tekniset määrittelyt olennaisimmille ja välttämättömille komponenteille, joita tarvitaan todellisen tuotteen tai osajärjestelmän prototyyppien rakentamisessa. Tässäkin vaiheessa yritys voi sopia tietyn osajärjestelmän suunnittelusta alihankkijan kanssa. Osajärjestelmää koskevat simulointitulokset välittyvät PLM-järjestelmän kautta järjestelmän suunnittelusta vastaaville yhteistyöyrityksille.

Nykyaikaiset 3D-suunnitteluohjelmistot mahdollistavat tuotteen muotoilun ja ergonomian kehittämisen rinnakkain muun teknisen suunnittelun kanssa. Pk-yrityksellä ei useinkaan ole mahdollisuutta palkata omaa teollista muotoilijaa, vaan on kustannustehokkaampaa joko käyttää sopimuspalkkaista muotoilijaa tai ostaa tarvittavat muotoilupalvelut muotoilutoimistosta (Ekman & Sääskilahti 2002). Yrityksen ulkopuolisen muotoilijan kytkeminen tuotekehitysprosessiin hoituu käytännöllisimmin PLM-järjestelmässä, josta muotoilusta vastaava toimija saa kaiken tarvittavan ja ajan tasalla olevan tuotetiedon. Muotoilija voi vastaavasti toimittaa muotoilua koskevat parannusehdotukset saman järjestelmän kautta esimerkiksi kuvien ja mallitiedostojen muodossa. Jos toimitaan ilman PLM-järjestelmää, ollaan pelkästään manuaalisen tiedonsiirron varassa ja tällöin on vaarana, että muotoilijat tekevät turhaa työtä puutteellisilla tai vanhentuneilla tiedoilla. Sama riski koskee myös kaikkia muita suunnittelun osa-alueita.

Esimerkki verkottuneesta tuotekehityksestä

Kuvassa 14 on esimerkki virtuaalisuunnittelun vastuualueiden jakamisesta verkostomaisessa tuotekehityksessä. Yritys/tiimi 1 koordinoi virtuaalimallinnuksen, sovittaa osamallit yhteen, suorittaa simuloinnit ja analysoi niiden tulokset.

Tiimi 2 puolestaan vastaa mekaniikkasuunnittelusta ja toimittaa tuotteen 3D-osamallit simulointiympäristöön. Tiimi 3 selvittää ja laserkeilaa työkoneen tyyppilliset käyttöympäristöt, käsittelee keilatun pistepilven ja muodostaa niistä 3D-pintamallin, joka edelleen siirretään simulointiympäristöön.



Kuva 14. Esimerkki moniteknisen tuotteen verkottuneesta virtuaalisuunnittelusta.

6. Digitaalinen tuoteprosessi – uusi konsepti

Simulointitulokset ohjaavat tuotteen reaalisuunnittelusta vastaavia tiimejä. Tiimi 1 tarkistaa rakenteiden lujuuden tuotteen dynaamisesta mallista määritettyjen kuormitusten avulla ja tekee niihin muutoksia tarvittaessa. Vastaavasti tiimi 5 suunnittelee ja mitoittaa hydraulijärjestelmän simuloinnissa määritettyjen virtaus- ja painetasojen mukaisesti. Tiimi 5 suunnittelee myös tuotteen automaatio-ratkaisut (anturointi, ohjauslogiikka, käyttöliittymä jne.), jotka todennetaan virtuaalimallin avulla. Tiimi 6 vastaa tuotteen muotoilusta sekä käytettävyyden ja ergonomian kehittämisestä.

Virtuaalisuunnittelu tuotteen elinkaaren aikana

Tuotteen elinkaaren aikana virtuaalisuunnittelu voidaan jakaa karkeasti ottaen kahteen vaiheeseen: alustaviin simulointeihin ja yksityiskohtaiseen virtuaalisuunnitteluun. Alustavat simuloinnit ovat hyödyllisiä erityisesti uusien tuotteiden kohdalla, kun uutta tuotetta vastaavia aikaisempia prototyyppisiä tai tuoteversioita ei ole olemassa. Alustavilla simuloinneilla määritetään tuotteen toiminnallisuuden kannalta kriittisten parametrien suuruusluokat suunnittelun eri osaluueille (mekaniikka, hydraulikka, ohjelmisto, anturointi). Tuotekehityksen alkuvaiheessa teknisten vaatimusten konkretisoiminen on erityisen tärkeää, koska tuotteen toiminnallisuus saattaa olla vasta osittain määritelty eikä teknistä toteutusta ole välttämättä vielä luonnosteltu. Alustavat simuloinnit auttavat määrittelemään selkeät tavoitteet eri osa-alueista vastaaville suunnittelijoille ja siten minimoimaan puutteellisista tiedoista johtuvaa turhaa suunnittelutyötä.

Siirtyminen alustavista simuloinneista yksityiskohtaiseen virtuaalisuunnitteluun tapahtuu asteittain, eikä näiden suunnitteluvaiheiden ero ole aina yksiselitteinen. Alustavan simuloinnin ja yksityiskohtaisen suunnittelun vaiheet voidaan erottaa toisistaan ainakin seuraavilla kriteereillä:

- Alustavien simulointien tavoitteena on etsiä periaatteellisia teknisiä ratkaisuja (mekaanisia ja ohjelmallisia) tuotteen toteuttamiseksi.
- Yksityiskohtaisen virtuaalisuunnittelun tavoitteena on mitoittaa ja määrittellä hw-komponentteja (toimilaitteet, anturit, ohjausventtiilit, I/O-rajapinta) sekä tuottaa lopulliseen tuotteeseen sulautettavaa ohjelmakoodia.
- Siirryttäessä alustavista simuloinneista yksityiskohtaiseen virtuaalisuunnitteluun mallien alustus ja simulointituloksien analysoiminen vaativat huomattavasti enemmän aikaa ja muita resursseja.

Virtuaalisuunnittelu on suositeltavinta aloittaa luonnostellen karkeilla malleilla, joita tarkennetaan iteratiivisesti suunnittelun edetessä. Muutoksien tekeminen virtuaalimalliin on nopeaa ja joustavaa verrattuna siihen, että tuotteesta tai sen kriittisistä osista rakennettaisiin todellisia prototyyppejä (Ahola ym. 2011a). Varsinkin luonnosteluvaihetta 3D-simuloinnit tehostavat merkittävästi, koska simulointitulokset ovat visuaalisuudessaan konkreettisia ja niissä havaitut puutteet ovat nopeasti korjattavissa. Simulointimallien monimutkaisuus kasvaa kuitenkin nopeasti, kun edetään yksityiskohtaiseen suunnitteluun. Laajojen simulointimallien alustamisessa ohjelmointivirheen riskiä voidaan pienentää hyödyntämällä mahdollisimman paljon ennalta testattuja ohjelmistokomponentteja ja automaattisia ohjelmakoodin generointityökaluja.

Hyvistä ohjelmointikäytännöistä huolimatta numeeriset simulointitulokset saattavat olla epäluotettavia, jos mallin parametrit eivät vastaa todellisuutta. Toisaalta simulointimalli saattaa olla hyvin spesifinen, jolloin se pätee vain hyvin rajatussa käyttötilanteessa. Mallinnuksen kustannusten ja riskien pienentämiseksi onkin käytännöllisempää simuloida tuotteen eri toimintoja useammalla yksinkertaisella mallilla, kuin rakentaa yksi kaiken kattava, mutta samalla monimutkainen ja virheille altis malli. Riskejä voidaan myös pienentää kiinnittämällä huomiota mallien verifiointiin ja validointiin sekä dokumentoimalla mallien rajoitukset. (Ahola ym. 2011a).

Verkostoituminen on pk-yrityksille tärkeää, koska moniteknisten virtuaalimallien verifiointi ja validointi ovat eri alojen asiantuntemusta vaativaa tiimityötä eikä niitä voida jättää pelkästään mallintajan vastuulle. Simulointien kustannustehokkuuden kannalta on tärkeää määritellä ennen mallinnuksen aloittamista, mitkä ovat simuloinnin tavoitteet, tuloksilta vaadittava tarkkuus sekä mallien alustukseen käytettävissä oleva aika ja resurssit.

Moniteknisen tuotteen kokonaissimulointimallin alustaminen ja ohjelmistot

Tuotteen mekaniikan ja ohjauksen samanaikainen simulointi vaatii yleensä erillisen ohjelmiston, joka tukee monikappalesysteemien (MBS) mallinnusta. Vaikka useimmista CAD-ohjelmistoista löytyvät työkalut mekanismien mallintamiseen ja simulointiin, niiden hyödyntäminen tuotteen ohjauksen suunnittelussa on hankalaa pääasiassa kahdesta syystä:

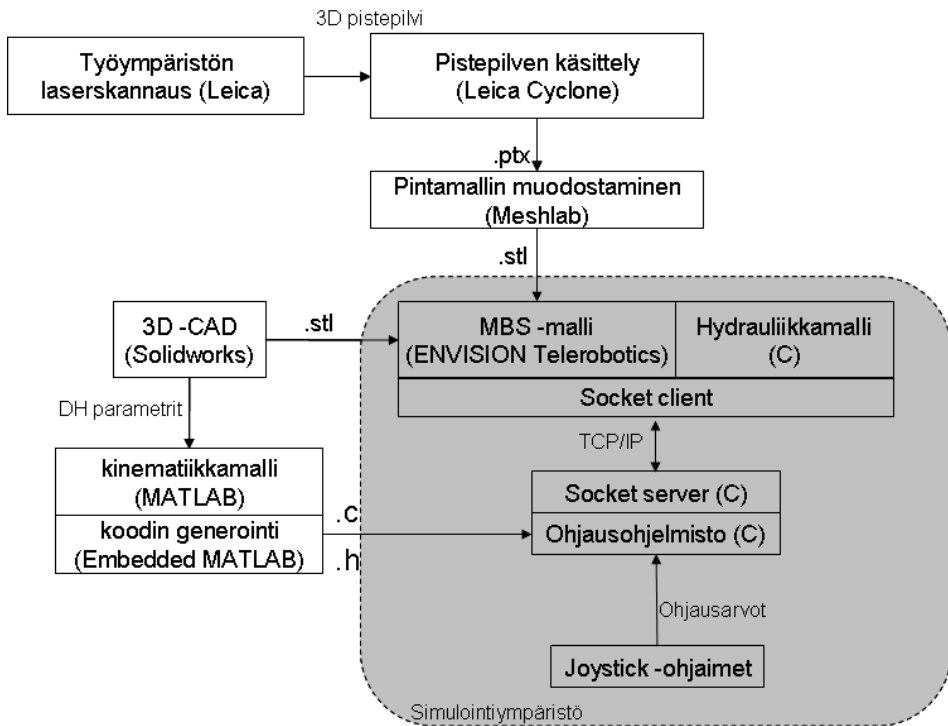
6. Digitaalinen tuoteprosessi – uusi konsepti

1. CAD-ohjelmistoista puuttuu ohjelmointirajapinta, jonka avulla mekaaniseen malliin voitaisiin helposti linkittää ohjaukseen liittyvää ohjelmakoodia.
2. Mekanisimimallit ovat CAD-ohjelmistokohtaisia eivätkä ne ole yhteensopivia muiden ohjelmistojen kanssa. (Ahola ym. 2011b).

Joihinkin CAD-ohjelmistoihin on kuitenkin saatavilla joko ohjelmistovalmistajan tai kolmannen osapuolen tekemiä lisäosia mekanisimimallien siirtoon.

Edellä mainituista syistä tuotteen mekaaninen malli on suositeltavaa luoda erillisellä MBS-simulointiin tarkoitetulla ohjelmistolla, kuten ADAMS (Msc Software, Santa Ana, Kalifornia) tai ENVISION Telerobotics (Dassault Systèmes, Vélizy-Villacoublay, Ranska). Näihin ohjelmistoihin voidaan tuoda osamalleja eri CAD-järjestelmistä muuntamalla osamallit simulointiohjelmiston kanssa yhteensopivaan tiedostoformaattiin. CAD-mallien siirron jälkeen osien väliset vapausasteet määritellään MBS-ohjelmiston omilla työkaluilla. ADAMS-ympäristössä voidaan edelleen määritellä rajapinta MATLAB/Simulink-ohjelmaan (Mathworks Inc., Natick, Massachusetts, USA), luomalla MBS-mallista Simulink-ohjelmaan sopiva ohjausblokki (ControlsPlant). Tämä ohjausblokki sisältää muun muassa kaikki ohjattavat suureet sekä MBS-ohjelmassa tehtävät mittaukset. Vastaavasti ENVISION TR:ssä olevaan virtuaalimalliin voidaan linkittää ohjauksen ohjelmakoodia TCP/IP-rajapinnan avulla. (Ahola ym. 2011b).

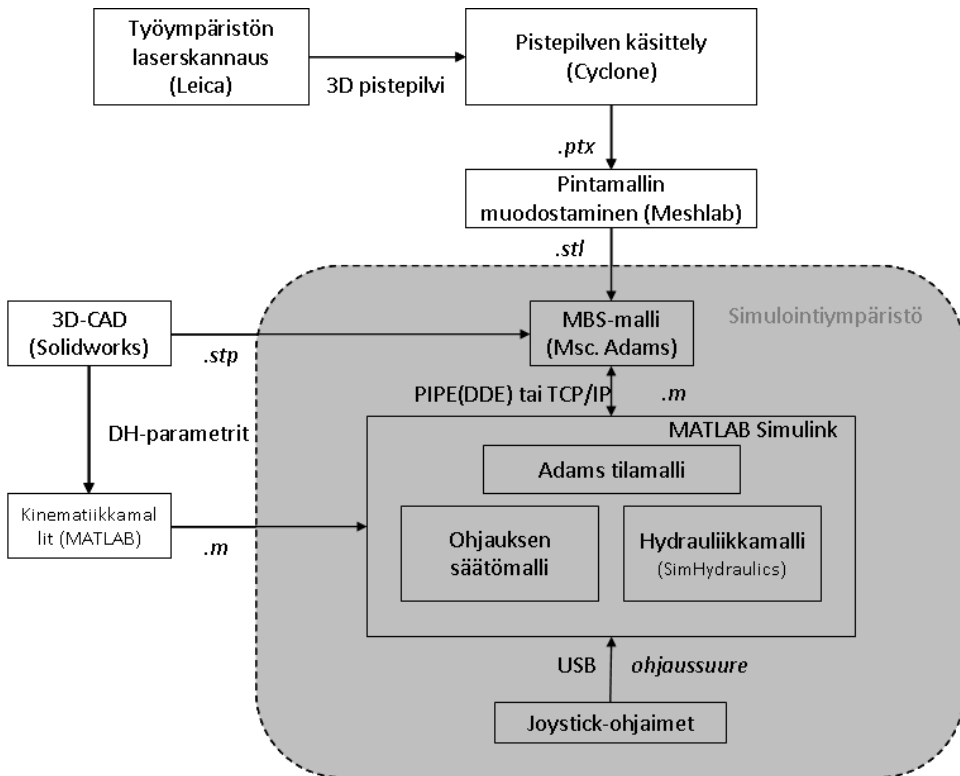
Kuvissa 15, 16 ja 17 on esimerkkejä kokonaissimulointimallin alustukseen ja simulointiin käytetyistä ohjelmistoista MoniDigi-projektissa. Puomin osat mallinnettiin Solidworks-ohjelmalla ja osien 3D-mallit tallennettiin ENVISION TR-ohjelman kanssa yhteensopivaan stl-formaattiin. Mallien siirron jälkeen ENVISIONissa koottiin CAD-malleista tuotteen MBS-simulointimalli määrittelemällä osien väliset vapausasteet (nivelet).



Kuva 15. Simulointiympäristö käyttäen ENVISION TR:ää ja C-kielistä ohjausohjelmistoa.

Dynamiikan simulointia varten ENVISION TR -ohjelman jaettuun kirjastoon (shared library) ohjelmoitiin C-kielinen hydrauliikkamalli sisältäen toimilaitteiden ja ohjausventtiilien mallit. Jaettuun kirjastoon ohjelmoitiin myös TCP/IP socket -asiakaskone (client), joka kommunikoi palvelimena (server) toimivan ohjausohjelmiston kanssa. Puomin kinematiikkamallin lähtöarvoina olivat DH-parametrit, jotka määritettiin puomin osien 3D-malleista. Kinematiikkamalli ohjelmoitiin ensin MATLABilla, josta generoitiin edelleen valmis C-koodi sulautettavaksi ohjausohjelmistoon simulointiympäristössä. Työympäristö keilatettiin Leican laserkeilaimella (Leica Geosystems Oy, Espoo). Laserkeilattu 3D-pistepilvi käsiteltiin edelleen Leica Cyclone- ja Meshlab-ohjelmistoilla, joilla muodostettiin pistepilvestä kolmioitu pintamalli, joka siirrettiin ENVISIONiin yhteensopivassa stl-formaatissa.

6. Digitaalinen tuoteprosessi – uusi konsepti



Kuva 16. ADAMS- ja MATLAB Simulink -ohjelmistoihin perustuva simulointiympäristö.

Toinen projektin aikana käytetty MBS-ohjelma oli ADAMS, jonka simulointiympäristö on esitetty kuvassa 16. Peruseriaate ADAMS ja MATLAB Simulink ohjelmistojen yhteiskäytössä on sama kuin aikaisemmin kuvatussa ENVISION- ja MATLAB Simulink -ympäristössä, eli MBS-ohjelmaan tuodaan 3D-CAD-ohjelmalla mallinnetut osat käyttämällä esimerkiksi osien stl-siirtomuotoa ja DH-parametrien avulla mallinnetaan rakenteen kinematiikka MATLABiin, josta dynamiikkamallia ohjataan. ADAMS sisältää erillisen Controls-osion, jonka avulla voidaan määrittellä MBS-mallin ulos- ja sisääntuloja. Controls määrittelee tilamuuttujat ja luo MATLAB-ohjelmaa varten erillisen m-tiedoston, joka ajetaan MATLABissa. M-tiedosto luo MATLABin tietokantaan tarvittavan koodin, josta voidaan erillisellä komennolla muodostaa Simulinkiin ns. ADAMS-block, joka sisältää liittynät ADAMS-malliin. ADAMS ja MATLAB Simulink käyttävät ohjelmistojen väliseen tiedonsiirtoon Windows-pohjaista PIPE (DDE) -menetelmää. Ohjelmistojen synkronointi hoidetaan mää-

rittelemällä MATLAB Simulinkistä yhtenevä simuloinnin askelväli sekä MATLABille että ADAMSille.

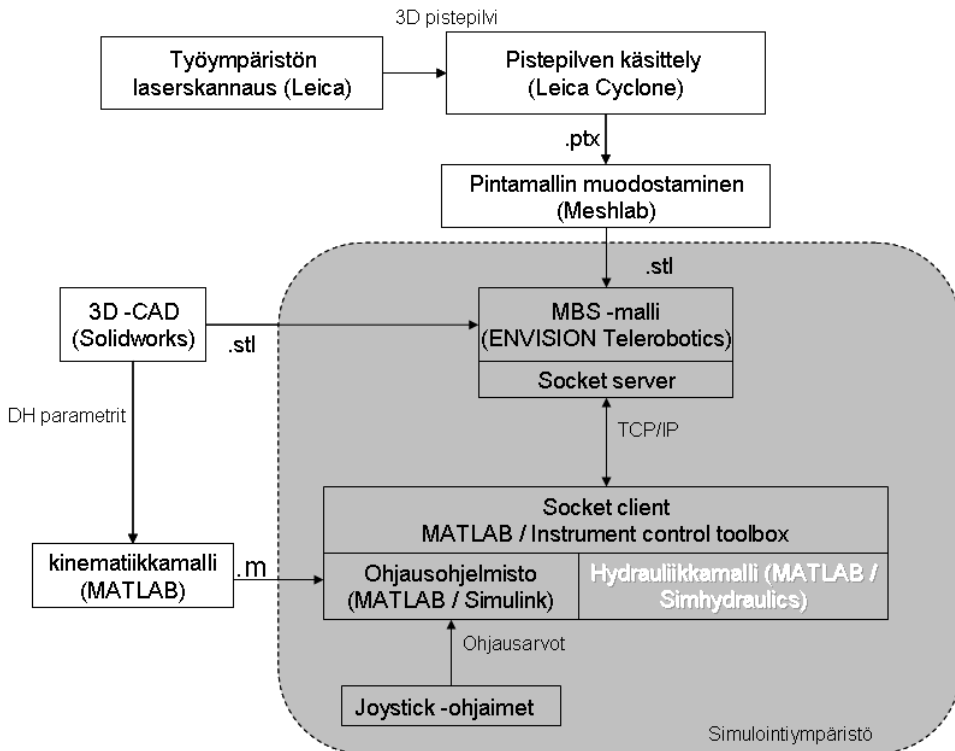
Molemmille MBS-ohjelmistoille on yhteistä se, että niiden mekaanisten rakenteiden mallinnusominaisuudet eivät ole 3D-CAD-ohjelmien tasolla. Mekaaniset rakenteet kannattaa mallintaa 3D-CAD-ohjelmalla ja siirtää mallit sopivassa siirtomuodossa (stl, stp tai muu vastaava) MBS-ohjelmiin. Mallin kokoamista helpottaa huomattavasti, jos mallin mukana siirtyvät tiedot esimerkiksi nivelpisteistä. Nykyään pisteet pitää liittää malliin käytännössä manuaalisesti yksi kerrallaan tai esimerkiksi ADAMSissa ajamalla cmd-tekstitiedosto, johon pisteiden luontikoodi on kirjoitettu. ENVISION TR:n tapauksessa hydraulikkamallit toteutettiin C-ohjelmointina, mutta ADAMS-ohjelmiston tapauksessa on mahdollista käyttää ADAMSiin Controls-osion avulla liitettävää EASY5-ohjelmistoa (Msc Software, Santa Ana, Kalifornia). EASY5 on Msc Softwaren ohjelma, jolla voidaan mallintaa, simuloida ja suunnitella koneen hydraulikkaa.

Muita ohjelmistoratkaisuja simulointiympäristöjen toteuttamiseen

MoniDigi-projektissa toteutettiin myös kuvan 17 mukainen simulointiympäristö hyödyntäen MATLABin Instrument Control Toolboxia, joka sisältää tarvittavat funktiot TCP/IP-protokollan mukaiseen tiedonsiirtoon. MATLABin kirjastoilla voidaan toteuttaa vain asiakaskone, mutta ohjelmistojen välinen tiedonsiirto on kuitenkin mahdollinen, koska ENVISION TR:ää voidaan käyttää myös palvelimena. Koordinoidun ohjauksen simuloinnissa puomin käänteinen kinematiikka ratkaistiin MATLABissa ja nivelarvot lähetettiin TCP/IP-yhteydellä ENVISIONiin, joka visualisoi puomin liikkeit kolmiulotteisella puomimallilla. Instrument Control Toolboxin avulla TCP/IP-ohjelmointirajapinta voidaan toteuttaa myös Simulinkiin, jolloin puomin hydraulikka voidaan mallintaa graafisesti Simhydraulics-lisäosan avulla. Simhydraulics-simulointiesimerkki on esitetty liitteessä B.

TCP/IP-ohjelmointirajapinta MATLABin ja ENVISIONin välillä lisäsi virtuaalisuunnittelun joustavuutta merkittävästi, koska ohjausohjelmat voitiin suorittaa MATLAB-koodina. Tämä helpotti myös simulointiohjelmistojen hallintaa ja ylläpitoa, koska ohjausalgoritmeista ei tarvinnut tuottaa C-kielisiä versioita simulointia varten. TCP/IP-ohjelmointirajapinta mahdollisti myös MATLABin laskentaominaisuuksien täysimittaisen hyödyntämisen simuloinnissa, koska vain noin 270:stä MATLAB-funktiosta voidaan generoida C-koodia. (Ahola ym. 2011b).

6. Digitaalinen tuoteprosessi – uusi konsepti



Kuva 17. Simulointiympäristö, jossa on TCP/IP-yhteys ENVISION TR:n ja MATLAB Simulinkin välillä (Ahola ym. 2011b).

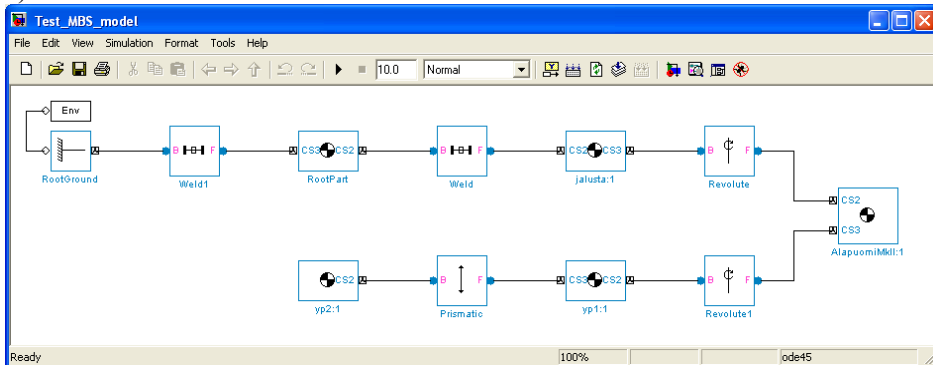
Uutena ohjelmistona mainittakoon esimerkiksi 3D-mekaniikkamallinnusta tukeva MATLABin SimMechanics-lisäosa, johon voidaan siirtää kokoonpanoja CAD-suunnitteluohjelmistoista Autodesk Inventor, Solidworks ja Pro/Engineer. Kokoonpanojen siirtäminen tapahtuu SimMechanics Link -lisäosan avulla, jonka voi ladata maksutta Mathworksin www-sivuilta (Mathworks 2011). SimMechanics Link generoi automaattisesti stl-mallit CAD-kokoonpanosta sekä xml-tiedoston, jossa osien väliset nivelet, osien massakeskipisteet ja hitausmomentit määritellään. MBS-malli luodaan MATLABiin lataamalla kokoonpanoa kuvaava xml-tiedosto, jolloin Simulink-malli ja 3D-visualisointimalli muodostuvat automaattisesti. MoniDigi-projektissa testattiin kolmen vapausasteen mekaniikkamallin siirto Autodesk Inventor -ohjelmasta MATLABiin SimMechanics Linkin avulla.

Xml-tiedostosta automaattisesti generoitu Simulink-malli on kuvassa 18 a. Malli muodostuu SimMechanics -lohkoista, joissa määritellään kinemaattisen

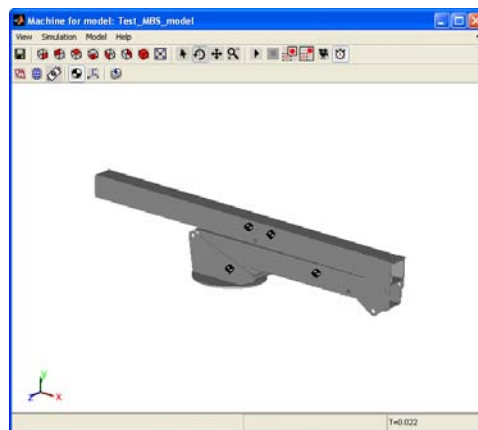
ketjun osien ja nivelten ominaisuudet. Oletusarvoisesti mekanismimallin nivelet ovat ideaalisia ilman liikerajoituksia tai kitkaa. Mallia voidaan kuitenkin tarpeen mukaan laajentaa Simulinkin komponenttikirjastolla lisäämällä niveliin tarvittava määrä rajoituksia, antureita ja toimilaitteita. Myös ohjauksen ohjelmakoodin liittäminen malliin on mahdollista. Kuvassa 18 b on mekanismimallin 3D-visualisointi, joka perustuu osien stl-tiedostoihin.

Ympäristömallien yhdistäminen SimMechanicsin 3D-visualisointiin ei ollut mahdollista. Työympäristön ja puomin liikkeiden visualisointi olisi kuitenkin mahdollista Simulink 3D Animation -lisäosalla, jolla voidaan luoda stl-malleista virtuaalitodellisuuden kuvauskieleen (vrml) perustuvia virtuaalimalleja. Simulink 3D animation -lisäosaa ei testattu MoniDigi-projektissa, mutta Mathworksin sivuilta löytyy lisätietoja aiheesta.

a)



b)



Kuva 18. Virtuaalimallinnus käyttäen MATLAB SimMechanics -ohjelmistoa: a) Puomivariston MBS-malli MATLAB Simulinkissä ja b) mallin 3D-visualisointi.

6. Digitaalinen tuoteprosessi – uusi konsepti

Katsaus monitekniisten järjestelmien simulointiin soveltuviin kaupallisiin ja avoimen lähdekoodin ohjelmistoihin on lisäksi teoksessa Lehtonen (2006). Lisätietoja kannattaa etsiä myös eri ohjelmistovalmistajien kotisivuilta. Avoimen lähdekoodin ohjelmistoista erityisen lupaavia ovat MATLABiin ja Simulinkiin verrattavat Scilab- ja Xcos-ohjelmistot sekä Modelica-ohjelmointikieleen perustuva OpenModelica.

Työprosessilähtöinen virtuaalisuunnittelu

Virtuaalisessa työympäristössä voidaan tutkia monitekniisen tuotteen geometrian ja ohjausjärjestelmän soveltuvuutta työtehtävään (kuva 19). Erityisen hyvin virtuaalinen työympäristö soveltuu tuotteen mekaanisten rakenteiden ja työkiertojen suunnitteluun. Työvaiheita simuloimalla voidaan ennakoida esimerkiksi ulottuvuuden tai törmäysten kannalta ongelmallisia työvaiheita. Simulointitulosten perusteella tuotteen geometriaa voidaan muokata työtehtävään sopivammaksi. Luotettavan soveltuvuusarvion tekeminen vaatii kuitenkin mahdollisimman todenmukaisen mallin työympäristöstä. (Ahola ym. 2011b).

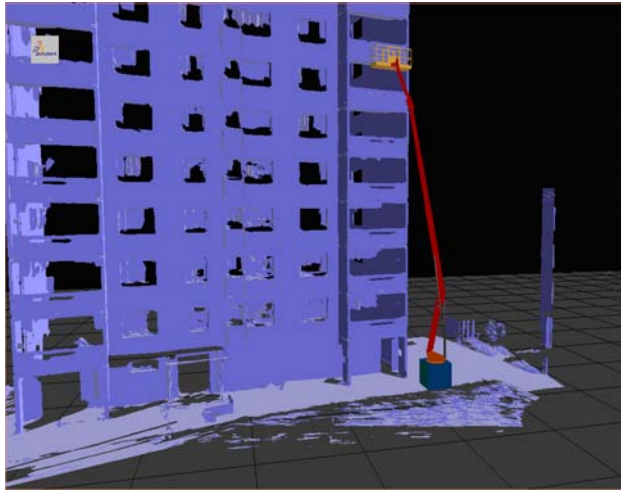
Työprosessilähtöisten alustavien simulointien tavoitteena ei ole mahdollisimman todellinen ohjaustuntuma, vaikka nykyisillä simulointiohjelmistoilla sekin on mahdollista. Lähes reaaliaikainen työvaiheiden simulointi virtuaaliympäristössä kuitenkin helpottaa käyttöliittymän ja tuotteeseen liittyvän mittausjärjestelmän suunnittelua. Virtuaaliympäristössä voidaan tutkia esimerkiksi loppukäyttäjälle esitettävää informaatiota, joka helpottaa tuotteen käyttöä ja ohjaamista.

Mahdollisimman todellinen työympäristö simulointeihin saadaan vain mitaamalla tuotteen toimintaympäristöä tai kohteita, joissa tuote tulee toimimaan. Laserkeilaus on käyttökelpoinen ja varsin tehokas menetelmä tarkkojen ympäristömallien luomiseen. Laserkeilaamalla ympäristöstä voidaan luoda todellisuutta vastaava malli kaikkine yksityiskohtineen, kuten kuvasta 20 nähdään. Kuvan 20 talomalli sekä ympäristö koostuvat noin 2,5 miljoonasta pisteestä. Mallin visualisointia varten yksittäiselle pisteelle voidaan antaa väri, jonka avulla mallista saadaan havainnollisempi. Kuvan 20 a pisteille annetut värit ovat peräisin keilauksen yhteydessä otetuista valokuvista ja kuvassa 20 b pisteet on väritetty intensiteettiarvon voimakkuuden mukaan (Ahola ym. 2011b).

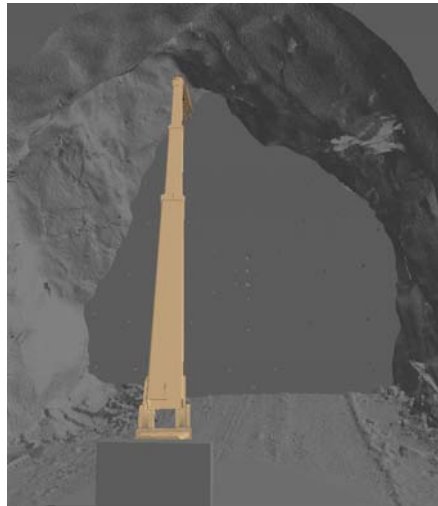
Osa nykyaikaisista laserkeilaimista tallentaa mitatun pisteen koordinaattiarvojen lisäksi myös intensiteettiarvon. Intensiteetti kertoo palaavan lasersäteen voimakkuuden, joka voi vaihdella etäisyyden tai heijastuspinnan mukaan. Intensi-

teetin avulla voidaan esimerkiksi tasomaiselta pinnalta erottaa tekstuuria. (Joala 2006)

a)



b)

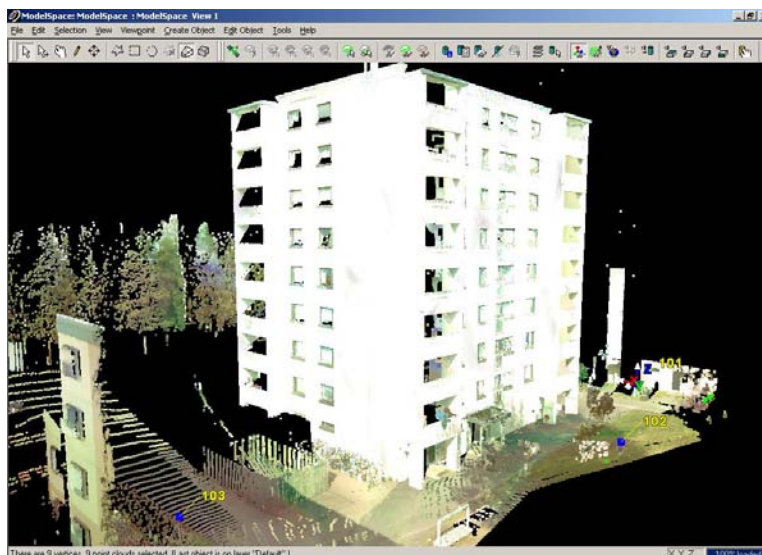


Kuva 19. Erilaisten puomiratkaisujen ulottuvuustarkastelut ENVISION Telerobotics -ohjelmalla hyödyntäen a) laserskannattua kerrostalon mallia (Ahola ym. 2011a) ja b) laserskannattua tunnelimallia (Ahola ym. 2011b).

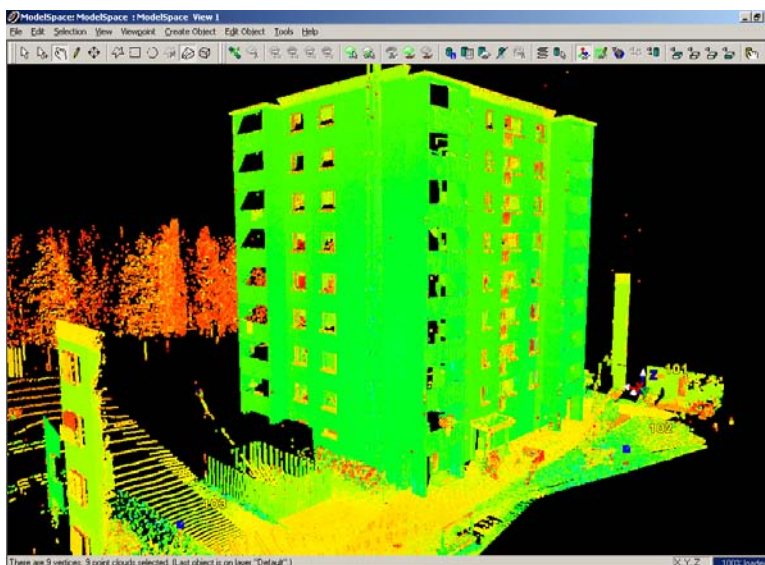
6. Digitaalinen tuoteprosessi – uusi konsepti

Laserkeilatut pistepilvet voivat sellaisinaan käydä joihinkin simulointiohjelmistoihin, mutta esimerkiksi ENVISIONia varten pistepilvi jouduttiin kolmioimaan ja muodostamaan siitä pintamalli, joka sen jälkeen siirrettiin ENVISIONiin.

a)



b)



Kuva 20. Laserkeilatun kerrostalon 3D-malli Cyclone-ohjelmassa (Ahola ym. 2011b).

Hydraulijärjestelmän esisuunnittelu virtuaalisuunnittelun työkaluilla

Hydraulijärjestelmän mallinnuksen ja simuloinnin tavoite on mitoittaa hydraulijärjestelmän pääkomponentit työprosessin vaatimien liikeratojen ja kuormitusten lähtökohdista. Integroimalla hydrauliikkamalli osaksi tuotteen MBS-mallia voidaan simuloida työkierron aiheuttamia staattisia ja dynaamisia kuormituksia, joiden perusteella mitoitetaan toimilaitteet ja valitaan järjestelmän painetaso. Toimilaitteiden mitoituksen jälkeen voidaan määrittää työkiertojen vaatimat toimilaittekohtaiset öljyvirtaukset sekä järjestelmän kokonaisöljyvirtaus, jotka ovat oleellisia tietoja hydraulijärjestelmän mitoituksessa. Määritettyjen virtaus- ja painetietojen perusteella voidaan edelleen mitoittaa ohjausventtiilit, hydrauliikkapumppu ja teholähde, paineakut, putkistot, suodattimet ja öljysäiliön tilavuus. Suunnittelun edetessä ja virtuaalimallin tarkentuessa simulointiin on mahdollista yhdistää vielä virtaushäviöiden ja lämpötaseen laskentaa, joiden perusteella mitoitetaan jäähdytysjärjestelmä. (Ahola ym. 2011b).

Hydraulijärjestelmän osalta virtuaalisuunnittelu pienentää olennaisesti virheellisten komponenttivalintojen riskiä, koska valinnat perustuvat koko hydraulijärjestelmän simulointiin yhdessä tuotteen mekaanisen mallin kanssa. Simuloinnilla myös varmistetaan, että mekaanishydraulinen kokonaisuus vastaa mahdollisimman hyvin tuotteen toiminnallisia vaatimuksia. (Ahola ym. 2011b).

Liikkuvan kaluston tapauksessa hydraulikomponenttien selkeästä ja luotettavasta määrittelystä on etua myös ajoneuvoalustan suunnittelussa, koska komponenteille voidaan suunnitella riittävät tilavaraukset alustan 3D-malliin. Nämä seikat tarkoittavat kustannussäästöjä ensimmäisen 0-sarjan tuotteen rakentamisvaiheessa, koska valitut komponentit ovat sopivia eikä alustaan tarvitse tehdä kalliita muutostöitä tilanpuutteen takia.

Hydrauliikkamallin integrointi tuotteen virtuaalimalliin riippuu käytettävästä MBS-ohjelmistosta. ADAMS-ohjelmiston tapauksessa hydrauliikkamalli voidaan toteuttaa MATLAB/Simulink- tai EASY5-ohjelmistoilla, jotka linkitetään MBS-simulointimalliin. ENVISION Telerobotics -ohjelman tapauksessa hydrauliikkamallin toteuttamiseen on kolme vaihtoehtoa:

1. Hydrauliikkamalli ohjelmoidaan C-kielillä jaettuun kirjastoon, joka linkitetään MBS-ohjelman simulointimalliin.
2. Toisessa työasemassa suoritettava hydrauliikkamalli linkitetään MBS-simulointimalliin TCP/IP-yhteydellä.
3. Toisessa sovellusohjelmassa tai työasemassa suoritettava hydrauliikkamalli linkitetään MBS-malliin PIPE-yhteydellä.

6. Digitaalinen tuoteprosessi – uusi konsepti

Jos hydraulikkamalli toteutetaan MATLAB Simscape Simhydraulicsilla, simulointiympäristön arkkitehtuuri on kuvan 16 mukainen. MATLABissa on lisäksi oltava asennettuna Instrument Control Toolbox, jonka avulla MATLABia voidaan käyttää TCP/IP-asiakaskoneena. Simhydraulicsin komponenttikirjasto sisältää valmiiksi kaikki hydraulikan peruskomponentit, kuten paineakut, yksi- ja kaksitoimiset sylinterit ja rotaattorit, vakio- ja säätötilavuuspumput ja -moottorit, suuntaventtiilit sekä paineen ja virtauksen säätöventtiilit. Simhydraulicsissa on valmiina esimerkiksi 10 erilaista 4-tiesuuntaventtiiliä, joiden kytkentäkaaviot esitetään ohjelmiston dokumentaatiossa. Tarvittaessa valmiiden venttiilimallien lisäksi on mahdollista luoda myös täysin uusia venttiilimalleja Simhydraulicsin komponenttikirjaston avulla.

Simhydraulicsin venttiilimallit eivät oletusarvoisesti vastaa minkään venttiilivalmistajan tuotteita, ja mallit on parametrisoitava todellisia venttiilejä vastaaviksi. Käytännössä tämä tarkoittaa venttiilimallin paine-virtauskäyrän sovittamista venttiilin dokumentaatiossa esitettyihin paine-virtauskäyriin. Simhydraulicsin venttiilimallit eivät myöskään ota huomioon dynamiikkaa, joka on mallinnettava erikseen esimerkiksi Simulinkin kirjastosta löytyvillä siirtofunktioidilla. Siirtofunktioiden parametrit on myös asetettava siten, että mallin taajuus- tai askelvasteet ovat venttiilin dokumentaation mukaiset. Simhydraulicsin suuntaventtiilien parametrisointiohje on esitetty lähteessä Tchkalov & Miller (2010). Esimerkki Simhydraulicsin yhteiskäytöstä ulkoisen MBS-simulointiohjelmiston kanssa on liitteessä B.

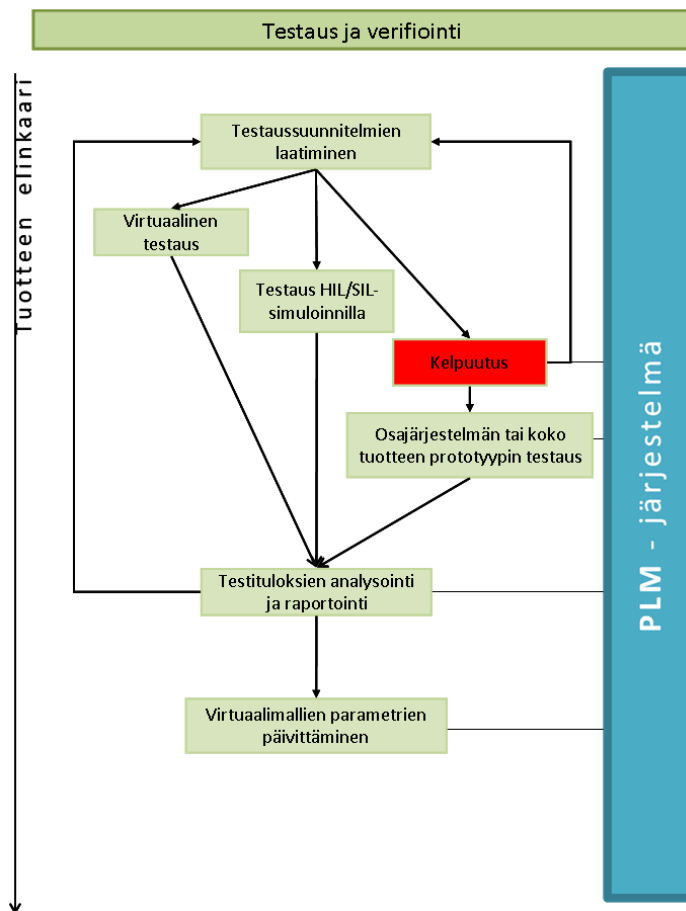
6.1.5 Testaus ja verifiointi

Moniteknisen piensarjatuotteen digitaalisen tuoteprosessin konseptissa testaus ja verifiointi on vaiheistettu virtuaaliseen testaukseen, HIL- ja SIL-simulointiin (Hardware-in-the-Loop/Software-in-the-loop) simulointiin ja todellisilla prototyypeillä tehtäviin testeihin (kuva 21). Vaiheistuksen tavoitteena on minimoida fyysisten prototyyppien rakentamiseen liittyvät riskit ja kustannukset. Tuote ja sen tekniset ratkaisut testataan mahdollisimman pitkälle virtuaalisesti, jolloin havaittuja puutteita ja suunnitteluvirheitä voidaan korjata edullisesti. HIL- tai SIL-simuloinnilla on mahdollista testata eri ohjainvaihtoehtoja tuotteen mekaanishydraulisen virtuaalimallin kanssa, ennen kuin aloitetaan kalliimpi koko prototyypin kattava rakennusprojekti.

Testauksen ja verifiointin tavoitteena on varmistaa moniteknisen tuotteen ohjelmiston, osajärjestelmien ja koko tuotteen virheetön toiminta ja parempi laatu.

Virtuaalisuunnitteluvaiheessa tapahtuvan testauksen tavoitteena on mahdollisimman kattavasti varmistaa ohjelmistojen virheettömyys, mekaanisten rakenteiden ja toimilaitteiden mitoitukset sekä oikeat vaatimukset komponenttivalinnoille (esimerkiksi anturit, hydrauliventtiilit, ohjainmoduulit).

Virtuaalinen testaus voidaan edelleen jakaa kahteen osaan, virtuaalisuunnittelun aikana tehtäviin suunnitteluosa-alueiden testauksiin sekä kokonaisvirtuaalimallin testaukseen. Mallinnuksen aikana tehtävät testaukset ja verifiointit liittyvät osa-alueiden (mekaniikka, hydrauliiikka, anturointi jne.) toiminnallisuuksien tarkastamiseen ja lähtötietojen hankkimiseen seuraavaa suunnittelukierrosta varten. Kokonaisvirtuaalimallin testauksessa keskitytään ensisijaisesti tuotteen eri vaatimusten testaukseen.



Kuva 21. Testauksen ja verifiointin vaiheet.

6. Digitaalinen tuoteprosessi – uusi konsepti

Varsinkin koko tuotteen virtuaalisista testauksista on hyvä tehdä täydelliset raportit, koska niitä voidaan verrata todelliselle prototyypille tehtäviin testeihin. Käytännön testien jälkeen on myös tärkeää päivittää virtuaalimallien parametrit siten, että virtuaalisen prototyypin käyttäytyminen vastaa mahdollisimman hyvin todellisen prototyypin mitattua käyttäytymistä. Todenmukainen virtuaaliprototyyppi on tuotteen jatkokehityksen kannalta hyödyllinen, koska virtuaaliprototyypillä voidaan testata paljon sellaista, jossa todellinen prototyyppi saattaisi rikkoutua: esimerkiksi henkilönostimen stabiilisuustarkastelut, mekaaninen ylikuormitus, vakavat ohjelmistovirheet sekä anturien ja komponenttien yhteisvikaantuminen. Mikään viranomainen ei kuitenkaan vielä myönnä esimerkiksi henkilönostimille sertifikaatteja pelkästään simulointimallilla tehtyjen stabiilisuustarkastelujen perusteella. On mielenkiintoista nähdä, kuinka paljon simulointiohjelmistojen ja tyyppihyväksyntäprosessien tulee kehittyä, että näin voitaisiin menetellä ainakin osittain.

Nykypäivää testauksen osalta ovat HIL- ja SIL-simulointi, joissa osa prototyypistä toteutetaan virtuaalisena simulaatiomallina. HIL-simuloinnissa virtuaalisena toteutetaan tyyppillisesti prototyypin mekaniikka ja SIL-simuloinnissa ohjainlaitteisto (Hardware). Tällaiset osittain virtuaaliset prototyypit vaativat simulointiympäristön, joka tukee reaaliaikaisuutta eli simulaatiomallin tahdistamista todellisen ajan kanssa. HIL- ja SIL-simuloinnin etuina voidaan mainita, että ne nopeuttavat ohjelmistojen ja ohjaimien testaamista sekä parantavat testausturvallisuutta, koska virhetilanteet eivät aiheuta prototyypin rikkoutumista tai vakavia vaaratilanteita. (Lehtonen, 2006, Hanselmann 1996.) Vaikka HIL-simuloinnissa hyödynnetään fyysisiä laitteita, testauksen luotettavuus riippuu edelleen virtuaalimallin toteutuksesta. Mallinnusongelmia ovat esimerkiksi todenmukaisten anturisignaalien (esimerkiksi lämpötila) tuottaminen simulaatiomallista (Hanselmann 1996).

Virtuaalisuunnittelun ja virtuaalisten testien jälkeen testataan kaikkien osaluokkien suunnittelutietojen perusteella valmistettu ja kokoonpanttu prototyyppi (0-sarjan tuote). Todellisten prototyyppitestien tavoitteena on varmistaa, että kaikki tuotteelle asetetut vaatimukset täyttyvät myös todellisessa tuotteessa tai sen osajärjestelmässä. Todellisilla prototyypeillä testataan virtuaalisuunnittelun perusteella suunnitellut järjestelmät ja valitut komponentit erilaisissa käyttöoloissa (esimerkiksi kylmässä, kosteassa ja värinässä).

Todellisella prototyypillä testataan myös kaikki kriittiset turvatoiminnot, kuten esimerkiksi henkilönostimen hätälasku ja korin vakaus sekä ohjausjärjestelmän vikasietotilat. Näissä testeissä kiinnitetään huomiota myös tuotteen er-

gonomiaan sekä käytettävyyteen, ja havaittujen puutteiden perusteella tuotteen tehdään vaadittavat muutokset. Todelliselle prototyypille tehtävien testien suunnitelmat tulee asianmukaisesti kelpuuttaa varsinkin, jos testit liittyvät turvatoimintoihin. (ks. turvallisuusprosessin malli, luku 6.1.3).

6.2 Tuotannon suunnittelu ja valmistus

Tuotannon suunnittelussa voidaan toteuttaa digitaalista tuoteprosessia yhtä lailla kuin tuotesuunnittelussakin. Tehokkaassa tuotannon suunnittelussa voidaan hyödyntää simulointi- ja visualisointiohjelmistoja, joiden avulla on mahdollista kehittää muun muassa tuotannon eri vaiheita ja materiaalivirtoja sekä laatia tarvittaessa optimaalisten tuotantotilojen layout-suunnitelma. Teollisuusrobottien ja numeeristen työstökoneiden etäohjelmointiin soveltuvilla työkaluilla on mahdollista tuottaa ohjauskoodia CAD-mallien pohjalta sekä verifioida liikeradat ja työstöprosessit tuotannon ollessa käynnissä. Simulointityökalut säästävät siten aikaa ja tarjoavat turvallisen tavan tarkistaa esimerkiksi mahdolliset törmäykset tai muita ristiriitoja. Verifioitu ohjauskoodi voidaan myös ottaa lyhyessä ajassa käyttöön, kun todellinen tuotanto käynnistetään. Simulointien avulla on mahdollista reagoida nopeasti ja turvallisesti tuotannossa ilmeneviin muutostarpeisiin. Myös mahdolliset laiteinvestoinnit tai tuotantoprosessien muutokset pystytään simuloimaan etukäteen ja tarkistamaan niiden toimivuus ja vaikutus valmistusprosessiin.

Digitaalinen tuotannon suunnittelu pohjautuu samoihin tuotetietoihin ja -malleihin, jotka on luotu tuotesuunnittelussa. Hallittu kokonaisprosessi edellyttää selkeää tuotetiedon hallintaa. Digitaalisen tuotannon suunnittelun etuihin voidaan lukea simulointien lisäksi sen joustavat muutoksenhallinnan mahdollisuudet. Tuotetiedon päivitettävyyden ja muutosprosessien selkeä ohjaus PLM-järjestelmän avulla ovat jo nykyään erittäin hyvällä tasolla. Tämä mahdollistaa muutosprosessien hallinnan tuotteen suunnittelussa ja valmistuksessa sekä siihen osallistuvassa organisaatiossa.

MoniDigi-projektissa tuotannon suunnittelu on rajattu tutkimuksen ulkopuolelle, eikä siihen tässä syvennytä sen enempää. Koska tuotannosuunnittelu on kuitenkin merkittävä osa kokonaisvaltaista digitaalista tuoteprosessia, sen pääpiirteet on syytä tuoda tässä esille. Tuotannon suunnittelu kytkeytyy vahvasti myös tuotannon ohjausjärjestelmiin ja PLM-järjestelmiin sekä niiden väliseen rajapintaan ja sen yli tapahtuvaan tiedonsiirtoon.

6.3 Jälkimarkkinointi

Jälkimarkkinointi on tuotteen myyntitapahtuman jälkeistä tuotteeseen liittyvien palvelujen myymistä. Jälkimarkkinointi rinnastetaan myös asiakassuhteen ylläpidoksi (Jokinen ym. 2000). Nykypäivänä tuote ja sen käytönaikainen ylläpito-palvelu pyritään liittämään yhteen yrityksen liikevaihdon kasvattamiseksi (Tikkanen 2007); esimerkiksi uutta autoa ostettaessa siihen voidaan liittää huolto- ja varaosapalvelut.

Jälkimarkkinoinnin tavoitteiksi voidaan asettaa asiakkaiden tyytyväisyys, tarpeiden ja hyödykkeiden kohtaaminen, pikainen ongelmanratkaisu ja pitkäaikainen asiakkuus (Jokinen ym. 2000). Voidaankin todeta, että asiakkaisiin ja asiakassuhteisiin keskittyvä markkinointistrategia on tulosityksikkötasolla yksi tärkeimmistä toiminnoista (Tikkanen 2007). Tärkeän jälkimarkkinoinnista tekee tosiasia, että tyytyväinen asiakas asioi uudelleen samassa yrityksessä varmemmin ja suosittelee saamaansa palvelua (Lahtinen 2004). Jälkimarkkinoinnin keinoja ovat erilaiset asennuspalvelut, täsmälliset toimitukset, koulutukset, huolto-toimenpiteet, yhteydenpito asiakkaaseen, asiallisesti hoidettu laskutus ja tuotteen alkuvaiheen ongelmien ratkaiseminen (Jokinen ym. 2000, Lahtinen 2004).

Monitekniisiä tuotteita myyvän ja valmistavan yrityksen järkevällä jälkimarkkinoinnilla yksittäiseen tuotteeseen voidaan liittää useita palveluita, joiden avulla pysytään luomaan kiinteä asiakassuhde, joka toimii tuotteen myynnin jälkeenkkin. Monitekniisen laitteen käyttökoulutukset, päivitykset ja huollot ovat palveluita, joita on syytä myydä mieluummin tuotteen rinnalla kuin erikseen. PLM-järjestelmän avulla tiedetään tarkasti esimerkiksi myydyn rakenteen konfiguraatio ja ohjelmistoversiointi. Huoltojen, päivitysten ja muiden asennusten yhteydessä tehtävät muutokset on kirjattava aina PLM-järjestelmään, josta seuraavalla kerralla on luettavissa olemassa oleva tuotekonfiguraatio. Näin saadaan heti tietoon esimerkiksi, minkälainen kone asiakkaalla on ja minkälaisia varaosia siihen tarvitaan. Samalla päästään eroon aikaa vievästä tuotetiedon etsimisestä ja välttyään esimerkiksi turhilta huoltokäynneiltä, kun tuotetiedot ovat ajan tasalla.

6.4 Käytöstä poisto ja kierrätys

Käytöstä poisto ja kierrätys ovat tuotteen elinkaaren viimeinen vaihe, jossa tuote poistetaan kokonaan tai osittain käytöstä. Kiihtyvä kilpailu ja monitekniisten tuotteiden modulaarisuus mahdollistavat tuotteen eri osien kierrätyksen. Modulaaristen tuotteiden eri moduulien elinkaaret voivat olla hyvinkin erilaisia. Esi-

merkiksi työkoneen runko tai varsisto voi sellaisenaan kelvata käyttöön hyvinkin kauan, kun taas toimilaitetekniikan kehittymisen myötä toimilaitteet kannattaa uusia tiheämmin. Kierrätykseen kelpaavat niin suunnittelutiedot kuin rakenteen fyysiset osat ja ohjelmistot. Hyvin suunniteltu ohjelmisto voidaan pienin muutoksin päivittää esimerkiksi koordinoitujen ohjauksen tarkoituksiin. Koordinoituun ohjaukseen päivitettäessä tarvitaan tietysti muutoksia myös hydraulikkaan ja anturointiin.

Käytettyjen työkoneiden jälleenmyynti ja erilaiset konehuutokaupat keräävät paljon asiakkaita. Käytettyä konetta myytäessä ostajan kannalta olisi tärkeää, että koneen historia tunnettaisiin mahdollisimman tarkasti. Tuotteen elinkaaren hallintaohjelmistoista olisi periaatteessa mahdollisuudet tallennetun tuotehistorian jakamiseen uudelle omistajalle. Käytännössä järjestelmän toteuttaminen siten, että tietty koneen omistaja sitoutuisi noudattamaan huolto-ohjelmaa ja tallentamaan kaikki tapahtumat, on lähes mahdotonta. Totuus kuitenkin on se, että mitä paremmin koneen historia ja toteutunut huolto-ohjelma tunnetaan, sitä paremmin koneen jälleenmyyntiarvo säilyy.

Tässä vaiheessa tuotteen elinkaarta tuoterakenne voidaan poistaa PLM-järjestelmästä, jos sitä tai sen osia ei käytetä uudelleen sellaisenaan. Vanhat osat, jotka käytetään uudelleen tai kierrätetään, voidaan liittää suoraan uuden tuotteen tuoterakenteeseen kaikkine dokumentteineen.

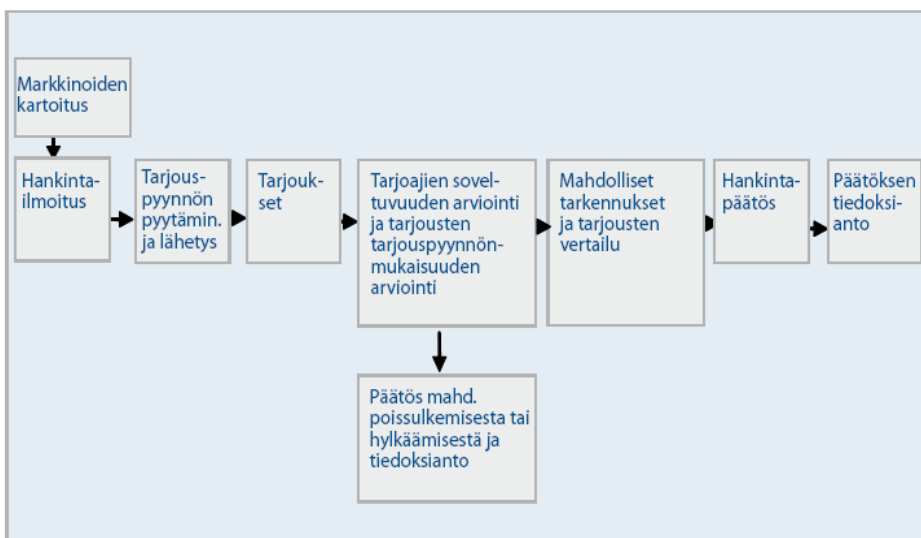
6.5 Tarjouspyynnöt ja komponenttien valinta

Tarjouspyynnön avulla yhdeltä tai useammalta toimittajalta pyydetään tarjous jostakin hankittavasta komponentista, järjestelmästä tai palvelusta. Tarkoituksena on saada vertailukelpoisia vaihtoehtoja, joista valitaan ja tilataan paras. Jotta tarjouspyyntö olisi yksiselitteinen ja tasapuolinen kaikille, on varmistettava, että tarjouspyynnössä annetut spesifikaatiot ovat riittäviä ja selkeästi esitettyjä ja että kaikki tarjouskilpailuun osallistuvat saavat samat lähtötiedot. Lisäksi on tuotava esille kriteerit, joiden perusteella hankintapäätös tehdään. Isommilla yrityksillä ja etenkin julkisilla virastoilla on hyvin yksityiskohtaiset menettelyohjeet, mutta tässä julkaisussa kyseistä aihepiiriä ei käsitellä laajemmin. Tasapuoliseen tarjouskilpailuun kuuluu myös se, että mikäli spesifikaatioita joudutaan jostain syystä tarkentamaan tai muuttamaan, siitä on tiedotettava kaikille osapuolille.

Digitaalisen tuoteprosessin yksi vaihe on tarjouspyyntöjen ja koko tarjouskilpailuprosessin hallinta. PLM-järjestelmissä on tähän kehitettyjä työkaluja ja valmiita prosesseja. Hankintamenettelyjä on erilaisia, ja menettelyn valinta riip-

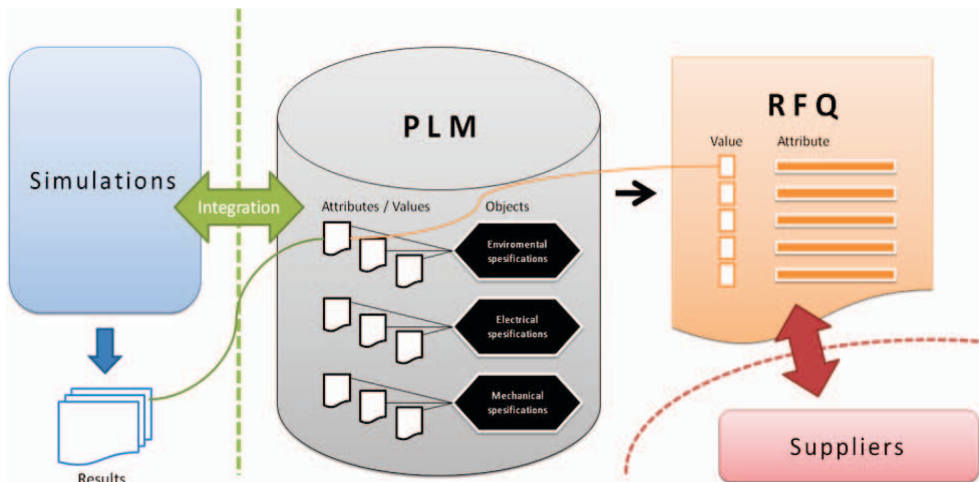
6. Digitaalinen tuoteprosessi – uusi konsepti

puu kaupan kohteesta ja luonteesta. Kuvassa 22 on esitetty Suomen valtion määräämä avoin hankintamenettelyn prosessikaavio. Siitä ilmenee tyypilliset päävaiheet, jotka ovat myös yksityisellä sektorilla tyypillisiä.



Kuva 22. Avoimen hankintamenettelyn prosessikaavio (Hytönen & Lehtomäki 2010).

MoniDigi-projektissa käytettiin ENOVIA V6:n Sourcing Management Centralin tarjoamia työkaluja ja prosesseja tarjouskilpailun läpiviemiseen. Esimerkkihankintana käytettiin puomin nivelanturia, jonka spesifikaatio määriteltiin simulointitulosten ja käyttökohteen sekä ympäristövaatimusten pohjalta. Ideaalitapauksessa simulointitulosten perusteella saatuja, komponentin valintaan liittyviä parametriarvoja hallitaan PLM-järjestelmässä esim. sopiviin objekteihin liitettyinä parametreina. Tällainen objekti voisi esim. olla ”Ympäristöolot” ja sen parametrit, kuten ”Käyttölämpötila”, ”Kosteus”, ”Tärinä” jne.



Kuva 23. PLM-järjestelmän avulla hallittu tarjouspyyntöprosessi.

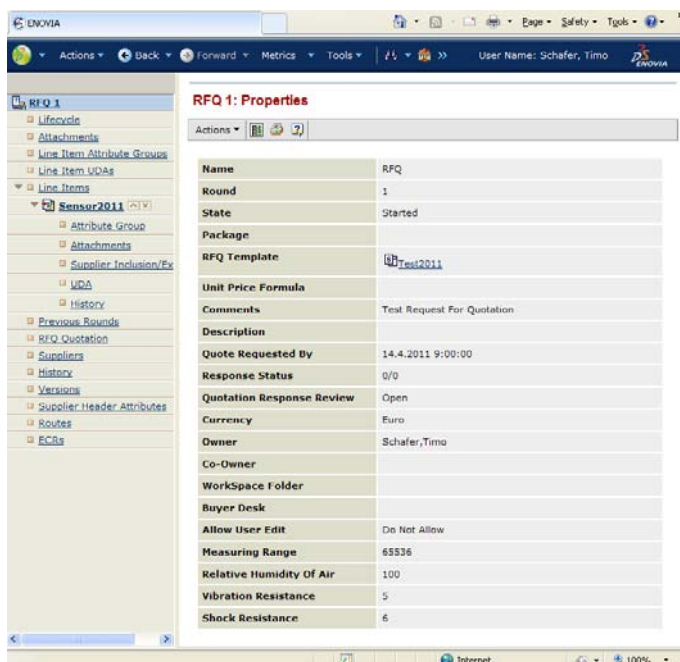
Tarjouspyyntöä (engl. Request for Quote, RFQ) koskevaan sähköiseen lomakepohjaan määritetään tarvittavat attribuutit, joiden arvot tulostetaan interaktiiviseen lomakepohjaan päivittämällä lomake (kuva 23). Tämän menettelyn etuna on se, että mikäli spesifikaatiot muuttuvat esimerkiksi tuotesuunnittelun iteraatiokierroksien vuoksi, arvot päivittyvät tarjouspyyntölomakkeeseen, ja siitä muodostuu uusi versio. Silloin arvoja ei tarvitse kirjoittaa manuaalisesti spesifikaatioon, ja näin minimoidaan riski, että jokin arvo jää vahingossa päivittämättä. Myös olemassa olevien tarjouspyyntöpohjien uudelleenkäyttö on erittäin nopeaa ja vaivatonta. Jotta tällainen menettely voidaan ottaa käyttöön, tarvitaan hyvä simulointi- ja suunnitteluohjelmistojen integraatio tuotetiedon hallintajärjestelmään. Lisäksi PLM-järjestelmän on pystyttävä tarjoamaan riittävän joustavat mahdollisuudet määrittellä tarjouspyyntöjä. Käytännössä tehdään lomakepohja (template), johon määritellään tarvittavat attribuutit. Niiden arvot luetaan simuloinneista saaduista ja PLM-järjestelmään tallennetuista tuloksista. MoniDigi-projektissa digitaalisen tuoteprosessin toteuttamisessa ei päästy näin pitkälle, joten spesifikaatio laadittiin manuaalisesti. Tarjouspyyntöön voidaan liittää lisätietoja, esimerkiksi pdf-dokumentteja tai 3D-malleja.

ENOVIA V6:n sisäänrakennetut tarjouspyyntöspesifikaatiot eivät olleet riittäviä esimerkkinä käytetylle anturin tarjouspyynnölle, mutta sopiva tarjouspyyntöpohja olisi ollut mahdollista rakentaa, mutta resurssisyydestä sitä ei tämän projektin puitteissa lähdetty toteuttamaan. Projektissa on kuitenkin luotu uusi anturiobjekti ja sille omat attribuuttinsa, kuten mittausalue ja tärinänkesto. Tarjous-

6. Digitaalinen tuoteprosessi – uusi konsepti

pyyntölomakkeeseen kyettiin valitsemaan nämä attribuutit, mutta niiden arvot määriteltiin manuaalisesti tarjouspyyntöä laadittaessa.

Kuvassa 24 on esimerkki tarjouspyynnöstä. Tarjouspyyntöön valitut attribuutit ja niiden arvot välittyivät ongelmitta toimittajille, ja siten prosessin toteuttaminen olisi mahdollista. Seuraavassa kehitysvaiheessa attribuuttien arvojen tulisi tallentua simulointituloksista ja siirtyä tarjouspyyntölomakkeeseen. Näin mahdolliset muutokset esimerkiksi simuloinnin lähtöarvoihin ja simulointituloksiin päivittyisivät myös tarjouspyyntölomakkeisiin ilman tietojen manuaalisia syöttövaiheita.



Kuva 24. Esimerkki tarjouspyynnöstä, johon on liitetty Sensor2011-niminen nimike.

Kuvassa 24 olevaan tarjouspyyntöön on liitetty kaikki ne toimittajat, joille kyseinen tarjous on tarkoitus lähettää. Kun toimittajille avataan yhteys PLM-järjestelmään, heillä on mahdollisuus katsoa, mistä osasta tai palvelusta tarjous on pyydetty ja mitkä ovat tarjouksen ehdot. Kun tarjous on valmis, se siirretään elinkaarensa seuraavaan vaiheeseen, jolloin tarjous näkyy tarjouspyynnön lähettäneen yrityksen käyttöliittymässä. Siellä eri alihankkijoiden tarjouksia voidaan verrata keskenään ja tehdä hankintapäätös. Tarvittaessa tieto valinnasta lähete-

tään kaikille alihankkijoille. PLM-järjestelmän käytön etuja tässä hankintaprosessissa on se, että tietoa ei lähetetä sähköpostin välityksellä, vaan kaikki on yhdessä järjestelmässä ja aina ajan tasalla. Jos tulee muutoksia tai tarkennuksia, tarjouspyynnöstä generoituu automaattisesti seuraava versio, joka on kaikkien saatavilla. Näin vältetään vanhentuneen tiedon käyttö ja siihen liittyvät epäselvyydet. Myös muutoshistoria jää muistiin. Tarjouspyynnön lähettävän yrityksen ei tarvitse olla yhteydessä kaikkiin potentiaalisiin toimittajiin erikseen, vaan järjestelmä hoitaa tiedotuksen automaattisesti.

6.6 Tyyppihyväksyntä

Uuden tuotteen suunnittelu- ja valmistusprosessiin saattaa liittyä hyväksyntöjä, jotta tuotetta voidaan markkinoida ja myydä. Yksittäistuotteiden hyväksynät ovat mahdollisia, mutta jos samanlaisia tuotteita on tarkoitus tehdä useita, on suositeltavaa hankkia niille tyyppihyväksyntä. Tyyppihyväksynnällä voidaan myös välttää yksittäistuotteiden varsin työläs hyväksymisprosessi. Tyyppihyväksyntöjä saavat suorittaa vain valtuutetut tahot – Suomessa esimerkiksi VTT Expert Services Oy erilaisille rakennustuotteille ja Liikenteen turvallisuusvirasto (Trafi) ajoneuvoille, ilma-aluksille, rautatiekalustolle ja vesikulkuneuvoille sekä näiden komponenteille. Uuden tuotteen tyyppihyväksyntävaatimukset on syytä selvittää jo tuotteen vaatimusmäärittelyvaiheessa, ja koko suunnitteluprosessia on ohjattava niin, että tuote täyttää nämä vaatimukset. Muun muassa nostolaitteiden tarkastuksia tekee Inspecta Oy, jolta saa palveluja suunnitteluvaiheesta alkaen. Ajatus on, että jo vaatimusmäärittely- ja suunnitteluvaiheissa tarkastukseen liittyviä seikkoja otettaisiin huomioon niin, että itse tarkastuksen yhteydessä ei syntyisi odottamattomia tekijöitä, jotka vaikeuttavat tuotteen kaupallistamista. Lisätietoja löytyy Inspectan verkkosivuilta (www.inspecta.fi).

Esimerkiksi ajoneuvojen osalta Trafian internetsivuilla kerrotaan, että ennen tyyppihyväksynnän myöntämistä hakijan on osoitettava tuotannon vaatimustenmukaisuus. Tyyppihyväksyntä voidaan myöntää koko ajoneuvolle tai ajoneuvon osalle, järjestelmälle tai erilliselle tekniselle yksikölle. Tyyppihyväksyntää hakee tuotteen valmistaja tai valmistajan edustaja. Tyyppihyväksyntää ei voi hakea yksittäiskappaleena valmistetulle tai maahantuodulle ajoneuvolle tai komponentille, vaan tällöin ajoneuvo yksittäishyväksytään tai rekisteröintikatsastetaan. Tyyppihyväksyntää voidaan hakea joko kansallisesti tai kansainvälisesti. Lisätietoja löytyy internetistä sivulta www.trafi.fi.

6. Digitaalinen tuoteprosessi – uusi konsepti

Laitteen tyyppi hyväksynnässä tarkistetaan muun muassa tuotteen tekniset dokumentit, suunnitelmat ja laskelmat sekä suoritetaan tarvittaessa käytännön koeket ja tarkastukset. Hyväksynnässä käytetään tarvittaessa tutkimuslaitosten asiantuntemusta hyväksi.

6.7 Dokumentointi

Dokumentoinnilla tarkoitetaan tässä yhteydessä teknistä viestintää sekä tuoteprosessien dokumentoimista. Teknisen viestinnän dokumentti voi olla esimerkiksi tuotteen tai palvelun kuvaus, valmistusohje, käyttö- ja huolto-ohje, kokoonpano- tai asennusohje, varaosaluettelo, koulutusmateriaali tai markkinointiaineisto. Tuoteprosessin dokumentoiminen kattaa pöytäkirjat, muistiot, vaatimuslistat, kyselytulokset, asiakirjat, sopimukset jne.

Dokumentteja on yrityksen sisäiseen käyttöön, yritysverkoston käyttöön sekä myyntiorganisaation ja loppukäyttäjien käyttöön. Kaikkien näiden dokumenttien tarkoitukset ovat hyvin erilaisia, mutta niitä myös yhdistävät useat tekijät. Dokumenttien hallinnan kannalta on äärimmäisen tärkeää, että jokaisesta dokumentista on aina käytössä vain uusin versio, joten dokumenttien versiohallintaa on aina pidettävä yllä. Muutoshistoria on hyvä säilyttää, jotta kehitysprosessi on mahdollista jäljittää myöhemmin. Tämä vaikeutuu, kun joudutaan hallinnoimaan useita eri kieliversioita ja dokumentteja on paljon. Siinä vaiheessa dokumenttien hallintajärjestelmät auttavat pitämään dokumentit järjestyksessä. MoniDigi-projektissa dokumenttienhallinta hoidettiin PLM-järjestelmällä ENOVIA V6.

Digitaalisen tuoteprosessin tuotedokumentoinnissa voidaan hyödyntää pitkälti suunnittelussa luotuja 3D-CAD-malleja. Niiden avulla saadaan erilaisiin dokumentteihin tarvittavat visuaaliset, jopa fotorealistiset, kuvaukset. Ajatus on, että itse dokumentit ovat lähinnä esivalmisteltuja pohjia, joihin on tehty layout-suunnittelu käyttäen tilavarauksia ja linkityksiä erityyppisiin elementteihin. Nämä voivat olla esimerkiksi tekstiosuoksia tai yksittäisiä lauseita, kuvia, numeerisia arvoja jne. Esimerkkinä voidaan tässä käyttää lyhyttä huolto-ohjetta. Aluksi viitataan alkusanoihin, minkä jälkeen tulee esimerkiksi kuva laitteesta. Sen jälkeen on tekstiä painesäädöstä sekä painearvotaulukko. Dokumentin rakenne olisi kuvan 25 mukainen. Ajatus interaktiivisesta lomakepohjasta lähtee siitä, että esimerkissä käytetyt tilanvaraukset (”otsikko”, ”johdanto”, ”kuva”, ”säätöohje” ja ”säätöarvot”) ikään kuin täytetään linkityksillä. Nämä linkit viittaavat tietokannassa oleviin objekteihin, joita tähän dokumenttiin halutaan tulostaa.

The diagram shows two vertical panels, labeled 1 and 2 at the bottom. Panel 1 contains three rectangular boxes: the top one is labeled 'OTSIKKO', the middle one is labeled 'JOHDANTO' and contains five horizontal lines, and the bottom one is labeled 'KUVA'. Panel 2 contains two rectangular boxes: the top one is labeled 'SÄÄTÖOHJE' and contains five horizontal lines, and the bottom one is labeled 'SÄÄTÖARVOT' and contains a 3x5 grid.

Kuva 25. Esimerkki yksinkertaisesta dokumenttipohjasta.

Oletetaan, että tietokannassa on objekti, jonka tyyppi on ”Otsikko”, jolla on yksilöllinen tunnus ja jonka arvo tai parametri on ”Huolto-ohje”. Tässä tapauksessa otsikkokenttään tehdään linkitys, joka viittaa objektityyppiin ”Otsikko” kyseisellä tunnukseella. Vastaavasti linkitetään muutkin tilanvarauskentät. Kun halutaan tulostaa dokumentti, päivitetään kyseinen lomakepohja, jolloin linkkien perusteella dokumenttiin saadaan niiden valittujen objektien viimeisimmät versiot. Tämän menettelyn etu on, että kun tehdään muutoksia malleihin, teksteihin, säätöarvoihin jne., nämä muutokset päivittyvät kaikkiin niihin sähköisiin dokumentteihin, joissa kyseisiä objekteja on käytetty. Itse dokumentteja ei siten tarvitse mennä muuttamaan. Kun dokumenttien lukemiseen käytetään tietoteknisiä päätelaitteita, ajankohtainen tieto on aina saatavana. Paperiversiot vanhentuvat muutosten tapahtuessa, joten niihin on syytä yksilöidä aina tarkka versiotieto dokumentista.

Tietokannassa olevat, dokumentteihin tulostettavat objektit on syytä linkittää myös esimerkiksi tiettyihin koneityyppeihin ja tarkemmin vielä koneyksilöihin, joita kyseinen tieto koskee. Kun tieto muuttuu, on päätettävä, onko kyseessä esimerkiksi virheellisen tiedon korjaaminen, joka koskee kaikkia koneita, vai koneeseen tehty muutos, joka koskee vain uusia koneita siitä hetkestä eteenpäin. Silloin on yksilöitävä, mistä sarja- tai konenumeroista lähtien muutos on voimassa. Tämä mahdollistaa sen, että jonkun tietyn koneyksilön dokumentteja voidaan

6. Digitaalinen tuoteprosessi – uusi konsepti

tulostaa milloin tahansa. Myös huolto-organisaatiossa on aina tieto siitä, mitä komponentteja missäkin koneessa on ja mitkä olivat niihin liittyvät säätöarvot, ohjelmistoversiot tms.

Sähköisiä dokumentteja voidaan hyödyntää lisäksi tuotannossa ja huoltotoiminnassa, kun päätelaitteille (kuten näyttöruudut, tablet-tietokoneet tai matkapuhelimet) voidaan tulostaa asennusohjeet jopa erittäin havainnollisina animaatioina. Samaan sovellukseen on mahdollista lisätä palautemahdollisuus, jolloin saadaan suunnitteluosastolle tietoa mahdollisista asennusongelmista tai muuten huomiota vaativista seikoista. Sitä kautta tuotetiedon laatua ja organisaation sisäistä tiedonvaihtoa voidaan parantaa entisestään.

Digitaalista tuotetietoa hyödyntävät tuotedokumentit voidaan 3D-mallien ansiosta laatia jo ennen ensimmäistä prototyypin, jolloin esimerkiksi asennusohjeet ovat käytettävissä heti prototyypin asennuksesta lähtien. Myös muut tuotedokumentit, kuten käyttö- ja huolto-ohjeet, voidaan laatia etukäteen, jolloin ne kulkevat mukana aina ensimmäisestä konetoimituksesta lähtien. Hyvä ja kattava dokumentaatio on myös laadunvalvonnan edellytys. Hyödyntämällä digitaalisen tuoteprosessin mukaisia toimintatapoja dokumenttien laatiminen ja hallitseminen helpottuu ja dokumenttien saatavuus on aina taattu. Samoin tuotedokumenttien historiatiedot tallentuvat ja kehityspolku on selkeästi dokumentoitu. Tieto on siten aina käytettävissä myös tulevaisuudessa, vaikka kriittiset henkilöresurssit vaihtuisivat.

6.8 PLM-järjestelmän rajapinnat muihin järjestelmiin

Digitaalisen tuoteprosessin implementointiin ja myös yleisesti yrityksen tietojärjestelmiin liittyen on korostettava erityisesti haasteita, joita yritykset kohtaavat eri tietojärjestelmien yhteensovittamisessa. On olemassa lukuisa määrä erilaisia tietojärjestelmiä, kuten EDM-, PDM-, PLM-, ERP- ja CRM-järjestelmät. Jossain vaiheessa elinkaarta yritys ottaa käyttöön näistä ensimmäisen, jolla hallitaan suunnittelutietoa, tuotantoa, asiakkuuksia tai taloutta. Riippumatta siitä, mikä näistä järjestelmistä otetaan käyttöön ensimmäisenä, yritys pyrkii yleensä hallinto- ja taloussyistä hoitamaan mahdollisimman paljon toimintoja valitun järjestelmän avulla. Hankittua järjestelmää laajennetaan myöhemmin kattamaan muitakin yrityksen toimintojen osa-alueita, ja siten rajapinnat eri järjestelmien välillä hämärtyvät. Myöhemmin laajennettujen järjestelmien toimivuus ei välttämättä vastaa todellisia tarpeita, mutta kompromissi hyväksytään kustannussyistä. Myös ohjelmistotarjoajat pyrkivät tarjoamaan kokonaisvaltaisempia ratkaisuja asiak-

kailleen ja sisällyttävät omiin järjestelmiinsä yhä enemmän erilaisia ominaisuuksia.

Kun yritys kasvaa ja toiminta laajenee, todetaan jossain vaiheessa, että yrityksessä on käytössä monia eri järjestelmiä, jotka eivät välttämättä ole lainkaan keskenään yhteensopivia. Tästä aiheutuu monenlaisia lisävaiheita tiedon käsittelyssä, koska tietoja joudutaan siirtämään manuaalisesti järjestelmästä toiseen. Tämä on suuren työmäärän lisäksi myös erittäin virhealtis vaihe, ja tiedon pitäminen ajan tasalla on sekin erittäin haastavaa tai mahdotonta. Yrityksen on syytä suunnitella mahdolliset tulevat tarpeensa järjestelmän laajentamiseksi. Järjestelmähankinnassa on mietittävä, miten järjestelmän laajennus tarvittaessa toteutetaan ja mitä työtä siitä aiheutuu. Tämä on käytännössä vaikea arvioida siinä vaiheessa, ja usein voidaan vasta jälkikäteen todeta, oliko hankintapäätös hyvä vai ei.

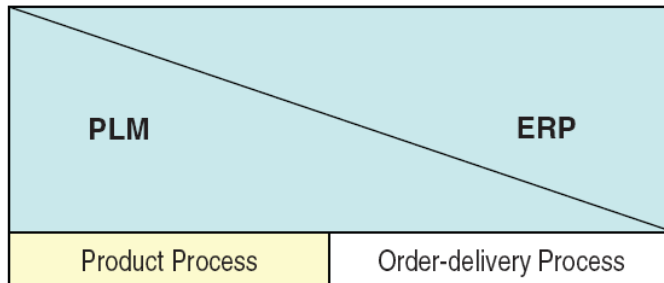
Järjestelmämuutokset ovat aina kalliita ja työläitä, joten huolellinen valintaprosessi ja tiedonkeruu on järjestelmän onnistuneen hankinnan edellytys. Tätä vaikeuttavat järjestelmien suuri valikoima markkinoilla ja niiden keskinäisen vertailun varsin työllistävä vaikutus. Järjestelmien ominaisuudet ja lähestymistavat poikkeavat siinä määrin toisistaan, että suora vertailu on miltei mahdotonta. Kuten PLM-järjestelmän valintaa koskevassa luvussa 5.4 on esitetty, valintaan vaikuttavat monenlaiset tekijät.

PLM-järjestelmän yksi rajapinta etenkin valmistavassa teollisuudessa on ERP-järjestelmä. Toiminnanohjausjärjestelmissä hallitaan tyypillisesti valmiita ja hyväksytyjä tuotteiden valmistusrakenteita (MBOM), tuotannon vaiheistusta, materiaalivirtoja sekä varastosaldoa. Järjestelmästä ja sen laajuudesta riippuen ERP-järjestelmään voi sisältyä myös ostoon ja myyntiin liittyvät toiminnot, asiakashallinta, taloushallinta jne. Usein nämä toiminnot hallitaan kuitenkin erillisissä järjestelmissä. PLM- ja ERP-järjestelmän rajapinta on kuvattu karkealla tasolla kuvassa 26. Kuvasta ilmenee, kuinka ERP-järjestelmästä tulee sitä tärkeämpi, mitä lähempänä tuotevalmistusta ollaan. Tätä rajaa ei pystytä vetämään tarkasti, mutta kuvassa on esitetty yleinen periaate. Yrityksen tuoteprosessin kannalta on erittäin tärkeää, että tuotetieto siirtyy vaivattomasti suunnittelun ja valmistuksen välillä ja että se on käytettävissä myös muualla yrityksessä, kuten myynnissä ja jälkimarkkinoinnissa. Tähän tiedonsiirtoon ja yritysprosessien integrointiin sopivat PLM-järjestelmät tarjoavat hyvän alustan. On myös tärkeää päättää, missä mitäkin tietoa hallitaan (mm. missä sijaitsee alkuperäinen tieto ja missä sen kopiot).

Yrityksillä on monesti haasteita hallita muuttuvaa tietoa ERP-järjestelmissä. Lisäksi monet päätökset, kuten raaka-ainevalinta, ympäristötekijät ja potentiaali-

6. Digitaalinen tuoteprosessi – uusi konsepti

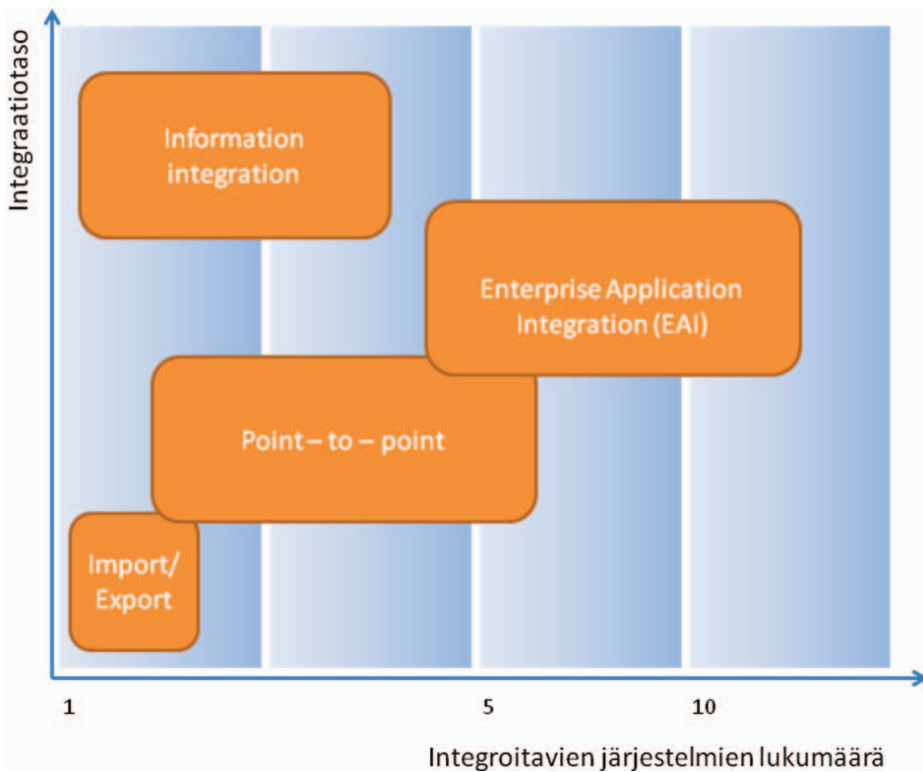
set toimittajat, on tehtävä jo suunnitteluvaiheessa. Tässä voisi olla ratkaisu hallita näitä tekijöitä PLM-järjestelmässä. Kun ne on hyväksytty, ne voidaan siirtää ERP-järjestelmään tilausta ja materiaalinhallintaa varten. Tuotetiedon siirto PLM- ja ERP-järjestelmien välillä on syytä miettiä jo järjestelmien hankintavaiheessa, koska ohjelmistojen integroitavuus keskenään edellyttää sopivien rajapintojen olemassaoloa. Nykyiset järjestelmät yleensä tarjoavat nämä rajapinnat ja integraatiokin voi tiettyjen ohjelmistojen välillä olla jopa esivalmisteltu. Tyyppillisesti integraatio vaatii kuitenkin oman projektinsa, jossa määritetään esimerkiksi se, mitä tietoa siirretään ja milloin. Yrityksessä on monesti olemassa ERP-järjestelmä, joka määrää silloin tarkasti, mitä tuotetietoa PLM-järjestelmässä vähintään täytyy olla, ja integraatio tehdään siten ERP-järjestelmän ehdoilla. Tavoitetilanne järjestelmien osalta voisi olla, että yritys suunnittelee samanaikaisesti molempien järjestelmien käyttöönottoa ja integraation järjestämistä, vaikka itse hankinnat ja käyttöönotot ajoittuisivatkin todellisuudessa eri vaiheisiin esim. resurssipulan vuoksi.



Kuva 26. PLM- ja ERP-järjestelmien roolit tuotteen elinkaaren aikana (Sääksvuori 2008, s.59).

MoniDigi-projektissa käytössä oleva ENOVIA V6 soveltuu eritasoisiin integraatiomahdollisuuksiin yrityksen muihin järjestelmiin, kuten ERP-järjestelmiin. Integraatiot voidaan jakaa karkeasti neljään eri tasoon, jotka voidaan toteuttaa myös rinnakkain. Kuvassa 27 on esitetty mahdolliset integraatiotasot sekä arvioitu, monelleko eri järjestelmälle ne soveltuisivat parhaiten. Arvio on yleisluonteinen ja sen tarkoitus on antaa kuva siitä, mitä integraatiomahdollisuuksia on olemassa. On korostettava, että päätökset integraation syvyydestä on aina tehtävä yrityskohtaisesti ja että ne riippuvat yrityksen tarpeista ja resursseista sekä järjestelmien ominaisuuksista. Esimerkiksi CAD-järjestelmä voisi olla integroitu

point-to-point-menetelmällä PLM-järjestelmään, joka taas on integroitu EAI-alustan kautta muihin järjestelmiin, kuten CRM:ään tai SCM:ään. Integraation syvyyttä rajaavia tekijöitä ovat mm. käytössä oleva budjetti, aika, henkilömäärä ja henkilöstössä oleva tietotekninen osaaminen. Kuvan 27 alapuolella kuvatut integraatiotavat on selostettu lyhyesti. Selostukset ovat lähinnä suuntaa antavia eivätkä ne ota kantaa tietoteknisiin kysymyksiin, joten yrityksen on syytä selvittää omat tarpeensa ja siihen tarjolla olevat ratkaisut asiantuntevan tahon kanssa.



Kuva 27. Järjestelmien integraatiotasot järjestelmien lukumäärään suhteutettuna.

Import/Export:

Yksinkertaisin tapa siirtää tietoa järjestelmästä toiseen. Yleensä käytetään taulukkoformaattia siirtotiedostona, jota tuotetaan yhdessä järjestelmässä ja siirretään toiseen järjestelmään esimerkiksi verkkolevyn tai sähköpostin välityksellä. Tämä käytäntö ei vaadi integraatioprojektia lainkaan, ja se soveltuu muutaman henkilön kokoiselle organisaatiolle. Menettely on kuitenkin erittäin virhealtis, ja toimintatavat on syytä määrittää tarkasti. Mi-

6. Digitaalinen tuoteprosessi – uusi konsepti

käli organisaation henkilömäärä kasvaa tai toiminta laajenee muuten, tämäntyyppinen tiedonsiirto on todennäköisesti liian työlästä. Lisäksi se ei aina noudata DTP:n kehitystä, koska tiedonsiirrossa tarvitaan manuaalisia välivaiheita.

Point-to-point:

Tässä integraatiossa linkitetään kaksi tai useampi järjestelmä suoraan keskenään. Esimerkkinä voidaan mainita CAD- ja PLM-järjestelmät. Tämä on etenkin pk-sektorilla hyvin yleinen tapa, jos on määritelty integraatorajapinnan yli kulkevan tiedon suunta, laatu ja siirtojen ajankohta. Lähes kaikissa tapauksissa tämä vaatii erillisen integraatioprojektin, joka voi sisältyä esim. PLM-järjestelmän käyttöönottoprojektiin. Integraatio voi olla yksi- tai kaksisuuntainen yrityksen tarpeista riippuen. Kaksisuuntaisessa PLM-ERP-integraatiossa tieto kulkee myös ERP-järjestelmästä PLM-järjestelmään, mikä ei ole kaikissa yrityksissä tarpeen. Se on yleensä kalliimpi toteuttaa kuin yksisuuntainen integraatio. Point-to-point-integraatio on tyypillinen valinta, kun halutaan integroida vain muutamia järjestelmiä toisiinsa. Järjestelmät toimivat edelleen itsenäisinä, mutta niiden välillä siirtyä tietoa määrättyllä tavalla.

EAI:

Enterprise Application Integration on integrointitapa, jossa voidaan linkittää useita järjestelmiä jouhevasti toisiinsa. Tyypillisesti tätä menettelyä käytetään keskisuurissa ja suurissa yrityksissä, sillä sen aiheuttamat kustannukset ovat pienemmille yrityksille monesti liian suuret. Lisäksi on eduksi, jos organisaatiossa on riittävät resurssit ja osaaminen järjestelmän ylläpitoon ja optimointiin. EAI vaatii erillisen hankittavan alustan (platform), joka linkittää eri järjestelmät yhteen ja mahdollisesti käsittelee tietoa sen siirtyessä järjestelmästä toiseen. Järjestelmät voivat tällä tavalla pyytää tietoa muualta tai käyttää toisen järjestelmän ominaisuuksia hyödyksi. Tämä integrointitapa soveltuu siihen, kun on monta eri järjestelmää ja paljon käyttäjiä (satoja tai jopa tuhansia). Eri järjestelmät käyttävät kuitenkin vielä omia tietokantojaan. EAI:n etuihin voidaan lukea myös se, että integrointialustaan ei välttämättä tarvita muutoksia, vaikka siihen liitettyihin järjestelmiin tehtäisiin muutoksia tai niitä jopa vaihdettaisiin uusiin.

Information integration:

Jossain yhteydessä käytetään myös termiä Data integration. Tässä yrityksen eri toimintoja ja palveluja sekä prosesseja integroidaan toisiinsa niin,

että käytetään vain yhtenäistä tietokantajärjestelmää. Integraatiossa eri tieto käytetään samasta lähteestä eikä siirtotiedostoja tai välillisiä alustoja tarvita. Perusjärjestelmänä voisi toimia laaja PLM-järjestelmä. Mikäli yritys ei ole kovin suuri, erillisiä tämän järjestelmän ulkopuolelle jääviä toimintoja, kuten taloushallintoa tai CRM, voidaan ostaa palveluna ”pilvestä”. Silloin järjestelmää ei osteta eikä hallita itse, vaan ostetaan käyttöaikaa palvelun tarjoajalta, joka vuorostaan hoitaa päivitykset ja IT-infrastruktuurin. Yritys selviää silloin yhden järjestelmän ylläpidolla.

7. Yhteenveto

Pienten ja keskisuurien teollisuusyritysten ydinprosesseja ovat tuote- ja tilaus-toimitusprosessit. Tuoteprosessi voidaan jakaa kahteen vaiheeseen: tuotekehitys- eli konstruktioprosessiin sekä ylläpito- ja kehitysprosessiin. Vaatimusten hallinta on tullut yhä merkittävämmäksi tuotekehitysprosessissa, koska suurin osa tuotteen vaatimuksista havaitaan tuotekehitysprosessin aikana eivätkä ne ole asiakaslähtöisiä. Useimmat tuotekehityksessä hyödynnettävät suunnittelumetodit sisältävät vaiheita, joissa asiakkaan vaatimukset saadaan määriteltyä tuotteeseen. Yleisesti käytettyjen suunnittelumetodien heikkous kuitenkin on, ettei niissä ole otettu huomioon nykyaikaisten tietojärjestelmien ja simulointiohjelmistojen mahdollisuuksia.

Parin viime vuosikymmenen aikana koneiden moniteknistyminen ja asiakas-kohtainen räätälöinti ovat lisänneet tarvetta joustavammille suunnittelumenetelmille koneenrakennuksessa. Katsaus 1990- ja 2000-luvulla suoritettuihin, TEKESin rahoittamiin teknologiaohjelmiin osoitti, että niissä tehdyllä tutkimuksella on ollut merkittävä rooli suomalaisten tutkimuslaitosten ja yritysten mallin- nus- ja simulointiosaamisen kehittämisessä. Teknologiaohjelmissä ei ole kuitenkaan onnistuttu kehittämään uusia suunnitteluperiaatteita tai -menetelmiä, jotka olisivat tehostaneet pk-yritysten tuotekehitysprosesseja merkittävästi tai muuttu- neet käytännöiksi valmistavassa teollisuudessa.

MoniDigi-projektissa toteutettiin pk-yritysten nykytilan kartoitus Suomessa, Saksassa ja Etelä-Koreassa. Tavoitteena oli selvittää, millä tasolla digitaalista tuoteprosessia tänä päivänä toteutetaan ja missä implementaation esteet ovat. Selvityksen tuloksena selvisi, että suunnittelussa hyödynnetään moderneja työ- kaluja ja tuotetaan digitaalista tuotetietoa, jota hyödynnetään osittain suoraan valmistuksessa. Tiedonsiirrossa pk-yritykset turvautuvat vielä vahvasti sähkö- posttiin ja liitetiedostoihin, mikä tuo tietynlaisia haasteita esimerkiksi tietojen versionhallintaan, tietoturvaan ja jakeluun.

Projektissa selvitettiin myös pk-yrityksen näkökulmasta tuotetiedon hallintaan liittyvät olennaiset seikat sekä tutkittiin PLM-järjestelmän valintaan vaikuttavia tekijöitä ja haasteita. Projektin käyttöön hankittiin PLM-järjestelmä, jonka avulla hallittiin projektia ja siinä syntynyttä tuotetietoa. Järjestelmän avulla tutkittiin tuotteen elinkaaren hallintaa ja siihen liittyvien prosessien soveltuvuutta monitekniselle piensarjatuotteelle.

MoniDigi-projektissa moniteknisiä piensarjatuotteita valmistaville pk-yrityksille kehitettiin uusi digitaalisen tuoteprosessin konsepti, jossa tuotteen elinkaari jaettiin neljään päävaiheeseen: tuotekehitys, valmistus, jälkimarkkinointi ja käytöstä poisto. Konseptin tavoitteena oli luoda perusta yritysten verkostomaiselle toiminnalle hyödyntäen PLM-järjestelmää, jolla hallitaan niin tuotekehitysprosesseja, tuotetietoa kuin alihankintaakin. Projektin aikana tuotepilotteihin toteutettiin virtuaalisuunnitteluvaihe, joka liittyy tuotekehityksen alkuvaiheeseen. Tästä johtuen myös projektin tuloksena esitetyssä digitaalisen tuoteprosessin konseptissa korostuivat tuotekehityksen alkuvaiheen prosessit.

Tuotekehitysprosessi jaettiin viiteen päävaiheeseen: markkinaselvitykset ja projektisuunnitelma, vaatimusmäärittely, virtuaalisuunnittelu, testaus ja verifiointi sekä turvallisuusprosessi. Turvallisuusprosessi kulkee rinnakkain tuotekehitysprojektin kanssa, ja sen tavoitteena on varmistaa, että tuote täyttää sille asetetut turvallisuusvaatimukset. Esitetyn tuotekehitysprosessin ominaisuuksia ovat kyky mukautua muuttuviin asiakasvaatimuksiin, monitekninen rinnakkainen suunnittelu ja verkottunut yhteistyö. Käytännön ongelmat verkottuneen virtuaalisuunnittelun hallinnassa johtivat prosessimalliin, jolla on yhteneväisyyksiä 1990- ja 2000-luvuilla kehitettyihin ketteriin ohjelmistokehitysmenetelmiin (Abrahamsson ym. 2002).

Asiakkaan vaatimukset tuotteelle saadaan markkinaselvityksissä, ja samalla arvioidaan myös yrityksen lähtötilanne eli suhde muihin samalla alalla toimiviin yrityksiin. Jos tuotekehitysprojekti päätetään käynnistää, aloitetaan vaatimusmäärittely, jossa määritellään tuotteen ominaisuudet ja toiminnallisuudet. Tuotevaatimukset myös priorisoidaan eli päätetään, mitkä vaatimuksista ovat pakollisia tai rajoittavia ja mitkä taas enemmän toiminnallisuutta helpottavia. Projektin aikana vaatimusmäärittely osoittautui erittäin tärkeäksi prosessivaiheeksi, koska selkeät tuotevaatimukset ja suunnittelutehtävät helpottivat virtuaalisuunnittelua.

Moniteknisyys lisää tuotekehityksen haastetta, koska laadun varmistamiseksi suunnittelua on hajautettava teknisen osa-alueen parhaiten hallitseville. Verkottuneessa toimintamallissa suunnittelutehtäviä hajautetaan yrityksen ulkopuolelle, jos yrityksellä ei ole niiden vaatimaa osaamista tai resursseja. Huolella tehdyn

7. Yhteenveto

vaatimusmäärittelyn perusteella kaikkien osapuolten on helpompi muodostaa käsitys kehitettävästä tuotteesta. Jos kaikki eivät ole täysin selvillä suunnittelun raja-arvoista ja tavoitteista, koko kehitysprojektin kannattavuus kärsii, koska suunnittelu ei etene aikataulussa eikä projektille asetettuja tavoitteitakaan välttämättä saavuteta.

Projektin aikana virtuaalisuunnittelu tehosti suunnittelutiimien välistä yhteistyötä, ja simulointimalleja voitiin hyödyntää niin tuotekonseptin kehittämisessä kuin yksityiskohtaisessa suunnittelussa. Tuotekonseptin kehittämisessä virtuaalinen 3D-prototyyppi paransi viestintää eri sidosryhmien kesken ja auttoi siten määrittelemään mahdollisimman hyvin eri vaatimuksia vastaavan tuotteen. Yksityiskohtaisessa suunnittelussa virtuaaliprototyypillä voidaan testata osajärjestelmien yhteensopivuus sekä määritellä alustavasti tuotteen pääkomponenttien ominaisuuksia.

Projektissa kehitetyssä digitaalisen tuoteprosessin konseptissa hyödynnetään mahdollisimman paljon virtuaalisia tuotemalleja ja prototyyppejä, mutta todellisilla prototyypeillä on testattava kaikki ne vaatimukset (esim. ympäristö-, turvallisuus- ja ergonomiavaatimukset), joihin virtuaaliset testit eivät sovellu. Osajärjestelmiä on mahdollista testata myös HIL- tai SIL-simuloinnilla, jos prototyypin rakentaminen koko tuotteesta todetaan liian riskialttiiksi. Erityisesti HIL-simuloinnin hyödyllisyys on kuitenkin aina harkittava tapauskohtaisesti, koska ohjelmistojen ja laitteistojen kova reaaliaikaisuusvaatimus saattaa lisätä testamisen kustannuksia merkittävästi.

Konseptin käyttöä rajaaviksi tekijöiksi voivat pk-yrityksissä muodostua virtuaalisimulointien hyödyntäminen sekä tuotetiedon hallintajärjestelmän käyttöönotto. Molemmat ovat isoja ja aikaa vieviä projekteja, jotka on tehtävä huolella. MoniDigi-projektin yritys-kyselyt osoittivat, että pk-yrityksissä virtuaalisuunnittelu rajoittuu tällä hetkellä mekaniikkasuunnitteluun (CAD) ja tekniseen laskentaan (FEM), mutta koko tuotteen kattava virtuaalimallinnus on harvinaista. Pienissä yrityksissä tähän ei ole riittävästi resursseja, mutta tehokkaasti verkottamalla vaativatkin tuotekehitysprojektit voidaan saada pk-yritysten ulottuville. Virtuaalisuunnittelun käyttöönoton riskejä ja kustannuksia voitaisiin pienentää, jos yrityksillä olisi mahdollisuus ostaa eri ohjelmistojen käyttöaikaa tarpeen mukaan. Ehdotetussa toimintamallissa yrityksillä olisi ohjelmistoihin niin sanottu verkkolisenssi, joka vaatisi verkkoyhteyden lisenssipalvelimelle. Toistaiseksi Suomesta kuitenkin puuttuvat simulointiohjelmistojen verkkolisenssejä myyvät palveluntarjoajat.

Digitaalisen tuoteprosessin tärkeä osa on tuotedokumentaatio ja sen ylläpito. MoniDigi-projektissa selvitettiin nykyaikaisia dokumentaatiomenetelmiä, jotka hyödyntävät suoraan muissa suunnitteluvaiheissa syntynyttä digitaalista tuotetietoa, kuten 3D-malleja. Digitaalisen dokumentoinnin etu on, että sitä pystytään ylläpitämään joustavasti ja hyödyntämään tehokkaasti koko yritysverkostossa. Lisäksi projektissa tutkittiin suunnittelun ja etenkin PLM-järjestelmän linkitystä valmistukseen ja tuotannonohjausjärjestelmiin. MoniDigi-projektissa selvitettiin tietojärjestelmien erilaiset integrointitavat sekä arvioitiin eri integraatiotasojen soveltuvuutta pk-yritysten muodostamalle verkostolle.

Lähdeluettelo

- Abrahamsson, P., Salo, O., Ronkainen, J. & Warsta, J. 2002. Agile software development methods. Review and analysis. Espoo 2002. VTT Publications 478. 107 s.
- Ahola, J. M., Hovila, J., Karhunen, E., Nevala, K. & Schäfer, T. 2011a. Monitekkinen tuote digitaaliseksi. Proessori 3/2011, s. 22–25.
- Ahola, J. M., Nevala, K. & Hovila, J. 2011b. Työympäristö laserkeilataan työkonoiden virtuaalisuunnitteluun. Automaatioväylä 4/2011, s. 23–25.
- Baxter, D., Gao, J., Case, K., Harding, J., Young, B., Cochrane, S. & Dani, S. 2008. A framework to integrate design knowledge reuse and requirements management in engineering design. Robotics and Computer-Integrated Manufacturing, Vol. 24 Issue 4.
- Bray, I. 2002. An Introduction to requirements engineering. Addison-Wesley. 413 s.
- Dassault Systèmes 2011a. ENOVIA Requirements Central–Datasheet.
[http://www.3ds.com/fileadmin/PRODUCTS/ENOVIA/PDF/Datasheets/enovia-
rmt.pdf](http://www.3ds.com/fileadmin/PRODUCTS/ENOVIA/PDF/Datasheets/enovia-
rmt.pdf) (25.1.2011).
- Dassault Systemes 2011b. ENOVIA Collaborative PLM.
www.3ds.com/products/enovia/portfolio/enovia-v6/overview (25.1.2011).
- Ekman, K. & Sääskilähti, M. 2002. Teollinen muotoilu tuotekehityksessä. Teknillinen korkeakoulu, konetekniikan osasto, koneensuunnittelun laboratorio. 3. korjattu painos. https://noppa.tkk.fi/noppa/kurssi/kon-41.4003/materiaali/Kon-41_4003_introduction_to_id__in_finnish_.pdf (4.5.2011).
- Ellman, A., Käppi, T., Kauranne, H., Kajaste, J., Heisanen, K. & Pietola, M. 2001. Simulation oriented R & D of hydraulically driven machines. Modelling and simulation of multitechnological machine systems. Espoo: VTT Symposium 209.
<http://www.vtt.fi/inf/pdf/symposiums/2001/S209.pdf> (7.4.2011).
- Eppinger, S. D. & Synterä, H. 2000. Improving Product Development Efficiency in Manufacturing Industries 1996–1999. Technology Programme report 4/2000. Evaluation Report. Helsinki: TEKES.
http://www.tekes.fi/fi/document/43371/rapid_evaluointi_pdf (8.4.2011).
- Hanselmann, H. 1996. Hardware-in-the-loop simulation testing and its integration into a CACSD toolset. Proceedings of the 1996 IEEE International symposium on computer-aided control system design. Dearborn, MI. September 15–18.
<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=555253> (13.4.2011).

- Heilala, J., Vatanen, S., Tonteri, H., Montonen, J., Lind, S., Johansson, B. & Stahre, J. 2008. Simulation-based sustainable manufacturing systems design. Proceedings of the 2008 Winter Simulation Conference. IEEE. s. 1922–1930.
- Hietikko, M., Malm, T. & Alanen, J. 2009. Koneiden ohjausjärjestelmien toiminnallinen turvallisuus. Ohjeita ja työkaluja standardien mukaisen turvallisuusprosessin luomiseen. [Functional safety of machine control systems. Instructions and tools for the creation of standard safety process]. Espoo: VTT Tiedotteita – Research Notes 2485. 75 s. + liitt. 14 s.
- Holopainen, T. 2001. Modelling and simulation of multitechnological machine systems. Espoo: VTT Symposium 209. <http://www.vtt.fi/inf/pdf/symposiums/2001/S209.pdf> (7.4.2011).
- Holviala, N. 2010. MASI Programme 2005–2009. TEKES Programme Report 3/2010. Final Report. http://www.tekes.fi/fi/document/44312/masi_loppuraportti_pdf (7.4.2011).
- Hytönen, T. & Lehtomäki, L. Valtion hankintakäsikirja 2010. Valtionvarainministeriö. www.vm.fi/julkaisut (12.4.2011).
- Joala, V. 2006. Laserkeilauksen perusteita ja mittauksen suunnittelu. Espoo: Leica Nilomark Oy.
- Jokinen, T., Heinämaa, L. & Heikkinen, I. 2000. Tervetuloa asiakas – Myyntityön ja asiakaspalvelun taito. Helsinki: Edita, s. 320.
- Kivikko, L. 2004. Koneenrakentaja kohtaa globalisaation. Koneenrakennusalan teknologiaohjelmien arviointi. Teknologiaohjelmaraaportti 17/2004. Väliarviointiraportti. Helsinki: TEKES. http://www.tekes.fi/fi/document/43230/masina_arviointi_pdf (8.4.2011).
- Korkealaakso, P. 2009. Real-time simulation of mobile and industrial machines using the multibody simulation approach. Doctoral dissertation. Lappeenranta University of Technology. 58 s.
- Kortelainen, J. 2006. Modular Modelling and Simulation of Mechanical Systems. TEKES, KONEMASINA. Research report: VTT-R-03984-06. Espoo: VTT. (Confidential)
- Kropsu-Vehkaperä, H. 2010. Valokynä 1/2010. "Tuoterakenne muutosten hallinnan välineenä tuotteiden elinkaaren aikana". Oulun Yliopiston PLMD2-tutkimusprojektin tuloksia.

- Kuuva, M., Airila, M. & Kivikko, L. 2001. Huomisen koneet ja järjestelmät, SMART 1997–2000. Teknologiaohjelmaraportti 1/2001. Arviointi- ja loppuraportti. Helsinki: TEKES.
- Lahtinen, J. & Isoviita, A. 2004. Markkinoinnin perusteet. Tampere: Avaintulos Oy. S. 164.
- Lalic, B., Cosic, I. & Anisic, Z. 2005. Simulation based design and reconfiguration of production systems. *International Journal of Simulation Modelling* 4, s. 173–183. www.ijsimm.com/Full_Papers/.../text4-4_173-183.pdf (23.12.2010).
- Lee, S. G., Ma, Y.-S. Thimm, G. L. & Verstraeten J. 2008. Product lifecycle management in aviation maintenance, repair and overhaul. *Computers in Industry*, Vol. 59, Issues 2–3. *Product Lifecycle Modelling, Analysis and Management*, s. 296–303. <http://www.sciencedirect.com/science/article/pii/S0166361507001108> (31.5.2011).
- Lehenkari, J., Juvonen, L., Lemola, T., Lintunen, J. & Lahtinen, H. 2010. Monialaistuva ohjelmatoiminta. MASI- ja Digitaalinen tuoteprosessi -ohjelmien arviointi. Arviointiraportti. TEKESin ohjelmaraportti 8/2010. http://www.tekes.fi/fi/document/48533/masi_ja_dtp_arviointi_pdf (8.4.2011).
- Lehtonen, M. 2006. Simulation-based design process of smart machines. Espoo: VTT Tiedotteita – Research Notes 2349. 184 s.
- Leppävuori, J., Olin, M., Valli, A., Lahti, S., Hasari, H., Koistinen, A. & Leppänen, S. 2009. SISU – Simuloinnin ja suunnittelun uudet sovellustavat ja liiketoiminta. Hyödyn elementit ja käyttöönoton prosessit: SISUQ8. Metropolia ammattikorkeakoulun julkaisuja, Sarja D: Työpaperit 1. http://www.metropolia.fi/fileadmin/user_upload/Julkaisutoiminta/METROPOLIA_sarjaD_tyopaperit.pdf (7.4.2011).
- LUT 2011. Asiakkuusajattelu. Asiakassuhteen arvonmuodostus, asiakastyytyväisyys ja asiakastarpeiden kartoitus. Powerpoint-esitys. Lappeenrannan teknillinen yliopisto. <http://www.kouvola.lut.fi/files/download/Asiakkuusajattelu.pdf> (26.1.2011).
- Malm, T. & Hämäläinen, V. 2006. Turvallisuustietoinen koneiden ja tuotantolinjojen modernisointiprosessi. Espoo: VTT Tiedotteita – Research Notes 2359.
- Mathworks 2011. SimMechanics Link Download. http://www.mathworks.com/products/simmechanics/download_smlink.html (2.8.2011)
- McIntosh, K. 1995. *Engineering Data Management*. McGraw-Hill. 279 s.

- Naur, P. & Randell, B. 1968. Software Engineering: Report of a conference. Julkaisussa: Nato Science Committee. The NATO software engineering conference 1968, October 7–11. Garmisch, Saksa.
<http://homepages.cs.ncl.ac.uk/brian.randell/NATO/nato1968.PDF> (31.5.2011).
- Pahl, G. & Beitz, W. 1992. Koneensuunnitteluoppi. 2. korjattu painos. Helsinki: Metalliteollisuuden kustannus Oy. 608 s.
- Palmberg J., Krus, P. & Janssen, A. 1995 Early prototyping in fluid power technology. The 4th Scandinavian Conference on Fluid Power. 26–29. September 1995, Tampere, s. 272–286.
- Paredis, C. J. J., Diaz-Calderon, A., Sinha, R. & Khosla P. K. 2001. Composable Models for Simulation Based Design. Engineering with Computers 17, s. 112–128.
- Parviainen, P., Hulkko, H., Kääriäinen, J., Takalo, J. & Tihinen, M. 2003. Requirements Engineering. Inventory of technologies. Espoo: VTT Publications 508. 106 s.
- Peltonen H., Martio A. & Sulonen R. 2002. PDM – Tuotetiedonhallinta. Helsinki: IT Press. 169 s.
- Puolitaival, O.-P. 2008. Adapting model-based testing to agile context. Mallipohjaisen testauksen soveltaminen ketterässä ohjelmistokehityksessä. Espoo: VTT Publications 694. 69 s. + liitt. 6 s.
- Pöyhönen, I. & Hukki, K. 2004. Riskitietoisien ohjelmiston vaatimusmäärittelyprosessin kehittäminen. Espoo: VTT Tiedotteita 2263. 36 s. + liitt. 9 s.
- Schramm, D., Franitza, D. & Lalo, W. 2005. Virtual prototyping and real-time simulation of heavy equipment manipulators including elastic deformations and hydraulics. 22nd International symposium on Automation and Robotics in Construction. ISARC 2005. September 11–14. Ferrara. Italia. 2005.
www.iaarc.org/publications/fulltext/isarc2005-26schramm.pdf (21.12.2010).
- SFS-EN ISO 12100-2 2004. Koneturvallisuus. Perusteet ja yleiset suunnitteluperiaatteet. Osa 2: Tekniset periaatteet. 2004-04-16. Helsinki: Suomen Standardisoimisliitto SFS ry. 76 s.
- SFS-EN ISO 13849-1 2007. Koneturvallisuus. Turvallisuuteen liittyvät ohjausjärjestelmien osat. Osa 1: Yleiset suunnitteluperiaatteet. 2007-06-01. Helsinki: Suomen Standardisoimisliitto SFS ry. 177 s.
- SFS-EN ISO 13849-2 2004. Koneturvallisuus. Turvallisuuteen liittyvät ohjausjärjestelmien osat. Osa 2: Kelpuutus. 2004-03-08 ed. Helsinki: Suomen Standardisoimisliitto SFS ry. 96 s.

- SFS-EN ISO 14121-1 2007. Koneturvallisuus. Riskin arviointi. Osa 1: Periaatteet. 2007-10-15. Helsinki: Suomen Standardisoimisliitto SFS ry. 64 s.
- Sommerville, I. & Sawyer, S. 1997. Requirements Engineering: A Good Practice Guide. John Wiley & Sons, s. 391.
- Syrjänen, M., Nikula, J., Hiltunen, J. & Raivio, T. 2008. Uusia tuotteita koneenrakennusalalle. MASINA-teknologiaohjelman loppuarviointi. Tekesin ohjelmaraportti 3/2008. Arviointiraportti.
http://www.tekes.fi/fi/document/42728/masina_arviointi_2008_pdf (8.4.2011).
- Sääksvuori, A. & Immonen, A. 2002. Tuotetiedon hallinta – PDM. Jyväskylä: Talentum Media. 201 s.
- Sääksvuori, A. & Immonen, A. 2008. Product lifecycle management. Berliini: Springer. 245 s.
- TEKES 2008. MASINA – Koneenrakennuksen teknologiaohjelma 2002–2007. Tekesin ohjelmaraportti 4/2008. Loppuraportti.
http://www.tekes.fi/fi/document/42730/masina_loppuraportti_pdf (7.4.2011).
- TEKES 2011. Digitaalinen tuoteprosessi 2008–2012. <http://www.tekes.fi/ohjelmat/dtp> (11.4.2011).
- Tchkalov, V. & Miller, S. 2010. Parametrization of directional and proportional valves in Simhydraulics. Mathworks Inc.
<http://www.mathworks.es/mason/tag/proxy.html?dataid=12968&fileid=63032> (31.5.2011).
- Tikkanen, H., Aspara, J. & Parvinen, S. 2007. Strategisen markkinoinnin perusteet. Helsinki: Talentum, s. 187.
- Tiusanen, R., Hietikko, M. & Alanen, J. 2007. System safety concept for remotely controlled mobile machine systems. 5th International Conference Safety Of Industrial Automated Systems (SIAS 2007). Tokyo, Japan, 12–13 Nov. 2007. Proceedings. JNIOASH; AIST; JMF; JEMA; JARA; NECA (2007), s. 52–57.
- Tiusanen, R., Hietikko, M., Alanen, J., Pátkai, N. & Venho, O. 2008. System Safety Concept for Machinery Systems. Espoo: VTT Tiedotteita 2437. 53 s.
<http://www.vtt.fi/inf/pdf/tiedotteet/2008/T2437.pdf> (31.5.2011).
- tSoft – Ohjelmistotuotannon tietokeskus 2007. Vaatimusmäärittely. Joensuun yliopisto, tietojenkäsittelytieteen laitos. <http://cs.joensuu.fi/tSoft/vaatimusmaarittely.htm> (31.5.2011).

- Tuomaala, J. 1995. Luova koneensuunnittelu. Tampere: Tammertekniikka.
- Työsuojeluhallinto 2008. Koneturvallisuus. Koneiden tekniset vaatimukset ja vaatimustenmukaisuus. Työsuojeluoppaita ja ohjeita.
<http://www.tyosuojelu.fi/fi/forward/file/1159> (31.5.2011).
- Ullman, D. G. 2009. The Mechanical Design Process. 4. painos. New York, NY: Mc Graw Hill.
- Vessonen, I. & Järviluoma, M. 2001. Simulation based design of mobile machine vibration control and active cabin suspension prototype. Modelling and simulation of multitechnological machine systems. Espoo: VTT Symposium 209.
<http://www.vtt.fi/inf/pdf/symposiums/2001/S209.pdf> (7.4.2011).
- Vuori, M. 2009. Vaatimusmäärittelyn huonoimmat käytännöt. Systemityö-lehti 2/2009. 2 s. <http://www.pcuf.fi/sytyke/lehti/kirj/st20092/ST092-20A.pdf> (8.2.2011).
- Young, R. 2004. The Requirements Engineering Handbook. Boston/Lontoo: Artech House. 275 s.

Liite A: Tutkimus myyntikonfiguraattorin kehittämisestä ENOVIA V6 -tuotetiedonhallintajärjestelmään



UNIVERSITY of OULU
OULUN YLIOPISTO

MoniDigi

Tutkimus myyntikonfiguraattorin kehittämisestä ENOVIA V6 -tuotetiedonhallintajärjestelmään

TUTKIMUSRAPORTTI

Tom Nevala 31.8.2009
Oulun yliopisto
Rakentamisteknologian tutkimusryhmä
0503200161
ossi.nevala@oulu.fi
tom.nevala@gmail.com

SISÄLLYSLUETTELO

1	TAUSTA	3
1.1	MoniDigi-projektin tausta	3
1.2	Tutkimuksen tausta.....	3
1.3	Tutkimuksen tavoite	3
2	MÄÄRITELMÄ	4
2.1	Merkinnät ja lyhenteet.....	4
2.1	Ohjelmointikielet ja työkalut.....	4
2.3	Tarvittavat asennukset ja asetukset	5
2.4	Myyntikonfiguraattori 3D-laatikolle	6
2.5	Myyntikonfiguraattori puomille	7
3	TOTEUTUS	9
3.1	Yhteys ENOVIA:an	9
3.1.1	AWE (Asynchronous Work Environment).....	9
3.1.2	MQL (Matrix Query Language).....	9
3.1.3	Ohjelmallinen yhteyden muodostaminen ENOVIA:an.....	9
3.1.4	Ohjelmallinen tiedon hakeminen ENOVIA:sta.....	10
3.1.5	Ohjelmallinen tiedon lisääminen ENOVIA:an	11
3.2	Myyntikonfiguraattorin toteuttaminen 3D-laatikolle	12
3.2.1	Uuden attribuutin lisääminen ENOVIA:an	12
3.2.2	Uuden tyylin lisääminen ENOVIA:an	12
3.2.3	Uuden osan lisääminen ENOVIA:an	12
3.2.4	Java Applet ohjelman toteuttaminen	13
3.2.5	Ohjelman asentaminen palvelimelle	21
3.3	Myyntikonfiguraattorin toteuttaminen monitoimipuomille	22
3.3.1	JSP-sivujen toteuttaminen	22
3.3.2	3DXML	29
3.3.3	JSP-sivujen asentaminen palvelimelle	30
3.4	Kaupalliset myyntikonfiguraattorit	31
3.4.1	Kaupallinen Tacton	31
3.4.2	Esimerkkejä.....	31
3.5	Yhteenvedo.....	31
4	LÄHTEET JA LIITTEET	32
4.1	Lähteet.....	32
4.2	Liitteet	32

1. TAUSTA

1.1 MoniDigi-projektin tausta

MoniDigi (Moniteknisen tuotteen digitaalisen tuoteprosessin kehittäminen tehokkaaseen piensarjatuotteiden suunnitteluun ja tuotantoon) -tutkimusprojektissa kehitetään työkoneiden ja erikoisajoneuvojen puomiratkaisuja valmistaville yrityksille moderneilla virtuaalisuunnittelun työkaluilla varustettu moniteknisten tuotteiden kehitysympäristö, digitaalisen tuoteprosessin konsepti sekä verkottunut toimintamalli, joka on sovellettavissa ja hyödynnettävissä laajasti pienissä ja keskisuurissa yrityksissä. Moniteknisessä tuotteessa tämä merkitsee tuotesuunnittelun osalta perinteisten koneenosien suunnittelun ja valmistuksen lisäksi mm. elektroniikan, ohjauksen, säädön ja ohjelmistojen suunnittelua sekä toiminnan verifiointia virtuaaliympäristössä.

Yksi digitaalisen tuoteprosessin ydinjärjestelmistä on tuotetiedon hallintajärjestelmä, jolla hallitaan kaikkea tuotteisiin liittyvää tuotetietoa erilaisissa tallennusmuodoissa. Tässä tuotteen elinkaaren aikana syntyvän, tuotetietoa hallitsevan järjestelmän nimityksenä käytetään lyhennettä PLM (Product Lifecycle Management).

Projektiryhmä on valinnut tuotetiedon hallinnan tutkimuskäyttöön ENOVIA V6 version. ENOVIA V6 on Dassault Systèmesin uusin versio tuotetiedon hallintajärjestelmästä, jonka jälleenmyyjänä Suomessa toimii Technia PLM Oy.

1.2 Tutkimuksen tausta

Projektin tuotepiloteiksi on valittu rakentamisessa käytettävä, automatisoidulla puomilla varustettu työkone sekä palo- ja pelastustehtävissä käytettävä monitoimipuomilla varustettu erikoisajoneuvo. Kummankin tuotepilotin tuotetiedon hallintaa on tarkoitus tutkia mm. mallintamalla puomien 3D-kuvat Dassault Systèmesin CATIA-ohjelman avulla, siirtämällä tuotetiedot ENOVIA:an ja toteuttamalla puomeille erilaisia tuotekonfiguraatioita ENOVIA:an integroidun Variant Configuration:in avulla. Enoviaan toteutettuja tuotekonfiguraatioita käytetään erilaisten myyntikonfiguraatioiden luomiseen selainpohjaisen käyttöliittymän avulla. Tämän osatehtävän raportissa keskitytään myyntikonfiguraattorin suunnitteluun ja toteutukseen.

1.3 Tutkimuksen tavoite

Myyntikonfiguraattoria koskevan tutkimuksen tavoitteena on tarjota yrityksille vaihtoehto toteuttaa myyntikonfiguraattorinsa itse sen sijaan, että käytettäisiin kaupallista myyntikonfiguraattoria. Kaupallisen toimittajan toteuttaman myyntikonfiguraattorin hinta saattaa nousta yllättävän korkeaksi. Siksi yrityksissä kannattaa miettiä tarkkaan myyntikonfiguraattoriin halutut ominaisuudet ja käytettävissä olevat henkilöresurssit ennen myyntikonfiguraattorin hankintaa. Jos yrityksestä löytyy tietoteknistä osaamista, kannattaa myyntikonfiguraattorin omatoimista toteuttamista harkita. Yleinen tietotekninen osaaminen ja varsinkin Internet-ohjelmointiin liittyvät (Java, JSP, Javascript, HTML, jne.) taidot ovat keskeisessä roolissa myyntikonfiguraattoria toteuttaessa. Toteuttaessa myyntikonfiguraattoria yrityksen www- sivuille on otettava huomioon myös konfiguraattorin graafinen ulkoasu. Todella näyttävien myyntikonfiguraattorien toteuttamiseen tarvitaan huippugrafiikkaa. Graafikon käyttäminen apuna on suositeltavaa, muttei välttämätöntä.

2. MÄÄRITELMÄ

2.1 Merkinnät ja lyhenteet

JSP	Java Server Pages (JSP) on Servlet-rajapintaa hyödyntävä tekniikka, joka käyttää yksinkertaista merkkäusmenetelmää XHTML/HTML-merkinnän seassa Java-koodin upottamiseksi web-sovelluksiin.
Java Applet	Java-sovelma (Java Applet) on epäitsenäinen, asiakaskoneessa selaimen yhteydessä suoritettava Java-ohjelma. Sovelmilla tehdään verkkosivuille vuorovaikutteisia ominaisuuksia, joita ei voida toteuttaa HTML:n avulla.
3DXML	3DXML on kevyt formaatti, jonka avulla 3-ulotteisen datan jakaminen ja esittäminen on nopeaa ja helppoa. 3dxml-päätteisiä tiedostoja saadaan ulos mm. Dassault Systemesin CATIA- ja ENOVIA-ohjelmista.
Java RMI	Remote Method Invocation (RMI) on Javan hajautettujen olioiden toteutustekniikka. RMI:n avulla Java-olio voi kutsua toisessa koneessa olevan Java-olion metodia.
JAR	Java Archive (JAR) on tuttuun ZIP- pakkausmenetelmään perustuva menetelmä koota esimerkiksi kaikki tiettyyn Java-applettiin liittyvät tiedostot kuten esim. kuvat ja äänet yhdeksi pakatuksi tiedostoksi.
MQL	Matrix Query Language (MQL) on kyselykieli, jolla voidaan esim. hakea, muokata ja lisätä tietoa ENOVIA:an liitetystä tietokannasta.
ORACLE	Oracle on tunnettu relaatiotietokanta.
Java Servlet	Java Servletit ovat pieniä, laitteistoriippumattomia Java ohjelmia, joita voidaan käyttää laajentamaan WWW- palvelimen toimintaa monella eri tavalla. Servletit ovat palvelimella vastaavia kuin appletit selaimella; ne laajentavat isäntäsovelluksen toimintamahdollisuuksia.
Java Beans	Java Beans (Java Pavut) mahdollistaa komponenttipohjaisten ohjelmistojen kirjoittamisen Java ohjelmointikielellä. Java Beans helpottaa mm. Java-luokan käyttämistä JSP-sivulta.

2.2 Ohjelmointikielät

ENOVIA tuotetiedon hallintajärjestelmä on koodattu pääosin Java-kielellä. Tästä syystä myös tässä tutkimuksessa on käytetty Java-ohjelmointikieltä. Samaa ohjelmointikieltä käyttämällä vältetään monilta ongelmilta, saadaan aikaan hyvä liitettävyyys ja säästetään aikaa. Java-kieli on myös mm. Java Applettien, JSP-sivujen ja RMI:n ansiosta erittäin hyvä vaihtoehto web-sovellusten tekoon. Varsinkin kun projektin käytössä on Apache Tomcat – palvelin, joka tukee vahvasti Java-kieltä. Tässä tutkimuksessa Java-kieltä on käytetty seuraavasti:

3D-laatikkoesimerkki:

- Toteutettu Java Applettina
- Käyttää eMatrixServletRMI.jar –pakettia MQL-kieltä ja RMI-kutsuja yhteyden muodostamisessa ENOVIA:an

Puomiesimerkki:

- Toteutettu JSP-sivuina
- Käyttää myös eMatrixServletRMI.jar –pakettia MQL-kieltä ja RMI-kutsuja
- Käyttää Java Beans-tekniikkaa

Tutkimuksessa on käytetty seuraavia ohjelmistoja ja työkaluja:

- Netbeans IDE 6.7.1 sovelluskehitin
- 3DXMLPlayer
- Apache Tomcat 6.0
- ENOVIA V6-tuotetiedon hallintajärjestelmä
- ORACLE-tietokanta
- Windows XP Professional ja Windows Server 2003 –käyttöjärjestelmät

NetBeans sovelluskehitin on ladattavissa ilmaiseksi osoitteesta:

<http://www.netbeans.org/downloads/index.html>. NetBeans IDE (Integrated Development Environment) sovelluskehitysympäristö tarjoaa laadukkaan, tehokkaan, monipuolisen ja ennen kaikkea ilmaisen ympäristön Java-sovellusten kehittämiseen. NetBeans on graafinen käyttöliittymä, joka tarjoaa myös mainion tuen Java Applettien ja JSP-sivujen kehittämiseen. Lisäksi NetBeans IDE-paketin mukana voidaan ladata myös Apache Tomcat 6.0, Sun GlassFish Enterprise Server v.2.1 ja Sun GlassFish Enterprise Server v3 prelude palvelimet, joilla JSP-sivujen testaaminen onnistuu. JSP tarvitsee toimiakseen JSP yhteensopivan palvelimen, kuten Apache Tomcat:in.

3DXMLPlayer on Dassault Systemesin tarjoama ilmainen ohjelmisto .3dxml –tyyppisten tiedostojen katseluun web-selaimessa. 3DXMLPlayer voidaan ladata osoitteesta:

<http://www.3ds.com/products/3dvia/3d-xml/1/>.

Apache Tomcat 6.0 on ilmainen webbipalvelin, joka tarjoaa tuen JSP sivuille ja Servleteille.

Apache Tomcat:in voi ladata osoitteesta:

<http://tomcat.apache.org/download-60.cgi>

Asennukseen voi katsoa ohjeita esimerkiksi osoitteesta:

http://stud.hamk.fi:6461/tk2/docs/OJ_14230013_HARJ/TOMCAT-ASENNUSOHJE.HTML.

MoniDigi-projektissa ENOVIA V6-tuotetiedon hallintajärjestelmä on asennettu Apache Tomcat 6.0 palvelimelle ja tietokannaksi on valittu ORACLE- tietokanta. ORACLE on maksullinen tietokanta. Toinen vaihtoehto voisi olla ilmainen MySQL-tietokanta, jota ENOVIA myös tukee. Maksullinen versio ORACLE:sta on tehokkaampi kuin ilmainen versio MySQL:stä.

Apache Tomcat 6.0, ENOVIA V6-tuotetiedonhallintajärjestelmä ja ORACLE-tietokanta on MoniDigi- projektissa asennettu Windows Server 2003 palvelintietokoneelle. Järjestelmätoimittaja Technia tekee ENOVIA:n asennuksen palvelimelle ja auttaa tarvittaessa asennukseen liittyvissä ongelmissa. Apache Tomcat ja Windows Server 2003 käyttöliittymä on asennettu projektissa kuitenkin omatoimisesti. Windows XP Professional-käyttöjärjestelmiä on käytetty projektissa asiakastietokoneina otettaessa yhteyttä Windows Server 2003 palvelimelle ja ENOVIA:an. Mitään ohjelmia/yhteyksiä ei ole testattu muilla käyttöjärjestelmillä.

2.3 Tarvittavat asennukset ja asetukset

Java on laitteisto- ja alustariippumaton kieli. Projektissa toteutetun 3D-laatikkoesimerkin käyttöä varten on asennettava päätelaitteelle uusien JRE (Java Runtime Environment). Alusta- ja käyttöjärjestelmäriippumattomuus tarkoittavat sitä, että Java toimii kaikilla käyttöliittymillä, kuten Windows, Linux, Unix Macintosh jne. Vain JRE on asennettava ensin. JRE tarjoaa Java Virtuaalikoneen, tarvittavat kirjastot sekä muut komponentit Java Applettien ajamista varten. JRE on ilmainen ja sen voi ladata osoitteesta: <http://www.java.com/en/download/manual.jsp>. Puomiesimerkkiä varten on ladattava Dassault Systemesin ilmaiseksi tarjoama 3DXML Player. Asennusohjeet löytyvät edellisestä kappaleesta.

2.4 Myyntikonfiguraattori 3D-laatikolle

Tavoitteena oli toteuttaa myyntikonfiguraattori, jonka avulla haetaan tietoa ENOVIA V6 tuotetiedon hallintajärjestelmän tietokannasta ja lisätään sinne tietoa. Myyntikonfiguraattori 3D-laatikolle on toteutettu Netbeans IDE 6.71 sovelluskehittimellä ja se on Java Appletti. 3D-laatikkoa käytetään esimerkkinä, jonka avulla testataan myyntikonfiguraattoria web-selaimella. Tavoitteessa onnistuttiin muutamien ongelmien jälkeen erittäin hyvin. Ongelmia tuotti mm. Applettien näkyminen eri tavalla eri JRE ympäristöissä ja Swing komponenttien näkyvyys. Laatikko-esimerkille toteutettiin kaksi myyntikonfiguraattoria: Swing komponenteilla varustettu (Kuva 1) ja perinteinen Appletti (Kuva 2). Swing-komponentit tarkoittavat Java Appleteissa uudempia ja tyylikkäämpiä komponentteja (nappulat, tekstikentät jne.), kuin mitä perinteisillä Appleteilla on mahdollista toteuttaa. Seuraavassa käydään läpi tämän myyntikonfiguraattorin ominaisuudet:

3D-laatikolle on toteutettu kaksi myyntikonfiguraattoria, jotka löytyvät osoitteista: <http://130.231.67.241:8081/boxsalesconfigurator1/SalesConfigurator.html> ja <http://130.231.67.241:8081/boxsalesconfigurator2/SalesConfigurator.html>.

Näitä konfiguraattoreita käytettäessä on oltava uusin JRE asennettuna ja IP-osoite yliopiston palvelimen palomuurissa.

-Konfigurointi tapahtuu valitsemalla monivalintakentistä (X, Y, Z) arvo 0.5-1.2. Nimet X, Y ja Z viittaavat koordinaatiston dimensioihin, joiden avulla määritellään laatikolle koko.

-Konfiguraatio käynnistyy, kun käyttäjä painaa Configure-nappia.

-Configure-nappia painettaessa, ohjelmasta otetaan yhteys ENOVIA V6 tuotetiedon hallintajärjestelmän ORACLE-tietokantaan MQL-käskyllä. Apuna käytetään eMatrixServletRMI.jar-pakettia, jonka voi pyytää PLM järjestelmän toimittajalta (tässä tapauksessa Technia:lta).

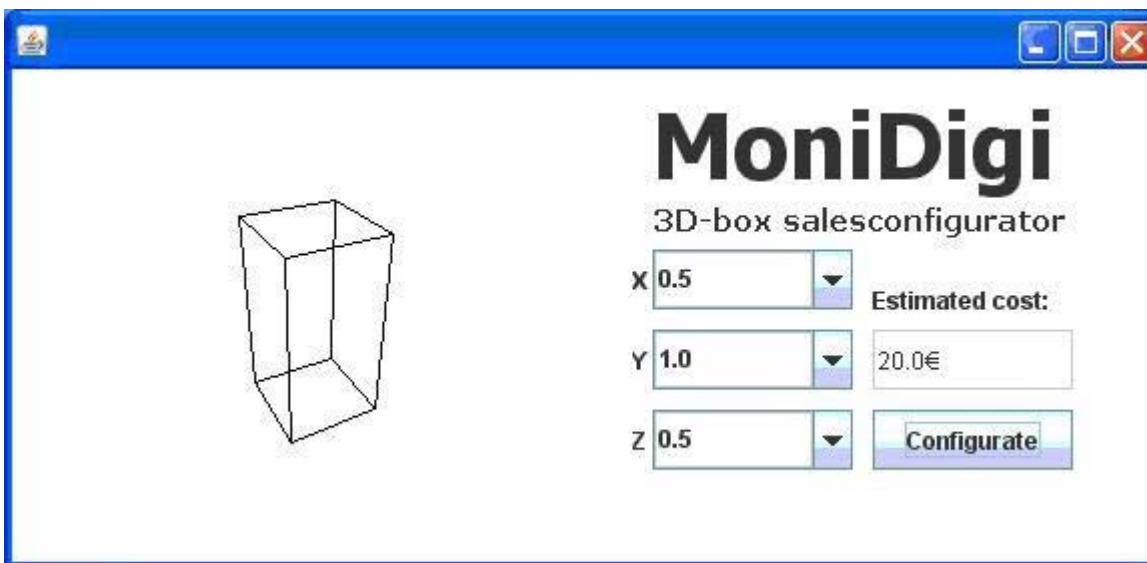
-Konfiguraatiossa asetetut arvot muokkaavat ENOVIA:an lisätyn osan attribuuttien XLength, YLength ja ZLength arvot konfiguraation mukaisiksi.

-Asetetut arvot haetaan ENOVIA:sta ja ohjelma piirtää arvojen mukaisen laatikon näytölle.

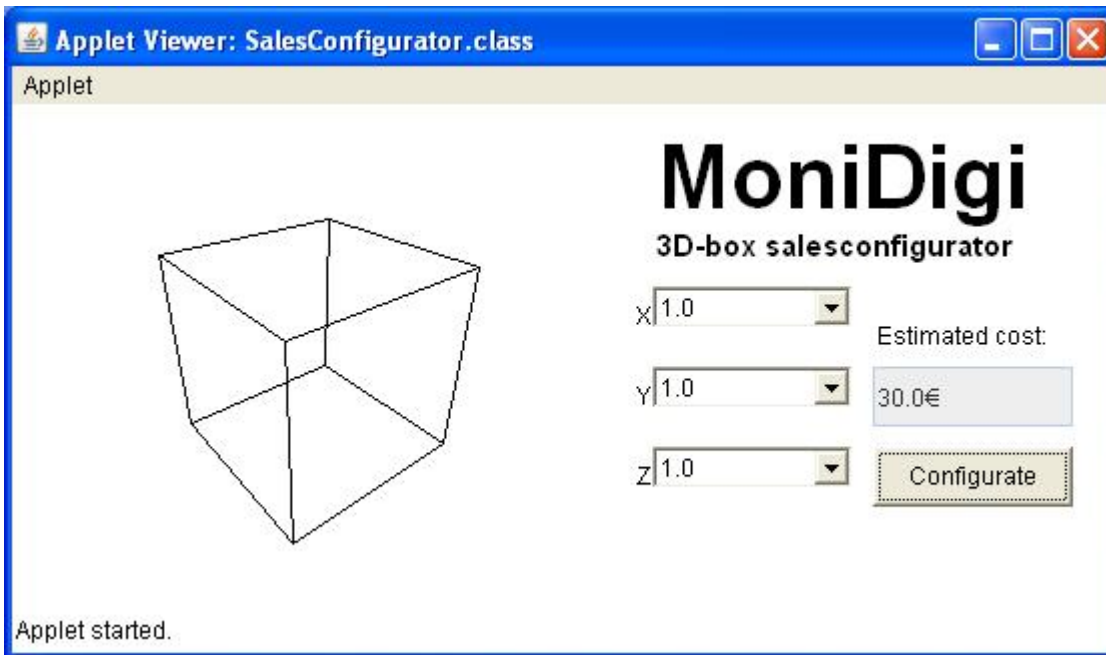
-Käyttäjä voi pyörittää konfiguroitua laatikkoa mielensä mukaan ja katsella sitä eri kuvakulmista.

-Ohjelma laskee konfiguroidulle laatikolle hinnan.

-Konfiguraatio tapahtuu nopeasti, muutamasta sekunnin kymmenesosa sekuntiin. Applettien lataaminen web selaimelle ensimmäisellä kerralla saattaa kestää muutamasta sekunnista n. kymmeneen sekuntiin.



Kuva 1. Myyntikonfiguraattori 3D-laatikolle, Swing komponentteja käyttävä Java Applet.



Kuva 2. Myyntikonfiguraattori 3D-laatikolle, perinteinen Java Applet.

2.5 Myyntikonfiguraattori puomille

Myyntikonfiguraattori puomille on toteutettu JSP-sivuina. Tämäkin myyntikonfiguraattori on toteutettu Netbeans IDE 6.71 sovelluskehittimellä. Tavoitteena oli toteuttaa myyntikonfiguraattori, jossa konfiguraatio tehdään MoniDigi-projektin pilottikohteelle, eli puomille. Tavoitteena oli myös tehdä tästä myyntikonfiguraattorista kehittyneempi ja yritysten kannalta tarkemmin ohjeistettu esimerkki kuin edellisestä. Tavoitteena oli käyttää toteutuksessa JSP- sivuja ensinnäkin sen takia, että ENOVIA V6 tuotetiedon hallintajärjestelmä on toteutettu lähes kokonaan Javalla ja toiseksi sen takia, että useimmat yritysten viralliset, Internetissä olevat myyntikonfiguraattorit ovat myös toteutettu JSP- sivuina. Tärkeä tavoite oli, että web-selaimelle piirretyn 3D-kuvan tulisi olla CATIA-kuvan tasoinen kuva. Käyttöliittymästä tehtyä konfiguraatiota vastaava kuva täytyi pystyä hakemaan ENOVIA:sta ja liittämään web-selaimelle. Tavoitteissa onnistuttiin muutamien ongelmien jälkeen erittäin hyvin. Ongelmia tuottivat aluksi mm. 3dxml-kuvan hakeminen (ohjelmallinen checkout) ENOVIA:sta ja Connector-luokan (luokka, jolla avataan yhteys ENOVIA:an) kutsuminen JSP-sivulta. Seuraavassa käydään läpi tämän myyntikonfiguraattorin ominaisuudet:

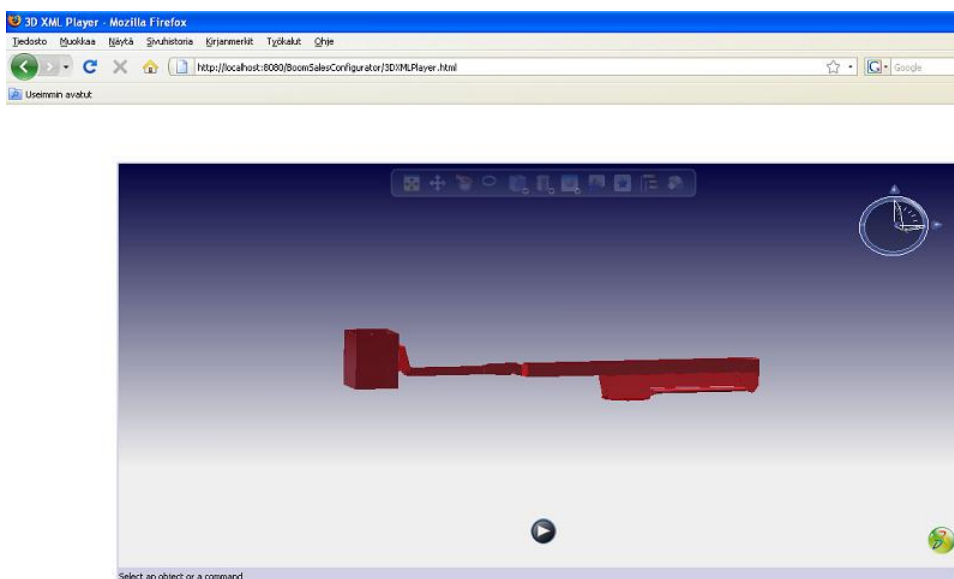
- Myyntikonfiguraattori löytyy osoitteesta: <http://130.231.67.241:8081/salesconfigurator2/index.jsp>.
- Käytettäessä tätä konfiguraattoria tarvitaan 3DXMLPlayer asennus ja IP-osoitteen lisäys yliopiston palvelimen palomuriin. Konfiguraatio tapahtuu JSP-sivuilla, radiobuttonien avulla (Kuva 3).
- Mahdollisia puomivaihtoehtoja on 8 ja vaihtoehdot ovat:

Name:	Color:	Control:	Equipment:
FogCrane1	Red	Manual control	Working platform
FogCrane2	Red	Manual control	Low pressure water monitor
FogCrane3	Red	Coordinated control	High pressure fog monitor
FogCrane4	Red	Coordinated control	Grapple
FogCrane5	Grey	Manual control	Working platform
FogCrane6	Grey	Manual control	Low pressure water monitor
FogCrane7	Grey	Coordinated control	High pressure fog monitor
FogCrane8	Grey	Coordinated control	Grapple

- Käyttäjälle näkyviä konfigurointiin käytettäviä JSP sivuja on siis kolme: index.jsp (tässä valitaan väri), control.jsp (valitaan control) equipmentSetA.jsp tai equipmentSetB.jsp (jos on valittu manual control, ohjelma ohjautuu sivulle equipmentSetB.jsp, jos taas on valittu coordinated control, ohjelma ohjautuu sivulle equipmentSetA.jsp).
- Sivujen ohjaaminen ja toimintojen toteuttaminen tapahtuu formAction.jsp sivun koodissa.
- FormAction.jsp sivulla käytetään Java Beans tekniikkaa, jonka avulla kutsutaan Java luokkaa.
- Tämän Java luokan tehtävänä on ottaa yhteyttä ENOVIA V6 tuotetiedon hallintajärjestelmään.
- Luokka on tässä esimerkissä nimeltään Connector ja se kuuluu connector pakettiin.
- Luokassa käytetään eMatrixServletRMI.jar pakettia yhteyden ottamiseen.
- Kun käyttäjä on tehnyt konfiguraation, konfiguraatiota vastaava kuva haetaan Connector-luokan koodin avulla ENOVIA:sta. Toiminto on nimeltään ohjelmallinen file checkout.
- Lopuksi ohjelma lisää juuri haetun 3dxml kuvan 3DXMLPlayer:iin. Käyttäjä voi katsella eri kulmista kuvaa juuri konfiguroimastaan tuotteesta (Kuva 4).



Kuva 3. Myyntikonfiguraattori puomille (FogCrane), ensimmäinen sivu, toteutettu JSP-sivuna.



Kuva 4. FogCrane1_CAT.3dxml –kuva 3DXMLPlayer:issä.

3. TOTEUTUS

3.1 Yhteys ENOVIA:an

Yhteys ENOVIA V6 tuotetiedonhallintajärjestelmään voidaan toteuttaa usealla eri tavalla. Ensimmäinen suositus oli se, että yhteys muodostettaisiin käyttämällä AWE:a. Toinen suositus oli, että ENOVIA:an otettaisiin yhteys Java koodin avulla, käyttämällä MQL kieltä ja apuna eMatrixServletRMI.jar –pakettia. Tässä työssä käytettiin jälkimmäistä tapaa. AWE on käyty tässä raportissa läpi hyvin suppeasti, koska tämän dokumentin aiheena on ohjelmallinen yhteys ENOVIA:an. AWE on kuitenkin otettu mukaan raporttiin, sillä se voi olla myös käyttökelpoinen tapa ottaa yhteys ENOVIA:n. Seuraavassa esitellään yhteyden muodostamista.

3.1.1 AWE (Asynchronous Work Environment)

AWE (Asynchronous Work Environment), eli tahdistamaton työympäristö on itsenäinen, erillinen, java sovellus, joka on suunniteltu käytettäväksi taustalle, pitkäaikaisiin tehtäviin. AWE tarjoaa hyvin rajoittuneen toiminnallisuuden, yleisesti vain tavan laajentaa toiminnallisuutta java koodin avulla, SPI (Service Provider Interface)-rajapinnan kautta, joka on samankaltainen tekniikka kuin web-selaimen plugin:it. AWE tarjoaa myös perusfunktiot työskentelyyn ja tietoihin käsiksi pääsyyn ENOVIA:ssa. Lisätietoja AWE:sta on esitetty raportin liitteessä (awedoc.pdf).

3.1.2 MQL (Matrix Query Language)

MQL, Matrix Query Language (ENOVIA:n aiempi nimi on ollut Matrix) sisältää samankaltaisuuksia yleisen tietokantakielen (SQL) kanssa. MQL tosin on tuntemattomampi Dassault Systemesin oma tietokantakieli, kun taas SQL on maailmanlaajuisesti tunnettu & käytetty. MQL on pääasiallinen työkalu työskentelyyn ENOVIA:n tietokannan (MoniDigi-projektissa ORACLE) kanssa. MQL:ää käytetään tietokannan asennukseen. MQL:ää voi käyttää myös tiedon hakemiseen ja lisäämiseen tietokantaan. Lisätietoja MQL:stä on esitetty raportin liitteessä (MQL108.pdf).

3.1.3 Ohjelmallisen yhteyden muodostaminen ENOVIA:an

Ohjelmallinen yhteys ENOVIA V6 tuotetiedonhallintajärjestelmään otetaan seuraavasti:

Ensin tehdään uusi projekti NetBeans IDE 6.71 sovelluskehittimeen.

File->New project->Java->Java Application->Next->Valitaan projektille nimi->Finish

Seuraavaksi tehdään uusi Java luokka tai Appletti.

Klikataan hiiren oikean puoleisella näppäimellä projektin nimen päältä->New->Java class, JApplet tai Applet (riippuu aivan siitä mitä halutaan tehdä)->Annetaan nimi->Finish

Seuraavaksi liitetään eMatrixServletRMI.jar –paketti projektiin.

Klikataan hiiren oikean puoleisella näppäimellä projektin nimen päältä->Properties->Libraries->Add JAR/Folder->Etsi eMatrixServletRMI.jar –tiedosto ja paina Open.

Seuraavaksi lisätään koodiin tarvittavat importit:

```
import matrix.db.Context;
```

```
import matrix.db.MQLCommand;
```

Tehdään luokka, jonka jäsenmuuttujaksi lisätään:

```
Context context=null;
```

Tehdään luokalle metodi ja lisätään seuraava koodinpätkä metodiin.

```
try {
    context=new Context("", "http://130.230.65.222:8081/enovia/");
    context.setUser("username");
    context.setPassword("password");
    context.connect();
}
catch (Exception e) {
    e.printStackTrace();
}
```

Koodin IP-osoitteen tilalle laitetaan IP-osoite tai verkko-osoite, johon yhteyttä ollaan ottamassa. Portin (8081) tilalle laitetaan myös se portti johon yhteyttä ollaan muodostamassa. Tekstien "username" ja "password" tilalle laitetaan luonnollisesti käyttäjätunnus ja salasana, jolla otetaan yhteys (ja jolla on oikeudet) ENOVIA:an. Kun tarvittavat tiedon lisäämiset ja hakemiset on tehty, suljetaan yhteys ENOVIA:an seuraavasti:

```
try {
    context.disconnect();
}
catch(Exception e) {
    e.printStackTrace();
}
```

3.1.4 Ohjelmallinen tiedon hakeminen ENOVIA:sta

Ohjelmallinen tiedon hakeminen ENOVIA:sta tapahtuu MQL:n avulla. Seuraava koodinpätkä liitetään kappaleessa (3.1.3) mainittuun kohtaan yhteyden avaamisen jälkeen ja ennen sulkemista.

```
try {
MQLCommand command=new MQLCommand();
    command.executeCommand(context, "print bus 'Box Part' 'ON-BOX-1' A select
attribute[XLength].value");
    uresultX=uresultX.trim();
    int ux_length=uresultX.length();
    char ux_first = uresultX.charAt(ux_length-3);
    char ux_second = uresultX.charAt(ux_length-1);
    int ux_first_i=Integer.parseInt(Character.toString(ux_first));
    int ux_second_i=Integer.parseInt(Character.toString(ux_second));
    double ux_first_d = (double)ux_first_i;
    double ux_second_d = (double)ux_second_i;
    double ux_value=ux_first_d+(ux_second_d/10);
    x_size = ux_value;
    System.out.println("XLength value is: " +xsize);
    command.close(context);
}
catch (Exception e) {
```

```
e.printStackTrace();  
}
```

Tässä esimerkki-koodinpätkässä haetaan ENOVIA:sta Box Part -tyyppisen ON-BOX-1 -nimisen osan attribuutin XLength arvo. A tarkoittaa tuotteen revisiota. Box Part -tyyppi, ON-BOX-1 -niminen osa ja XLength -niminen attribuutti on tietenkin ensin lisättävä ENOVIA:an. Se kuinka tämä tapahtuu, selitetään myöhemmin (kappaleessa 3.2). Haettu tieto tallennetaan resultX -nimiseen String (eli merkkijono) -tyyppiseen muuttujaan. Seuraavaksi haettu arvo trimmataan ja muutetaan double-tyyppiseksi. Lopuksi haettu arvo tulostetaan käyttäjälle ja MQL-yhteys suljetaan.

Tietoa haettaessa MQL-kyselyillä ENOVIA:sta on muistettava, että sanan ”bus” jälkeen kirjoitetaan ”tyyppi, nimi, revisio” seuraavasti:

'Tyyppi' 'Nimi' Revisio

Tyyppissä ja nimessä käytetään heittomerkkejä. Revisiossa niitä ei tarvitse käyttää.

3.1.5 Ohjelmallinen tiedon lisääminen ENOVIA:an

Seuraavan koodiesimerkin avulla voidaan muokata Box Part -tyyppisen, ON-BOX-1 -nimisen (revision A) osan XLength attribuutin arvo halutuksi.

```
double ix = 1.0;  
command.executeCommand(context, "mod bus 'Box Part' 'ON-BOX-1' A XLength " +ix);
```

Esimerkissä arvoksi lisätään (double) liukuluku arvo 1.0. ”bus” tarkoittaa esimerkeissä aina businessobjectia, ja ”bus” onkin MQL-kielellä alias sanasta ”businessobject”. vastaavasta ”mod” on alias sanasta ”modify” (muokkaa), ”temp” on alias sanasta ”temporary” jne.

Ohjelmallisesti MQL-käskyjen avulla voidaan ENOVIA:an tehdä myös uusi businessobjekti. Businessobjekti voi olla esimerkiksi tuote (product), osa (part), dokumentti (document) jne. Itseasiassa ENOVIA:ssa lähes kaikki muut paitsi tiedostot yms. ovat businessobjekteja. Businessobjekti tehdään MQL-käskyn avulla esim. seuraavasti:

```
add businessobject "Box Part" "ON-BOX-1" A policy "Development Part" vault "eService  
Production";
```

Tällainen kysely tehdään siis suoraan MQL-ohjelmassa palvelimella (jolle ENOVIA on asennettu). Vault tarkoittaa tietynlaista datatilaa tai aluetta ENOVIA:ssa. MoniDigi-projektissa tämän ”tilan” nimi oli juuri eService Production. Tämä nimi voi toki olla eri toisessa asennuksessa. Jos halutaan tehdä uusi tuote (product) se tehdään esim. seuraavasti:

```
add businessobject "Product" "Wrist Computer 00134" 1 policy "Development Part" vault "eService  
Production";
```

Suoraan ohjelmasta tehtynä MQL-käsky voisi olla seuraavanlainen:
command.executeCommand(context, "add businessobject 'Box Part' 'ON-BOX-1' A policy
'Development Part' vault 'eService Production' ");

3.2 Myyntikonfiguraattorin toteuttaminen 3D-laatikolle

Seuraavaksi esitetään, miten myyntikonfiguraattori 3D-laatikolle on toteutettu. Otsikot on järjestetty siten, että kappaleen (3.2) toteuttaminen alusta loppuun antaa kattavan ohjeistuksen toteuttaa tämän raportin 'Määritelmässä' esitelty myyntikonfiguraattori 3D-laatikolle. Samanlaista myyntikonfiguraattoria toteuttaessa, toteutuksen järjestys kannattaa pitää samana.

3.2.1 Uuden attribuutin lisääminen ENOVIA:an

Avataan ensin Business -niminen ohjelma palvelimelta, jolle ENOVIA on asennettu. (ks. Liite 2. Business.pdf) Valitaan Object->Find->Type. Näin saadaan näkyviin kaikki ENOVIA:n tyypit. Tyyppejä ovat esim. product, part, document jne. Valitaan myös valikosta Object->Find->Attribute ja nähdään kaikki ENOVIA:n attribuutit. Attribuutteja ovat mm. weight, name, description jne.

Aluksi tehdään ENOVIA:an kolme uutta attribuuttia Business-ohjelman avulla. Attribuutit ovat nimeltään: XLength, YLength ja ZLength. Nämä attribuutit tulevat siis sisältämään laatikon x, y ja z dimensioiden datan. Attribuutit tehdään Business-ohjelmassa seuraavasti:

Object->New->Attribute->Asetetaan kohtaan "Name" arvoksi "XLength", kohtaan "Type" arvoksi "real" ja kohtaan "Default" arvoksi "1.0". Laatikon x-dimension oletuspituus asetetaan siis arvoksi 1.0 ja tyypiksi reaalityyppi, jotta piirrettävästä laatikosta ei tulisi liian iso (kuten kävisi esim. jos käytettäisiin kokonaislukuja).->Siirrytään seuraavaan välilehteen "Ranges"->Valitaan Add->Between->Valitaan 0.4 inclusive ja 1.3 inclusive. Tämä siksi, että ohjelmallisestikin laatikon dimension minimipituus tulee olemaan 0.5 ja maksimipituus 1.2. Samalla tavalla tehdään attribuutit YLength ja ZLength.

3.2.2 Uuden tyyppin lisääminen ENOVIA:an

Uusi tyyppi tehdään myös Business-ohjelman avulla seuraavasti:

Object->New->Type->Valitaan kohtaan "Name" tyyppille nimeksi "Box Part", kohta "Description" jätetään tyhjäksi ja valitaan kohtaan "Derived From" "Part". Valitaan siten, että tämä uusi tehty Box Part-tyyppi perii ominaisuutensa (siis attribuuttinsa) Part-tyypistä. Uusia attribuutteja pystyy toki Box Part tyyppille lisäämään. ->Siirrytään Attributes välilehdelle->Add->XLength->Add->YLength->Add->ZLength->Create

3.2.3 Uuden osan lisääminen ENOVIA:an

Kun tehty Box-Part-tyyppi on ENOVIA:ssa ja siihen kuuluvat attribuutit XLength, YLength ja ZLength, tehdään Box Part-tyyppinen osa ENOVIA:an. Osa tehdään seuraavasti:

Avataan MQL palvelimelta, jolle ENOVIA on asennettu ja kirjoitetaan seuraava MQL-koodi sinne:

```
add businessobject "Box Part" "ON-BOX1" "A" policy "Development Part" vault "eService Production";
```

Osa voidaan tehdä myös suoraan ENOVIA:sta. Palvelin on kuitenkin käynnistettävä uudelleen, sillä muutokset eivät näy ENOVIA:ssa ennen sitä.

3.2.4 Java Applet ohjelman toteuttaminen

Käynnistetään NetBeans IDE 6.71 sovelluskehitin->File->New Project->Valitaan Java ja Java Application->Next->Valitaan projektille nimeksi esim. BoxSalesConfigurator->Otetaan ruksi pois kohdasta "Create main class"->Finish->Projekti aukeaa NetBeans:in Projects-ikkunaan.->Avataan projektia hieman vasemman puoleisesta "+"-napista-> Klikataan hiiren oikealla napilla Source packages-pakettia->New->Other->Java->Applet->Next->Annetaan Appletille nimeksi SalesConfigurator->Finish.

Liitetään eMatrixServletRMI.jar -paketti projektiin.

Klikataan hiiren oikean puoleisella napilla projektin nimen päältä->Properties->Libraries->Add JAR/Folder->Etsi eMatrixServletRMI.jar -tiedosto ja paina Open.

Seuraavaksi lisätään itse koodi. Koodia on esimerkissä sen verran paljon, ettei yksittäisiä koodinpätkiä tässä selitetä. Koodin pitäisi toimia tällaisenaan, kun valitaan NetBeans:issa SalesConfigurator.java hiiren oikean napin klikkauksella ja valitaan Run. Tärkeää on kuitenkin muistaa laittaa oikea käyttäjätunnus, salasana, IP-osoite ja portti esimerkkien tilalle.

Seuraavalla koodinpätkällä saadaan toteutettua perinteinen Applet (Kuva 2). Kuvan 1 mukainen Swing-komponentteja käyttävä Applet saadaan toteutettua käytännössä lisäämällä J jokaisen komponentin (Button, Label) ja Appletin eteen. Myös projektia luotaessa on hyvä luoda uusi JApplet Appletin sijasta.

```
import javax.swing.*;
import java.awt.event.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import matrix.db.Context;
import matrix.db.MQLCommand;
import java.lang.Object.*;
import java.applet.*;
import javax.swing.JFrame;

/**
 *
 * @author Ossi Nevala
 */

class Point3D {
    public double x, y, z;
    public Point3D(double X, double Y, double Z) {
        x = X; y = Y; z = Z;
    }
}

class Edge {
    public int a, b;
    public Edge(int A, int B) {
        a = A; b = B;
    }
}
```

}

```

public class SalesConfigurator extends Applet implements MouseListener,
    MouseMotionListener, ActionListener {

    Button start = new Button();
    Button close = new Button();
    Label printLabel = new Label();

    Context context=null;
    //Statement stat;
    private int xcounter = 0;
    private int ycounter = 0;
    private int zcounter = 0;
    private String[] combox_inc = { "0.5", "0.6", "0.7", "0.8", "0.9", "1.0", "1.1", "1.2" };
    private String[] comboy_inc = { "0.5", "0.6", "0.7", "0.8", "0.9", "1.0", "1.1", "1.2" };
    private String[] comboz_inc = { "0.5", "0.6", "0.7", "0.8", "0.9", "1.0", "1.1", "1.2" };
    double x_size = 1.0, y_size = 1.0, z_size = -1.0;
    Button drawButton = new Button("Configurate");
    Label labelX;
    Label labelY;
    Label labelZ;
    Label labelPrice;
    Label name;
    Label monidigi;
    Choice comboX;
    Choice comboY;
    Choice comboZ;
    JTextField username;
    JPasswordField password;
    JTextField tf;
    int width, height;
    int mx, my;
    Image boximage;
    Graphics box;
    int azimuth = 35, elevation = 30;
    Point3D[] point3D;
    Edge[] edges;

    public SalesConfigurator() { }

    public void init() {
        try {
            jbInit();
        }
        catch(Exception e) {
            e.printStackTrace();
        }
    }
}

```

```

public void jbInit() throws Exception
{

    try {
        context=new Context("", "http://130.231.67.241:8081/enovia/");
        context.setUser("username");
        context.setPassword("password");
        context.connect();
    }
    catch (Exception e) {
        e.printStackTrace();
    }

    //Container cp = getContentPane();
    //cp.setLayout(null);
    this.setLayout(null);
    labelX = new Label("X");
    labelY = new Label("Y");
    labelZ = new Label("Z");
    labelPrice = new Label("Estimated cost:");
    comboX = new Choice();
    comboY = new Choice();
    comboZ = new Choice();
    // drawButton = new JButton("Configure");
    tf = new JTextField();

    labelX.setBounds(new Rectangle(310,90,10,30));
    comboX.setBounds(320,90,100,30);
    labelY.setBounds(310,130,10,30);
    comboY.setBounds(320,130,100,30);
    labelZ.setBounds(310,170,10,30);
    comboZ.setBounds(320,170,100,30);
    labelPrice.setBounds(430,100,100,30);
    tf.setBounds(430,130,100,30);
    tf.setEditable(false);
    drawButton.setBounds(430,170,100,30);
    drawButton.addActionListener(this);

    name = new Label("3D-box salesconfigurator");
    monidigi = new Label("MoniDigi");

    name.setFont(new Font("verdana", Font.BOLD, 15));
    name.setBounds(320,50,230,40);
    monidigi.setFont(new Font("tahoma", Font.BOLD, 45));
    monidigi.setBounds(320,10,230,50);
    int i;

    for (i = 0; i < 8; i++) {
        comboX.addItem(combox_inc[xcounter++]);
    }
}

```

```
        comboY.addItem(comboy_inc[ycounter++]);
        comboZ.addItem(comboz_inc[zcounter++]);
    }

    xcounter=0;
    ycounter=0;
    zcounter=0;
    i=0;

    width = 300;
    height = 250;

    point3D = new Point3D[8];
    add3Dpoints(x_size, y_size, z_size);

    edges = new Edge[ 12 ];
    edges[0] = new Edge( 0, 1 );
    edges[1] = new Edge( 0, 2 );
    edges[2] = new Edge( 0, 4 );
    edges[3] = new Edge( 1, 3 );
    edges[4] = new Edge( 1, 5 );
    edges[5] = new Edge( 2, 3 );
    edges[6] = new Edge( 2, 6 );
    edges[7] = new Edge( 3, 7 );
    edges[8] = new Edge( 4, 5 );
    edges[9] = new Edge( 4, 6 );
    edges[10] = new Edge( 5, 7 );
    edges[11] = new Edge( 6, 7 );

    this.setSize(550,250);
    add(monidigi, null);
    add(name, null);
    add(labelX, null);
    add(labelY, null);
    add(labelZ, null);
    add(comboX, null);
    add(comboY, null);
    add(comboZ, null);
    add(drawButton, null);
    add(labelPrice, null);
    add(tf, null);
}

public void actionPerformed(ActionEvent evt) {

    Object source = evt.getSource();
    if (source==drawButton) {
```

```

Object oix = comboX.getSelectedItem();
String strix = oix.toString();
double dx = Double.valueOf(strix).doubleValue();
Object oiy = comboY.getSelectedItem();
String striy = oiy.toString();
double dy = Double.valueOf(striy).doubleValue();
Object oiz = comboZ.getSelectedItem();
String striz = oiz.toString();
double dz = Double.valueOf(striz).doubleValue();
updateValues(dx, dy, dz);
drawPic();
double xPrice = x_size*10;
double yPrice = y_size*10;
double zPrice = -(z_size)*10;
double resultPrice = xPrice+yPrice+zPrice;
tf.setText(+resultPrice+"â,-");
}
}

void add3Dpoints(double xx_size, double yy_size, double zz_size) {
    point3D[0] = new Point3D( -xx_size, -yy_size, zz_size );
    point3D[1] = new Point3D( -xx_size, -yy_size, -zz_size );
    point3D[2] = new Point3D( -xx_size, yy_size, zz_size );
    point3D[3] = new Point3D( -xx_size, yy_size, -zz_size );
    point3D[4] = new Point3D( xx_size, -yy_size, zz_size );
    point3D[5] = new Point3D( xx_size, -yy_size, -zz_size );
    point3D[6] = new Point3D( xx_size, yy_size, zz_size );
    point3D[7] = new Point3D( xx_size, yy_size, -zz_size );
}

void drawPic() {

    //repaint();
    boximage = createImage(width, height);
    box = boximage.getGraphics();
    drawBox(box);
    addMouseListener(this);
    addMouseMotionListener(this);
}

void drawBox(Graphics g) {

    double theta = Math.PI * azimuth / 180.0;
    double phi = Math.PI * elevation / 180.0;
    float cosT = (float)Math.cos(theta);
    float sinT = (float)Math.sin(theta);
    float cosP = (float)Math.cos(phi);
    float sinP = (float)Math.sin(phi);
    float cosTcosP = cosT * cosP;
    float cosTsinP = cosT * sinP;

```

```

float sinTcosP = sinT * cosP;
float sinTsinP = sinT * sinP;

Point[] points;
points = new Point[point3D.length];
int j;
int scaleFactor = width/4;
float near = 3;
float nearToObj = 1.5f;
for (j=0; j<point3D.length; j++) {
    double xx0 = point3D[j].x;
    double yy0 = point3D[j].y;
    double zz0 = point3D[j].z;
    float x0 = (float)xx0;
    float y0 = (float)yy0;
    float z0 = (float)zz0;
    float x1 = cosT*x0 + sinT*z0;
    float y1 = -sinTsinP*x0 + cosP*y0 + cosTsinP*z0;
    float z1 = cosTcosP*z0 - sinTcosP*x0 - sinP*y0;
    x1 = x1*near/(z1+near+nearToObj);
    y1 = y1*near/(z1+near+nearToObj);
    points[j] = new Point(
        (int)(width/2 + scaleFactor*x1 + 0.5),
        (int)(height/2 - scaleFactor*y1 + 0.5)
    );
}
//Jos kaytat Swing-komponentteja, poista seuraavasta koodinpatkasta kommenttimerkit ja
//kommentoi g.setColor(Color.white);
/*
float h = 0;
float s = 0;
float b = 0;
float[] hsbvals= {h,s,b};
Color.RGBtoHSB(238, 238, 238, hsbvals);
g.setColor( Color.getHSBColor(hsbvals[0], hsbvals[1], hsbvals[2]) );
*/
g.setColor(Color.white);
g.fillRect( 0, 0, width, height );
g.setColor( Color.black );
for ( j = 0; j < edges.length; ++j ) {
    g.drawLine(
        points[ edges[j].a ].x, points[ edges[j].a ].y,
        points[ edges[j].b ].x, points[ edges[j].b ].y
    );
}
}
}

public void mouseEntered( MouseEvent e ) { }
public void mouseExited( MouseEvent e ) { }
public void mouseClicked( MouseEvent e ) { }

```

```

public void mousePressed( MouseEvent e ) {
    mx = e.getX();
    my = e.getY();
    e.consume();
}

public void mouseReleased( MouseEvent e ) { }
public void mouseMoved( MouseEvent e ) { }
public void mouseDragged( MouseEvent e ) {
    int new_mx = e.getX();
    int new_my = e.getY();
    azimuth -= new_mx - mx;
    elevation += new_my - my;
    drawBox( box );
    mx = new_mx;
    my = new_my;
    repaint();
    e.consume();
}

public void update( Graphics g ) {
    g.drawImage( boximage, 0, 0, this );
}

public void paint( Graphics g ) {
    update( g );
}

public void updateValues(double ix, double iyy, double izz) {
    try {
        MQLCommand command2=new MQLCommand();
        command2.executeCommand(context, "mod bus 'Box Part' 'ON-BOX1' A XLength " +ix);
        command2.executeCommand(context, "mod bus 'Box Part' 'ON-BOX1' A YLength " +iyy);
        command2.executeCommand(context, "mod bus 'Box Part' 'ON-BOX1' A ZLength " +izz);
        command2.executeCommand(context, "print bus 'Box Part' 'ON-BOX1' A select
attribute[XLength].value");
        String uresultX=command2.getResult();
        uresultX=uresultX.trim();
        int ux_length=uresultX.length();
        char ux_first = uresultX.charAt(ux_length-3);
        char ux_second = uresultX.charAt(ux_length-1);
        int ux_first_i=Integer.parseInt(Character.toString(ux_first));
        int ux_second_i=Integer.parseInt(Character.toString(ux_second));
        double ux_first_d = (double)ux_first_i;
        double ux_second_d = (double)ux_second_i;
        double ux_value=ux_first_d+(ux_second_d/10);
        x_size = ux_value;

        command2.executeCommand(context, "print bus 'Box Part' 'ON-BOX1' A select
attribute[YLength].value");
    }
}

```

```

String uresultY=command2.getResult();
uresultY=uresultY.trim();
int uy_length=uresultY.length();
char uy_first = uresultY.charAt(uy_length-3);
char uy_second = uresultY.charAt(uy_length-1);
int uy_first_i=Integer.parseInt(Character.toString(uy_first));
int uy_second_i=Integer.parseInt(Character.toString(uy_second));
double uy_first_d = (double)uy_first_i;
double uy_second_d = (double)uy_second_i;
double uy_value=uy_first_d+(uy_second_d/10);
y_size = uy_value;

    command2.executeCommand(context, "print bus 'Box Part' 'ON-BOX1' A select
attribute[ZLength].value");
    String uresultZ=command2.getResult();
    uresultY=uresultZ.trim();
    int uz_length=uresultZ.length();
    char uz_first = uresultZ.charAt(uz_length-4);
    char uz_second = uresultZ.charAt(uz_length-2);
    int uz_first_i=Integer.parseInt(Character.toString(uz_first));
    int uz_second_i=Integer.parseInt(Character.toString(uz_second));
    double uz_first_d = (double)uz_first_i;
    double uz_second_d = (double)uz_second_i;
    double uz_value=uz_first_d+(uz_second_d/10);
    z_size = -(uz_value);

    command2.close(context);

    //x_size=ixx;
    //y_size=iyy;
    //z_size=-izz;

    add3Dpoints(x_size, y_size, z_size);
    repaint();

}
catch (Exception e) {
    e.printStackTrace();
}
}

public void destroy() {
    try {
        context.disconnect();
    }
    catch(Exception e) {
        e.printStackTrace();
    }
}
}
}

```


3.2.5 Ohjelman asentaminen palvelimelle

Ohjelma asennetaan palvelimelle (jolle ENOVIA on asennettu) seuraavasti:

NetBeans IDE 6.71 kääntää automaattisesti Java-tiedostot ennen ajamista, joten jos ohjelma on ajettu NetBeans sovelluskehittämissä, siitä on myös (.class) luokkatiedostot valmiina NetBeansProjects -kansiossa. NetBeansProjects -kansio löytyy oletuksena Omat tiedostot -kansion alta. NetBeansProjects -kansio avataan BoxSalesConfigurator kansio ja siirrytään build->classes kansioon. Tästä kansioista kopioidaan koko kansion sisältö, eli kaikki kansiossa olevat luokat. Kansiossa pitäisi olla tiedostot: SalesConfigurator.class, Edge.class ja Point3D.class.

Jos ei käytetä NetBeans tai mitään muutakaan sovelluskehittäjä, kohdassa 3.2.4 esitelty ohjelma pitää tallentaa .java päätteiseksi tiedostoksi, esim. muistiossa ja kääntää eMatrixServletRMI.jar –paketin kanssa toimiakseen. Kääntämisen voi tehdä mm. komentokehötteen avulla: Run->cmd->Ok->

```
javac -classpath C:\work\java\eMatrixServletRMI.jar;. SalesConfigurator.java
```

Missä C:\work\java kuvaa polkua, jossa eMatrixServletRMI.jar tiedosto on. Samassa kansiossa on oltava myös SalesConfigurator.java –tiedosto. Java kääntäjää, eli javac-komentoa käytettäessä on oltava Java SDK asennettuna tietokoneelle. Tätä käännoästä ei tarvitse tehdä erikseen jos käytetään jotain (kuten esim. NetBeans) sovelluskehittäjä.

Kopioidut luokkatiedostot siirretään palvelimelle, jolle ENOVIA on asennettu. Tarkemmin sanottuna tiedostot siirretään Apache Tomcat sovelluspalvelimen webapps kansioon tehtyyn alakansioon. Alakansion nimeksi voidaan laittaa esimerkiksi boxsalesconfigurator. Webapps –kansio palvelimella voi löytyä esim. osoitteesta:

```
C:/Program Files/Apache Software Foundation/Tomcat 6.0/webapps/
```

Tai:

```
C:/apps/apache_tomcat/webapps/
```

Riippuen siitä, että mihin kansioon Apache Tomcat on asennettu.

Kun luokkatiedostot on lisätty boxsalesconfigurator –kansioon, tehdään tähän samaan kansioon HTML-sivu, johon Java Applet liitetään. HTML-sivun voi tehdä esim. muistiosta tallentamalla tiedoston .html päätteellä. HTML-sivuun tulee seuraava koodi:

```
<html><head><title>MoniDigi salesconfigurator for 3D-Box</title></head>
<body>
<APPLET CODE="SalesConfigurator.class" archive="eMatrixServletRMI.jar" width="580"
height="280"></APPLET>
</body></html>
```

Sivu pitää tallentaa samalla nimellä, mikä luokallakin on, eli nimellä: SalesConfigurator.html.

Kansioon lisätään vielä yksi tiedosto, eMatrixServletRMI.jar. Boxsalesconfigurator –kansiossa pitäisi nyt olla tiedostot: SalesConfigurator.class, Edge.class, Point3D.class, eMatrixServletRMI.jar ja SalesConfigurator.html. Jos nämä tiedostot ovat kansiossa, konfiguraattorin pitäisi toimia.

3.3 Myyntikonfiguraattorin toteuttaminen puomille

Tällaista myyntikonfiguraattoria toteuttaessa, on tärkeää, että ENOVIA V6 tuotetiedon hallintajärjestelmässä on .3dxml päätteiset CATIA- kuvat puomista. Nämä .3dxml kuvat tallennetaan aina businessobjektin yhteyteen, eli yleensä dokumenttiin. .3dxml kuvat tallennetaan tiedostoksi ("File") dokumentin alle. Jokaista mahdollista puomikonfiguraatiota vastaava kuva on oltava ENOVIA:ssa. Tässä MoniDigi-projektin esimerkissä on kahdeksan erilaista mahdollista puomikonfiguraatiota, joten ENOVIA:an lisätyt .3dxml -kuvat ovat nimeltään:

FogCrane1_CAT.3dxml
FogCrane2_CAT.3dxml
FogCrane3_CAT.3dxml
FogCrane4_CAT.3dxml
FogCrane5_CAT.3dxml
FogCrane6_CAT.3dxml
FogCrane7_CAT.3dxml
FogCrane8_CAT.3dxml

Myyntikonfiguraattoria tehtäessä on hyvä ladata myös Internetistä ilmainen 3DXMLPlayer jo mahdollisimman aikaisessa vaiheessa ja perehtyä sen toimintaan.

3.3.1 JSP-sivujen toteuttaminen

Käynnistetään NetBeans IDE 6.71 sovelluskehitin->File->New Project->Java Web->Web application->Next->Asetaan kohtaan Project name nimeksi esimerkiksi BoomSalesConfigurator->Next->Jätä oletusarvot (GlassFish v3 Prelude Domain)->Finish

Laajennetaan projektia "+" napista vasemmalta->Laajennetaan Web Pages kansiota->klikataan hiiren oikeanpuoleisella napilla kansiota->New->JSP->Annetaan tiedostolle nimeksi index.jsp->Finish

Tehdään aluksi konfiguraatiosivu puomin värille (katso kuva 3). Lisätään tämä koodinpätkä index.jsp sivun oletuskoodin tilalle:

```
<% @page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">

<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>MoniDigi Boom salesconfigurator</title>
  </head>
  <body>
    <font face="tahoma" size="25"><b>MoniDigi</b></font>
    <h1>Boom Color:</h1>
    <form action="formAction.jsp" method="post">
      <input type="radio" name="boomcolor" value="red" checked> Red </input><br>
      <input type="radio" name="boomcolor" value="grey"> Grey </input><br>
      <input type="submit" value="Submit">

```

```
</form>
</body>
</html>
```

Tehdään taas uusi JSP-sivu. Annetaan sille nimeksi control.jsp. Lisätään seuraava koodinpätkä control.jsp sivulle:

```
<% @page contentType="text/html" pageEncoding="UTF-8"% >
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">

<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>MoniDigi Boom salesconfigurator</title>
  </head>
  <body>
    <h1>Control:</h1>
    <form action="formAction.jsp" method="post">
      <input type="radio" name="boomcontrol" value="joint spot control" checked> Manual
control </input><br>
      <input type="radio" name="boomcontrol" value="coordinated control"> Coordinated control
</input><br>
      <input type="submit" value="Submit">
    </form>
  </body>
</html>
```

Tehdään equipmentSetA.jsp –niminen JSP-sivu ja lisätään seuraava koodi sinne:

```
<% @page contentType="text/html" pageEncoding="UTF-8"% >
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">

<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>MoniDigi Boom salesconfigurator</title>
  </head>
  <body>
    <h1>Equipment:</h1>
    <form action="formAction.jsp" method="post">
      <input type="radio" name="equipmenta" value="fog cannon" checked> High pressure fog
monitor </input><br>
      <input type="radio" name="equipmenta" value="clearance grabber"> Grapple </input><br>
      <input type="submit" value="Submit">
    </form>
  </body>
</html>
```

Tehdään equipmentSetB.jsp –niminen sivu ja lisätään seuraava koodi sinne:

```
<% @page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">

<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>MoniDigi Boom salesconfigurator</title>
  </head>
  <body>
    <h1>Equipment:</h1>
    <form action="formAction.jsp" method="post">
      <input type="radio" name="equipmentb" value="personnel basket" checked> Working
platform </input><br>
      <input type="radio" name="equipmentb" value="low pressure water cannon"> Low
pressure water monitor </input><br>
      <input type="submit" value="Submit">
    </form>
  </body>
</html>
```

Seuraavaksi tehdään JSP-sivu, jossa otetaan vastaan edellisten sivujen käskyt ja lähetykset. Seuraava sivu on nimeltään formAction.jsp ja se toteutetaan seuraavasti:

```
<% @page contentType="text/html" pageEncoding="UTF-8"%>
<% @page language="Java"%>
<% @page import="connector.Connector"%>
<jsp:useBean id="con" class="connector.Connector" scope="request"/>
<jsp:setProperty name="con" property="*" />
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">

<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>formAction</title>
  </head>
  <body>

    <%!boolean jointControl = false;%>
    <%!boolean cordinatedControl = false;%>
    <%!boolean boomRed = false;%>
    <%!boolean boomGrey = false;%>

    <%!String filename1 = "FogCrane1_CAT.3dxml";%>
    <%!String filename2 = "FogCrane2_CAT.3dxml";%>
    <%!String filename3 = "FogCrane3_CAT.3dxml";%>
    <%!String filename4 = "FogCrane4_CAT.3dxml";%>
    <%!String filename5 = "FogCrane5_CAT.3dxml";%>
```

```

<%!String filename6 = "FogCrane6_CAT.3dxml";%>
<%!String filename7 = "FogCrane7_CAT.3dxml";%>
<%!String filename8 = "FogCrane8_CAT.3dxml";%>

<%!String docname1 = "DOC-0000014";%>
<%!String docname2 = "DOC-0000015";%>
<%!String docname3 = "DOC-0000016";%>
<%!String docname4 = "DOC-0000017";%>
<%!String docname5 = "DOC-0000010";%>
<%!String docname6 = "DOC-0000012";%>
<%!String docname7 = "DOC-0000013";%>
<%!String docname8 = "DOC-0000011";%>

<%

if(request.getParameter("boomcolor") != null) {
    if(request.getParameter("boomcolor").equals("red")) {
        boomRed=true;
        boomGrey=false;
        response.sendRedirect("control.jsp");
    }
    else if(request.getParameter("boomcolor").equals("grey")) {
        boomRed=false;
        boomGrey=true;
        response.sendRedirect("control.jsp");
    }
}
if(request.getParameter("boomcontrol") != null) {

    if(request.getParameter("boomcontrol").equals("joint spot control")) {
        jointControl=true;
        cordinatedControl=false;
        response.sendRedirect("equipmentSetB.jsp");
    }
    else if(request.getParameter("boomcontrol").equals("cordinated control")) {
        jointControl=false;
        cordinatedControl=true;
        response.sendRedirect("equipmentSetA.jsp");
    }
}
if(request.getParameter("equipmentb") != null) {
    if(request.getParameter("equipmentb").equals("personnel basket") && jointControl==true
&& boomRed==true) {
        //out.println("Joint control was selected<BR>");
        //out.println("Personnel basket was selected<BR>");
        //Connector c1 = new Connector();
        con.setConnection(filename1, docname1);
        response.sendRedirect("3DXMLPlayer.html");
    }
}

```

```

else if(request.getParameter("equipmentb").equals("low pressure water cannon") &&
jointControl==true && boomRed==true) {
    //out.println("Joint control was selected<BR>");
    //out.println("Low pressure water cannon was selected<BR>");
    //Connector c2 = new Connector();
    con.setConnection(filename2, docname2);
    response.sendRedirect("3DXMLPlayer2.html");
}
else if(request.getParameter("equipmentb").equals("personnel basket") &&
jointControl==true && boomGrey==true) {
    //out.println("Joint control was selected<BR>");
    //out.println("Low pressure water cannon was selected<BR>");
    //Connector c2 = new Connector();
    con.setConnection(filename5, docname5);
    response.sendRedirect("3DXMLPlayer5.html");
}
else if(request.getParameter("equipmentb").equals("low pressure water cannon") &&
jointControl==true && boomGrey==true) {
    //out.println("Joint control was selected<BR>");
    //out.println("Low pressure water cannon was selected<BR>");
    //Connector c2 = new Connector();
    con.setConnection(filename6, docname6);
    response.sendRedirect("3DXMLPlayer6.html");
}
}
if(request.getParameter("equipmenta") != null) {
    if(request.getParameter("equipmenta").equals("fog cannon") && coordinatedControl==true
&& boomRed==true) {
        //out.println("Cordinated control was selected<BR>");
        //out.println("Fog cannon was selected<BR>");
        //Connector c3 = new Connector();
        con.setConnection(filename3, docname3);
        response.sendRedirect("3DXMLPlayer3.html");
    }
    else if(request.getParameter("equipmenta").equals("clearance grabber") &&
coordinatedControl==true && boomRed==true) {
        //out.println("Cordinated control was selected<BR>");
        //out.println("Clearance grabber was selected<BR>");
        //Connector c4 = new Connector();
        con.setConnection(filename4, docname4);
        response.sendRedirect("3DXMLPlayer4.html");
    }
    else if(request.getParameter("equipmenta").equals("fog cannon") &&
coordinatedControl==true && boomGrey==true) {
        //out.println("Cordinated control was selected<BR>");
        //out.println("Clearance grabber was selected<BR>");
        //Connector c4 = new Connector();
        con.setConnection(filename7, docname7);
        response.sendRedirect("3DXMLPlayer7.html");
    }
}

```

```
else if(request.getParameter("equipmenta").equals("clearance grabber") &&
cordinatedControl==true && boomGrey==true) {
    //out.println("Cordinated control was selected<BR>");
    //out.println("Clearance grabber was selected<BR>");
    //Connector c4 = new Connector();
    con.setConnection(filename8, docname8);
    response.sendRedirect("3DXMLPlayer8.html");
}
}
%>
</body>
</html>
```

MoniDigi-projektin esimerkissä on siis tallennettu ENOVIA:an DOC-0000010, DOC-0000011 jne. Nimisiä dokumentteja, joiden jokaisen alla on yksi .3dxml -tiedosto, esim. FogCrane5_CAT.3dxml.

Koodinpätkä:

```
<jsp:useBean id="con" class="connector.Connector" scope="request"/>
<jsp:setProperty name="con" property="*" />
```

Tarkoittaa, että sivulla käytetään Java Beans:eja. Luokkaa (class) käytetään otettaessa yhteys ENOVIA:an. ENOVIA:an yhteyttä ottavaa koodia ei kannata missään nimessä käyttää suoraan JSP-sivulta, se todennäköisesti ei toimisikaan.

Connector -niminen luokka tehdään siis connector -nimisen paketin alle samaan projektiin seuraavasti:

Tuplaklikkaa Source Packages -pakettia->klikka hiiren oikealla napilla Source Packages -pakettia->New->Java package->Anna nimeksi connector->Finish->Klikkaa uutta connector pakettia hiiren oikealla napilla->New->Java Class->Anna nimeksi Connector->Finish->Lisää luokkaan seuraava koodi:

```
package connector;

//import matrix.db.BusinessObject;
//import matrix.db.Context;
import java.util.logging.Level;
import java.util.logging.Logger;
import matrix.db.*;
import matrix.util.MatrixException;

public class Connector {
    public Connector() {
        super();
    }
    public void setConnection(String file, String doc) throws MatrixException {
        String filename;
        filename = file;
```

```
boolean lockedornot = false;
String format = ("generic");
String directory = ("C:/Program Files/Apache Software Foundation/Tomcat
6.0/webapps/salesconfigurator2/model/");

Context context=null;
try {
    context=new Context("", "http://130.230.68.240:8081/enovia/");
    context.setUser("username");
    context.setPassword("password");
    context.connect();
}
catch (Exception e) {
    e.printStackTrace();
}

//BusinessObject bus1 = new BusinessObject();
String x= ("Document");
String y;
y = doc;
String z= ("0");
String bb=("eService Production");
BusinessObject bus2 = new BusinessObject(x,y,z,bb);

try {
    bus2.open(context);
    bus2.checkoutFile(context, lockedornot, format, filename, directory);
    bus2.close(context);
} catch (MatrixException ex) {
    Logger.getLogger(Connector.class.getName()).log(Level.SEVERE, null, ex);
    ex.printStackTrace();
}
try {
    context.disconnect();
}
catch(Exception e) {
    e.printStackTrace();
}
}
}
```

Seuraavaksi lisätään eMatrixServletRMI.jar –paketti projektiin.

Klikataan hiiren oikean puoleisella napilla projektin nimen päältä->Properties->Libraries->Add JAR/Folder->Etsi eMatrixServletRMI.jar –tiedosto ja paina Open.

Tehdään uusi kansio projektisivuille->Hiiren oikean napin klikkaus Web pages -kansion päällä->New->Folder->Anna nimeksi: model

Seuraavaksi lisätään sivut 3dXMLPlayer:eille.

3.3.2 3DXML

Hiiren oikean napin klikkaus projektinimen päällä->New->HTML->Anna nimeksi 3DXMLPlayer->Tallenna seuraava koodi sivulle:

```
<html>
<head>
<title>3D XML Player</title>
</head>
<script src="script/embed.js"></script>
<body style="MARGIN: 0px">
<table width='100%' height='100%'>
<tr colspan=3 height='10%'>
<td></td>
</tr>
<tr height='80%'>
<td width='10%'></td>
<td width='80%'>
<script language="javascript">
insertPlayer("model/FogCrane1_CAT.3dxml", "viewer", "100%", "100%");
</script>
</td>
<td width='10%'></td>
</tr>
<tr colspan=3 height='10%'>
<td></td>
</tr>
</table>
</body>
</html>
```

Tee samalla tavalla kaikki kahdeksan 3DXMLPlayer HTML-sivua.
Sivut ovat nimeltään:

3DXMLPlayer.html
3DXMLPlayer2.html
3DXMLPlayer3.html
jne.

Muuta aina kohdan insertPlayer("model/FogCrane1_CAT.3dxml nimeksi sivulukua vastaava nimi, esim. 3DXMLPlayer2.html -> FogCrane2_CAT.3dxml

Etsi lataamastasi 3DXMLPlayer ohjelman asennuskansioista 3dxml-player esimerkkiä. Löydät CATSettings -nimisen kansion. kopioi kansio kokonaisuudessaan NetBeansProjects->BoomSalesConfigurator->web kansion ja build->web kansion alle myös. Lisää myös kansio projektiin, jos se ei muuten näy.

Seuraavaksi klikataan hiiren oikean puoleisella napilla projektin päältä->Clean and build klikataan hiiren oikean puoleisella napilla projektin päältä->Run JSP-sivujen pitäisi nyt olla valmiit.

3.3.3 JSP-sivujen asentaminen palvelimelle

JSP-sivujen asentaminen palvelimelle tapahtuu seuraavasti:

Selvitä ensin, missä palvelimen Apache Tomcat:in webapps-kansio on.
Kansio voi olla esim. seuraavissa poluissa:

C:/Program Files/Apache Software Foundation/Tomcat 6.0/webapps/
Tai:

C:/apps/apache_tomcat/webapps/

Riippuen siitä, että mihin kansioon Apache Tomcat on asennettu.

Mene Omat tiedostot->NetBeansProjects->BoomSalesConfigurator kansioon
build->web->Valitse kaikki->kopioi kaikki esim. muistitikulle

Kopioi muistitikulta kopioitu web-kansio johonkin webapps -kansioon alakansioon, esimerkiksi
C:/apps/apache_tomcat/webapps/boomsalesconfigurator/
-kansioon. Kansiossa pitäisi olla nyt seuraavat tiedostot ja kansiot:

Kansiot:

CATSettings
model
META-INF
WEB-INF
script

Tiedostot:

index.jsp
control.jsp
equipmentSetA.jsp
equipmentSetB.jsp
formAction.jsp
3DXMLPlayer.html
3DXMLPlayer2.html
3DXMLPlayer3.html
3DXMLPlayer4.html
3DXMLPlayer5.html
3DXMLPlayer6.html
3DXMLPlayer7.html
3DXMLPlayer8.html

Käynnistä vielä palvelin uudestaan ja asennuksen pitäisi olla valmis. Jos edellä mainitut tiedostot löytyvät kansioista ja ohjelmat & asennukset on tehty ohjeiden mukaan, puomiesimerkin myyntikonfiguraattorin pitäisi nyt toimia.

3.4 Kaupalliset myyntikonfiguraattorit

Myyntikonfiguraattoreiden toteuttaminen vaatii suunnittelua, aikaa ja rahaa. Tämä raportti antaa erittäin hyvän pohjan myyntikonfiguraattorin toteuttamiselle käytettäessä ENOVIA V6 tuotetiedon hallintajärjestelmää. Toinen vaihtoehto on ostaa kaupallinen myyntikonfiguraattori.

3.4.1 Kaupallinen Tacton

Tacton on Ruotsalainen myyntikonfiguraattoreita tuottava yritys.

Katso tarkempaa tietoa Tacton:ista osoitteesta

www.tacton.com

3.4.2 Esimerkkejä

Esimerkkejä myyntikonfiguraattoreista löytyy esim. Tacton:in kotisivuilta ja hyvä esimerkki tyylikkäästä myyntikonfiguraattorista löytyy osoitteesta: <http://www.man-mn.fi>

3.5 Yhteenvedo

Edellä annettujen ohjeiden perusteella on mahdollista toteuttaa tämän raportin 'Määritelmä'-osiossa esitellyt myyntikonfiguraattorit 3D-laatikolla ja puomille. <http://www.man-mn.fi>-sivuilla on samalla tavalla JSP- sivuina tehty myyntikonfiguraattori, kuin tämän raportin esimerkissä. Suurin ero on siinä, että tässä toteutettu myyntikonfiguraattori tekee suoria hakuja ja muutoksia tuotetiedon hallintajärjestelmään, kun taas <http://www.man-mn.fi>-sivuilla konfiguraattori ei todennäköisesti tee suoria hakuja/muutoksia yrityksen tuotetiedon hallintajärjestelmään. Todennäköisesti <http://www.man-mn.fi>-sivuilla tehdyn konfiguraation tiedot menevät sähköpostiin tai tiedostoon, josta vasta myyntivahvistuksen jälkeen tehdään konfiguraatio ja myydään tuote.

Se, että MoniDigi- projektissa toteutettu konfiguraattori hakee suoraan tietoa ENOVIA:sta ja lisää tietoa sinne, on ehdottomasti tämän tutkimuksen kulmakivi. Todellisessa toteutuksessa ei ole useimmiten suositeltavaa antaa käyttäjän tehdä suoria hakuja/suoria muutoksia tuotetiedon hallintajärjestelmään. Halutun konfiguraation voisi lähettää esim. sähköpostiin ja Java ohjelman tekemään haun tai lisäyksen ENOVIA:an kun asiakkaan kanssa on sovittu myyntivahvistuksen jälkeen tuotteen myynnistä. Myyntikonfiguraattoria voi kyllä käyttää myös suoraan hakuun ENOVIA:sta. Esimerkiksi .3dxml kuvan hakeminen ENOVIA:sta voi olla hyödyllistä, jos halutaan mallintaa käyttäjälle kuva konfiguraatiosta jo ennen tuotteen ostamista/myymistä.

Jos kaupallinen myyntikonfiguraattori tulee hinnaltaan korkeaksi ja yrityksen henkilökunnasta löytyy tarvittavaa tietotaitoa ja osaamista, kannattaa harkita omaa myyntikonfiguraattorin toteutusta. Tässä raportissa esitettyjen esimerkkien ja ohjeiden avulla myyntikonfiguraattori on toteutettavissa kohtuullisen helposti ENOVIA V6 järjestelmään. Myyntikonfiguraattorien jatkokehityksessä voidaan keskittyä enemmän graafiseen ulkoasuun ja esimerkiksi edellä esittämäni idean toteuttamiseen tuotteen konfiguroimisesta tuotetiedonhallintajärjestelmään vasta myyntivahvistuksen jälkeen. Hyvän grafiikan lisäämisellä ja myyntikonfiguraattorin muokkaamisella yritysکوhtaisiin tarpeisiin raportissa esitettyjen perusteiden avulla, saadaan toteutettua erittäin tyylikkäitä myyntikonfiguraattoreita.

4. LÄHTEET JA LIITTEET

4.1 Lähteet

<http://www.dgp.toronto.edu/~mjmcguff/learn/java/>

<http://www.dgp.toronto.edu/~mjmcguff/learn/java/11-3d/>

JSP-ohjelmointi, DOCENDO, Tero Ahonen, Tapio Hämeen-Anttila

Inside Enterprise Java, IT Press, Pekka Niskanen, Mikko Kontio, Kimmo Vierimaa

4.2 Liitteet

Liite 1: awedoc.pdf

Liite 2: Business108.pdf

Liite 3: MQL108.pdf

Liite 4: adk10600.zip

Liite B: Matlab Simhydraulics -mallinnusesimerkki

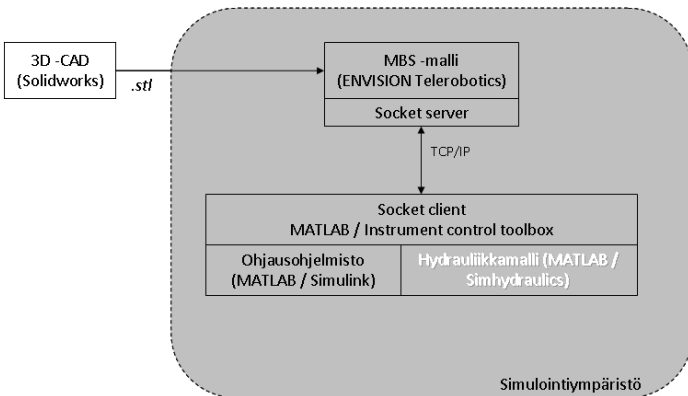
MoniDigi-projektin yhtenä tavoitteena oli selvittää moniteknisten tuotteiden mallinnukseen soveltuvia graafisia ohjelmointiympäristöjä. Hydrauliiikan mallinnuksessa kaupallisia ohjelmistovaihtoehtoja ovat EASY5 (Msc Software, Santa Ana, Kalifornia) ja MATLAB Simhydraulics -lisäosa (Mathworks Inc., Natick, Massachusetts). Tutkimusryhmän kokemuksen mukaan EASY5-ohjelmiston käyttöön liittyy kuitenkin lukuisia ongelmia, kuten suppea komponenttikirjasto ja komponenttimallien konfiguroinnin hankaluus, joten MoniDigi-projektissa päätettiin selvittää tarkemmin MATLAB Simhydraulics -ohjelmiston soveltuvuutta moniteknisen tuotteen mallinnukseen.

Tässä esimerkissä käydään läpi hydraulisesti ohjatun mekanismin mallintamisen periaate ENVISION TR:n ja MATLAB Simhydraulicsin avulla. Simulointiympäristön arkkitehtuuri on kuvassa B1 a. Socket-palvelin ohjelmoitiin C-kielellä ENVISIONin jaettuun kirjastoon, jossa suoritettiin myös nivelen kinematiikkaan liittyvä laskenta. Yksinkertaistuksen takia hydrauliiikkamallia ei tehty koko puomille, vaan mallinnuskokeiluun valittiin pelastuspuomin sylinteriohjatun alapuomin nostonivel (kuva B1 b).

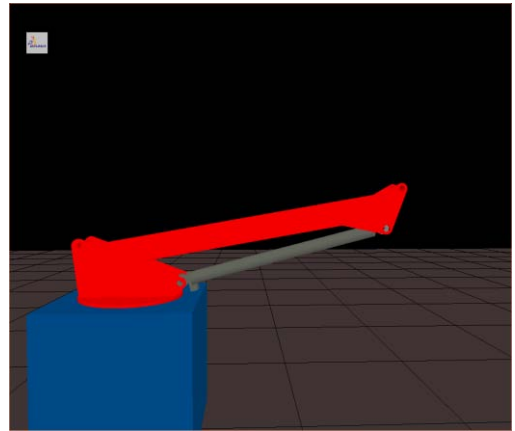
Kuvassa B1 c on MATLAB Simulinkiin ohjelmoitu hydrauliiikkamalli. Komponenttimalleissa on useita liitäntöjä eri suureille. Esimerkiksi hydraulisynterinin mallissa on neljä liitäntää: A ja B hydraulisille suureille (paine, virtaus) sekä C ja R mekaanisille suureille (voima, asema, nopeus). Venttiilimallissa on puolestaan viisi liityntää: portit A ja B toimilaiteliitännöille, P syöttöpaineelle ja T tankkilinjalle sekä S ohjaussignaali. Venttiilimalli parametrisoitiin Tchkalovin ja Millerin (2010) esittämällä menettelyllä, jossa venttiilille oletetaan lineaarinen paine-virtauskäyrä ja virtausaukon koko asetetaan siten, että nimellistä painehäviötä vastaava virtaus vastaa mallinnettavan venttiilin spesifikaatiota. Simulink-ohjelma lukee socketista männänvarren nopeuden ja palauttaa sylinterin tuottaman voiman.

Liite B: Matlab Simhydraulics -mallinnusesimerkki

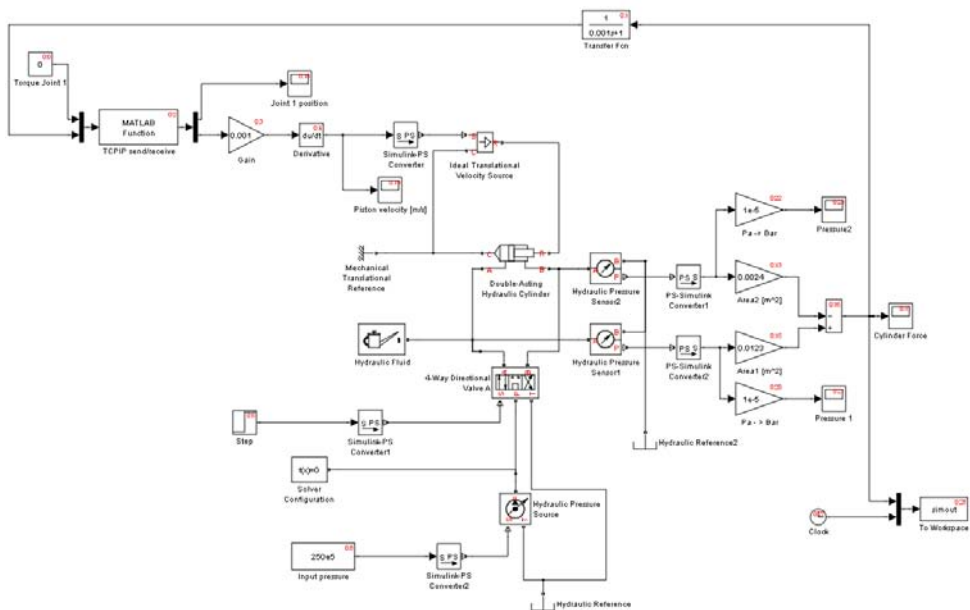
a)



b)



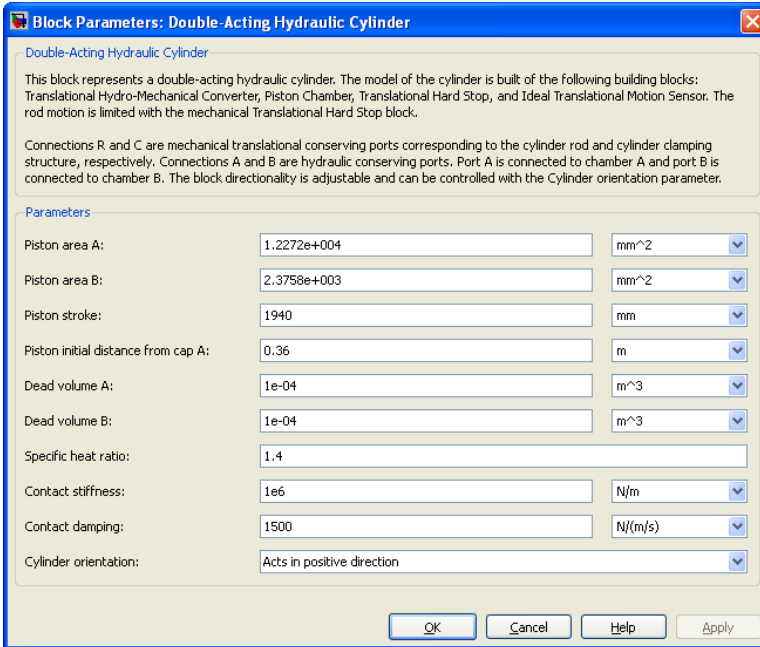
c)



Kuva B1 a) simulointiympäristön arkkitehtuuri b) sylinteriohjatun nivelen MBS-malli ENVISION TR:ssä c) nivelen hydrauliikkamalli MATLAB/Simulinkissä.

Ennen simulointia sylinterimallin halkaisijatiedot ja iskunpituus asetettiin mitoituksien mukaiseksi. Mallin alustamisessa on tärkeää ottaa huomioon se, että

Simhydraulicsin valmiissa sylinterimallissa ei ole mahdollisuutta asettaa alkuarvoa kammioiden paineille (kuva B2). Koska sylinterin kammioissa vaikuttavien paineiden alkuarvot ovat nolliä, myös sylinterin tuottama voima simulaation alussa on nolla.

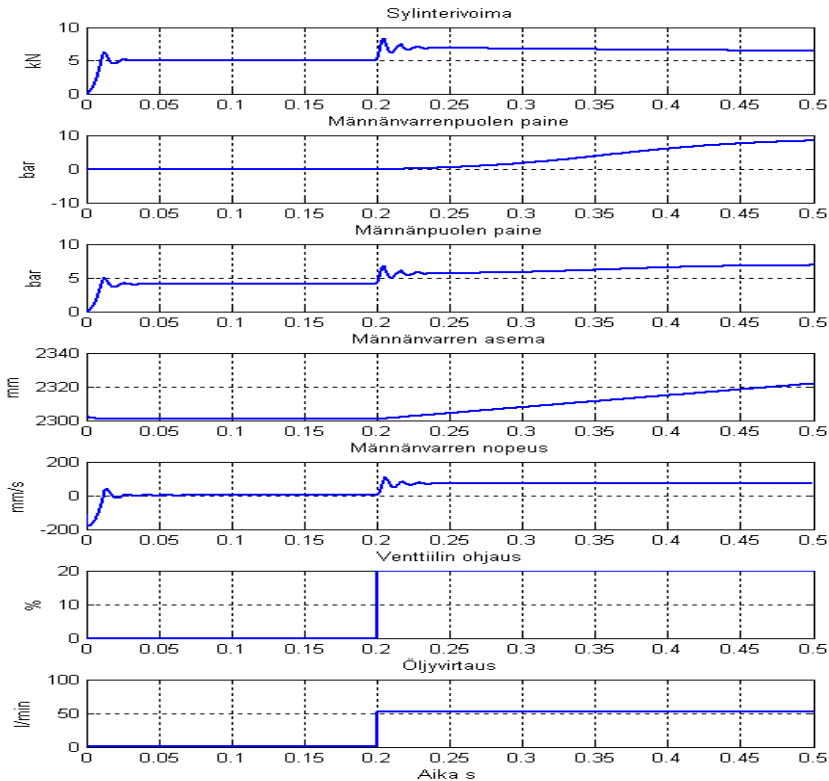


Kuva B2. Kaksitoimisen hydraulisynterimallin parametrisointi-ikkuna (Matlab Simhydraulics Version 1.6).

Kuvassa B3 ovat simulointitulokset alapuomin nostonivelen askelvastekokeesta. Hydraulisynterissä vaikuttavien paineiden nolla-alkuarvoista johtuen mekanismimalli ei ole simuloinnin alussa staattisessa tasapainossa. Suuntaventtiilin ollessa kiinni alapuomi lähtee maan vetovoiman vaikutuksesta liikkeelle levosta ja pysähtyy, kunnes männänpuoleisen kammion paine kasvaa ja sylinterivoima riittää pitämään nivelen tasapainossa.

Ajanhetkellä $t = 0,2$ s venttiilin ohjauksessa tapahtuu 20 %:n askelmainen muutos (venttiilin avautuma). Tällöin sylinterivoima kasvaa ja sylinterin varsi lähtee liikkeelle positiiviseen suuntaan (sylinterin pituus kasvaa). Sylinterivarren liike (kiihtyvyys ja nopeus) lasketaan ENVISION-ohjelman dynaamisesta mallista, jossa nivelen kulmakiihtyvyyden määräävät nivelen hitausmomentti ja kokonaisnivelmomentti.

Liite B: Matlab Simhydraulics -mallinnusesimerkki



Kuva B3. Simulointitulokset, alapuomin nostosylinterin askelvasteelle. Simuloinnin alkessa suuntaventtiili on kiinni (Ohjaus = 0) ja ajanhetkellä $t = 0,2$ s venttiilin ohjaussignaali tapahtuu 20 %:n askelmainen muutos.

Yhteenveto

MATLAB Simhydraulics -mallin linkitys ENVISIONiin oli haastava tehtävä, ja pääasiallisena syynä oli riittämätön dokumentaatio. MATLAB Simhydraulics -mallin linkitys toisessa sovellusohjelmassa suoritettavaan mekanismimalliin oli siinä määrin uutta, ettei siihen löytynyt ohjeita MATLABin dokumentaatiosta tai MATLAB Centralista. MATLABin oletushan on, että Simhydraulicsia käytetään yhdessä SimMechanics-lisäosan kanssa. Ohjelmien linkityksessä hyödynnettiin Instrument Control Toolboxia, jonka avulla MATLABia voitiin käyttää asiakas-koneena.

ENVISION TR:ään linkitetyn hydraulikkamallin simulointinopeus ei ollut lähellekään reaaliaikaista, joten Simhydraulicsin yhteiskäyttö ulkoisen MBS-

ohjelman kanssa vaikuttaisi soveltuvan huonosti tuotekehityksen alkuvaiheen luonnosteluun tai tuotteen toiminnallisuuden visualisointiin. Suositeltavampi käyttökohde olisi esimerkiksi nivelservojen yksityiskohtaisempi analyysi ja säätösuunnittelu.

Hydrauliikkamallin simulointinopeutta voi yrittää parantaa valitsemalla yksinkertaisempi integrointialgoritmi tai kasvattamalla integroinnin askelväliä. Tällöin on kuitenkin otettava huomioon, että pidempi askelväli lisää simulointivirhettä ja saattaa tehdä mallista epästabiilin. Reaaliaikaista simulointia tavoiteltaessa eräs mahdollisuus voisi olla myös Simulink Coder, jolla Simulink-mallista voidaan generoida C-kielinen versio.

MATLAB Simhydraulics näyttäisi tämän kokeilun perusteella soveltuvan moniteknisen tuotteen hydrauliikan mallinnukseen ja simulointiin. Selvää kuitenkin on, että mallin parametrisoinnissa on oltava erittäin huolellinen, jotta malli olisi mahdollisimman todenmukainen. C-ohjelmointiin verrattuna graafinen ohjelmointi helpottaa erityisesti ohjelmakoodin luettavuutta ja ylläpitoa. Simhydraulicsin etuja ovat myös laajahko komponenttikirjasto sekä hyvä dokumentaatio, jotka helpottavat mallien parametrisointia. Lisäksi ilmaisia neuvoja ohjelmointiin voi kysyä muilta MATLAB-ohjelmoijilta Mathworksin kotisivuilta löytyvästä MATLAB Centralista (www.mathworks.com/matlabcentral). MATLAB Centralissa on myös jatkuvasti päivittyvä ohjelmistojakelu, josta voi ladata ilmaisia Simulink-malleja ja MATLAB-ohjelmia.

Lähde

Tchkalov, V. & Miller, S. 2010 Parametrization of directional and proportional valves in Simhydraulics. Mathworks Inc.
<http://www.mathworks.com/matlabcentral/fileexchange/27260> (2.8.2011).



Tekijä(t) Jari M. Ahola, Jani Hovila, Eero Karhunen, Kalervo Nevala, Timo Schäfer & Tom Nevala		
Nimeke Moniteknisen piensarjatuotteen digitaalinen tuoteprosessi		
Tiivistelmä Viime vuosikymmenten nopea teknologinen kehitys on tarjonnut mahdollisuuden ajatella tuotekehitystä ja tuotteistamista uudella tavalla, jonka mukaan tuoteprosessi kattaa tuotteen koko elinkaaren tarvekartoituksesta uudelleen kierrätykseen. ”Moniteknisen tuotteen digitaalisen tuoteprosessin kehittäminen tehokkaaseen piensarjatuotteiden suunnitteluun ja tuotantoon” (MoniDigi) -tutkimusprojektin tavoitteena on kehittää moderneilla virtuaalisuunnittelun työkaluilla varustettu moniteknisten tuotteiden kehitysympäristö ja digitaalisen tuoteprosessin konsepti, joka on hyödynnettävissä laajasti pienissä ja keskisuurissa (pk)-yrityksissä. Pk-yritysten nykytilan kartoituksessa selvisi, että suunnittelussa käytetään usein moderneja ohjelmistoja ja digitaalista tuotetietoa hyödynnetään osittain suoraan valmistuksessa. Tuotetiedon ja tuotteen elinkaaren hallinnan järjestelmien (PLM-järjestelmät) hyödyntäminen on kuitenkin vielä vähäistä. Yritykset ovat kiinnostuneita digitaalisesta tuoteprosessista, mutta niillä ei ole siitä riittävästi tietoa eivätkä ne pysty panostamaan tarvittaviin selvityksiin. Pk-yrityksille suunnatussa digitaalisen tuoteprosessin konseptissa tuotteen elinkaari jaettiin neljään päävaiheeseen: tuotekehitys, valmistus, jälkimarkkinointi ja käytöstä poisto. Konseptin perusta on yrityksiä yhdistävä tietoverkko ja PLM-järjestelmä, jolla hallitaan niin tuotekehitysprosesseja, tuotetietoa kuin alihankintaakin. Tuotekehitysprosessia pilotoitiin tuotteilla, joiden konseptin kehittämisessä hyödynnettiin virtuaalisuunnittelua. Konseptin käyttöä pk-yrityksissä saattavat rajata virtuaalisuunnittelun sekä tuotetiedon hallintajärjestelmän käyttöönoton ongelmat.		
ISBN 978-951-38-7773-6 (nid.) 978-951-38-7774-3 (URL: http://www.vtt.fi/publications/index.jsp)		
Avainnimeke ja ISSN VTT Tiedotteita – Research Notes 1235-0605 (nid.) 1455-0865 (URL: http://www.vtt.fi/publications/index.jsp)		Projektinnumero 33899
Julkaisuaika Elokuu 2011	Kieli Suomi, engl. tiiv.	Sivuja 121 s. + liitt. 37 s.
Projektin nimi MoniDigi		Toimeksiantaja(t) TEKES
Avainsanat Product life-cycle management, product development, concurrent design, virtual design, simulation		Julkaisija VTT PL 1000, 02044 VTT Puh. 020 722 4404 Faksi 020 722 4374



Series title, number and
report code of publication

VTT Research Notes 2592
VTT-TIED-2592

Author(s) Jari M. Ahola, Jani Hovila, Eero Karhunen, Kalervo Nevala, Timo Schäfer & Tom Nevala		
Title Digital product process of a multitechnological small series product		
Abstract <p>The fast development of technology during past decades has enabled to think product development and productization in a new way in which the product process covers the whole product life-cycle from marketing research to recycling. The aim of the MoniDigi-project was to develop a modern virtual design environment for multitechnological products and a concept for digital product process (DPP) which can be utilized widely in small and medium sized enterprises (SMEs).</p> <p>The interview study about the present state of SMEs revealed that modern design softwares are used commonly and digital product data is exploited in manufacturing partially. However, the product data management and product life-cycle management (PLM) systems are used very little and e-mail is the most common way to transfer data which sets challenge for version management and data security. According to the surveys the SMEs are interested in DPP but they do not have sufficiently information and they cannot invest on required investigations.</p> <p>In the developed DPP concept the product life-cycle was divided in four main phases: product development, manufacturing, after-sales and dismantling. The concept is based on a data network connecting companies and a PLM system which is used to manage product development processes, product data and subcontracting. The product development process was verified with real product cases which were developed using virtual design tools. The commissioning of virtual design and product data management systems may delimit implementation of the DPP concept in SMEs.</p>		
ISBN 978-951-38-7773-6 (soft back ed.) 978-951-38-7774-3 (URL: http://www.vtt.fi/publications/index.jsp)		
Series title and ISSN VTT Tiedotteita – Research Notes 1235-0605 (soft back ed.) 1455-0865 (URL: http://www.vtt.fi/publications/index.jsp)		Project number 33899
Date August 2011	Language Finnish, Engl. abstr.	Pages 121 p. + app. 37 p.
Name of project MoniDigi		Commissioned by TEKES
Keywords Product life-cycle management, product development, concurrent design, virtual design, simulation		Publisher VTT Technical Research Centre of Finland P.O. Box 1000, FI-02044 VTT, Finland Phone internat. +358 20 722 4404 Fax +358 20 722 4374

MoniDigi-tutkimusprojektin tavoitteena oli kehittää moderneilla virtuaalisuunnittelun työkaluilla varustettu moniteknisten tuotteiden kehitysympäristö ja digitaalisen tuoteprosessin konsepti, joka on hyödynnettävissä laajasti pienissä ja keskiuurissa pk-yrityksissä. Pk-yrityksille suunnatussa digitaalisen tuoteprosessin konseptissa tuotteen elinkaari jaettiin neljään päävaiheeseen: tuotekehitys, valmistus, jälkimarkkinointi ja käytöstä poisto. Konseptin perusta on yrityksiä yhdistävä tietoverkko ja PLM-järjestelmä, jolla hallitaan niin tuotekehitysprosesseja, tuotetietoa kuin ali-hankintaakin. Tuotekehitysprosessin alkuvaihe todennettiin tuotepilottien avulla, joille toteutettiin tuotekonseptin kehittämisvaihe hyödyntäen virtuaalisuunnittelua.