

Elektroniikkatuotannon joustavan ohjauksen tietotekninen infrastruktuuri

Eila Niemelä
VTT Elektroniikka



ISBN 951-38-5454-X
ISSN 1235-0605
ISBN 951-38-5455-8
ISSN 1455-0865
Copyright © Valtion teknillinen tutkimuskeskus (VTT) 1999

JULKAISIJA – UTGIVARE – PUBLISHER

Valtion teknillinen tutkimuskeskus (VTT), Vuorimiehentie 5, PL 2000, 02044 VTT
puh. vaihde (09) 4561, faksi (09) 456 4374

Statens tekniska forskningscentral (VTT), Bergsmansvägen 5, PB 2000, 02044 VTT
tel. växel (09) 4561, fax (09) 456 4374

Technical Research Centre of Finland (VTT), Vuorimiehentie 5, P.O.Box 2000, FIN-02044 VTT, Finland
phone internat. + 358 9 4561, fax + 358 9 456 4374

VTT Elektroniikka, Sulautetut ohjelmistot, Kaitoväylä 1, PL 1100, 90571 OULU
puh. vaihde (08) 551 2111, faksi (08) 551 2320

VTT Elektronik, Inbyggd programvara, Kaitoväylä 1, PB 1100, 90571 ULEÅBORG
tel. växel (08) 551 2111, fax (08) 551 2320

VTT Electronics, Embedded Software, Kaitoväylä 1, P.O.Box 1100, FIN-90571 OULU, Finland
phone internat. + 358 8 551 2111, fax + 358 8 551 2320

Toimitus Kerttu Tirronen

Libella Painopalvelu Oy, Espoo 1999

Niemelä, Eila. Elektroniikkatuotannon joustavan ohjauksen tietotekninen infrastruktuuri. Espoo 1999, Valtion teknillinen tutkimuskeskus, VTT Tiedotteita – Meddelanden – Research Notes 1965. 42 s.

Avainsanat production control, electronics industry, computer integrated manufacture (CIM), software, flexibility

TIIVISTELMÄ

Joustavan tuotannonohjauksen tärkeimpiä ominaisuuksia ovat järjestelmän avoimuus, standardiperusteisuus ja käyttäjäystävällisyys. Joustavat ohjelmistoarkkitehtuurit ja hajautusalustat tukevat tuotannonohjauksen tietoteknisen infrastruktuurin toteuttamista. Standardiliityntöjä noudattavat osajärjestelmät ja ohjelmistokomponentit mahdollistavat tuotannonohjauksen inkrementaalisen kehittämisen, mutta standardien kirjavuus aiheuttaa ongelmia integroitujen järjestelmien kehittäjille. Käytännössä järjestelmät eivät ole yhteensopivia, ja integrointi vaatii huomattavan kehittämisen jälkeisen työpanoksen. Ongelmat ovat osittain seurausta kehittämisajaksista päätöksistä, joiden virheellisyyttä edesauttaa järjestelmän kehittäjän, tilaajan ja käyttäjien näkökulmia tukevan suunnittelumenetelmän ja kuvaustavan puuttuminen.

Yritystason tuotannonohjaukseen soveltuva kaupallinen SAP/R3 perustuu kolmitasoiseen asiakas-palvelinarkkitehtuuriin. Asiakasohjelmat ovat graafisia käyttöliittymiä, joiden avulla käyttäjä aktivoi keskitason sovelluslogiikan. Perinteiseen relaatiotietokantaan perustuva palvelin muodostaa kolmannen tason. Arkkitehtuuri on modulaarinen, ja lähestymistapana on informaatiojärjestelmille ominainen tietojen ja tietovirtojen hallinta. Järjestelmää voidaan soveltaa sekä pienten että suurten yritysten tuotannonohjaukseen. Hierarkkisuutensa ansiosta tuotannonohjauksen jäljitettävyyden on hyvin tuettu, mutta järjestelmän joustavuus ja integroitavuus vaativat kehittämistä.

CIM-kehysrakenne tarjoaa tuotannon ohjaukselle CORBA-yhteensopiviin tuotteisiin perustuvan alustan ja spesifikaatiot alustaan sopivien komponenttien kehittämiseen. Kehysrakennetta voidaan hyödyntää myös askelittain kehittämällä toimiviin ratkaisuihin kehysrakenteen mukaiset liittynät, jotka voidaan liittää kaupallisiin CORBA-tuotteisiin. Ohjelmistoarkkitehtuurin uudelleensuunnittelu on useimmissa tapauksissa välttämätöntä. Kehysrakenteen hyödyntämiseen tarvitaan asiantuntemusta oliotekniikasta ja CORBA-arkkitehtuurista

Kehysrakenteen käyttö edellyttää useiden CORBA-palveluiden olemassaolon. Tämä tarkoittaa kyseisten palvelujen ostamista, tilaamista alihankkijalta tai kehittämistä. Käyttöönotto-, testaus- ja ylläpitokustannusten alenemisesta saatava hyöty puoltaa kehysrakenteen kehittämistä ja käyttöönottoa. Kehysrakenne yhtenäistää eri tukiohjelmistojen toteuttamisperiaatteet ja tarjoaa yhtenäisen toimintamallin ohjaus- ja informaatiojärjestelmien kehittämiselle.

Sisällysluettelo

TIIVISTELMÄ	3
LYHENTEET	5
1. Johdanto	7
2. TUOTANNONOHJAUKSEN TIETOTEKNINEN INFRASTRUKTUURI	8
2.1 Hajautetun tuotannonohjauksen infrastruktuuriin liittyviä standardeja	8
2.2 Tuotannon tukijärjestelmien kaupallisia sovelluksia	10
2.3 Tuotannonohjausjärjestelmien kehittämiseen liittyviä tutkimushankkeita	11
2.4 Yritystason tuotannonohjaus - SAP R/3	13
2.5 Tuotannon suorituksen ohjaus - CIM-kehysrakente	15
3. TUOTANNONOHJAUKSEN JOUSTAVUUSTARPEET	21
3.1 Ohjausohjelmiston kehitys	21
3.2 Ohjausjärjestelmän käyttöönotto	22
3.3 Järjestelmän integrointi	23
3.4 Järjestelmän käyttäminen	23
3.5 Käytetyt ratkaisumallit	24
3.6 Kehittämiskohteet	24
4. CIM-KEHYSRAKENTEEN TARJOAMA TUKI OHJELMISTON JOUSTAVUDELLE	26
4.1 Jäljitettävyys ja optimointi	26
4.2 Yleiset rajapinnat ja palvelut	29
4.3 Toiminnalliset ryhmät ja komponentit	30
4.3.1 Tehtaanhallinta	31
4.3.2 Materiaalinhallinta	32
4.3.3 Prosessikuvausten hallinta	35
4.3.4 Materiaalinsiirto	36
4.3.5 Koneenohjaus	36
4.3.6 Laajennettu prosessinohjaus	37
5. Yhteenveto	40
LÄHTEET	42

LYHENTEET

CIM	Computer Integrated Manufacturing
CORBA [®]	Common Object Request Broker Architecture
DCE	Distributed Computing Environment
DCOM	Distributed Component Object Model
DRP	Distribution Resource Planning
EPRI	Electric Power Research Institute
ERP	Enterprise Resource Planning
F/S	Focused solutions
IDL	Interface Description Language
I-MES	Integrated MES: MES + F/S systems
ISA	Instrument Society of America. ANSI:n (American National Standards Institute) valtuuttama standardien kehittäjä v. 1976 lähtien. http://www.isa.org/ .
MESA	Manufacturing Execution Systems Association, http://www.mesa.org/
MRPII	Manufacturing Resource Planning
OMG	Object Management Group
ORB	Object Request Broker
RPC	Remote Procedure Call

[®] CORBA on OMG:n rekisteröimä tuotemerkki

RTDB Real-Time DataBase

SAP Systems, Applications and Products in Data Processing

SAP R/3. Integroitu, modulaarinen ohjelmistorakenne, joka tukee laajennettavuutta ja skaalautuvuutta.

SEMI Semiconductor Materials and Equipment International

Supply Chain Mgmt. Forecasting + DRP

UML Unified Modeling Language

1. JOHDANTO

Tämä raportti liittyy CASPER-projektin, vuosina 1997 ja 1998 toteutetun Tekes tavoitetutkimusprojektin, osatehtävään kolme. Raportissa tarkastellaan elektroniikkatuotannon ohjausjärjestelmien joustavuustarpeita ja ohjelmistoteknisiä keinoja näiden tarpeiden tyydyttämiseksi.

Tuotannonohjausjärjestelmien kehittämiseen on tarjolla ulkomaisten toimittajien järjestelmiä, jotka soveltuvat joko taloushallintoon, tuotannonohjaukseen tai materiaalinhallintaan. Kokonaisjärjestelmät, jotka tukisivat kaikkia osa-alueita, rakennetaan integroimalla järjestelmiä toisiinsa ja kehittämällä niiden välille tarvittava tukiteknologia. Osajärjestelmien kehittäjät käyttävät erilaisia teknologioita, joiden yhteensovittaminen tuottaa integroitujen järjestelmien kehittäjille suuria ongelmia. Siksi tässä raportissa tarkastellaan, kuinka jo kehitetyt ratkaisumallit soveltuvat ja mukautuvat integroidun järjestelmän toteuttamiseen. Ratkaisumallien soveltuvuutta tarkastellaan myös tuotantolinjan reaaliaikaisen seurannan ja ohjauksen kannalta, lisäksi selvitetään miten ratkaisumalleissa on otettu huomioon mallien kyky sopeutua mahdollisimman vähin muutoksin tuotantolinjassa ja sen toiminnassa tapahtuviin muutoksiin. Ratkaisumallien tarkastelu perustuu lähinnä kirjallisuuteen, jota on ollut saatavilla SAP/R3:sta ja CIM-kehysrakenteesta.

Raportin luvussa 2 kuvataan tuotantolinjan tietoteknistä infrastruktuuria koskevia standardeja ja niiden kehitykseen liittyvää tutkimusta. Siinä kuvataan myös tuotantoresurssien ohjausta tukeva SAP/R3 -järjestelmä sekä CIM-kehysrakenne. CIM-kehysrakenne tukee tuotannon kokonaisvaltaista ohjausta resurssien suunnittelusta yksittäisen koneen tai koneryhmän ohjaukseen. CIM-kehysrakenteen merkitys tuotannonohjausjärjestelmien kehittämisessä on tulevaisuudessa suuri, koska spesifikaation standardoimistyön tavoitteena on liittää se osaksi OMG:n CORBA -standardia sovellusaluekohtaisten palvelujen spesifikaationa.

Luvussa 3 esitetään kootusti kartoituksessa esille tulleet elektroniikkateollisuuden tarpeet tuotannonohjauksen joustavuuden lisäämiseksi.

Luvussa 4 kuvataan, miten CIM-kehysrakenne tukee tuotannonohjauksen joustavuustarpeita ja miten tuotannonohjauksen optimointi tulisi huomioida järjestelmän arkkitehtuuria suunniteltaessa.

Luvussa 5 esitetään lyhyt, arvioiva yhteenveto tilanteesta.

2. TUOTANNONOHJAUKSEN TIETOTEKNINEN INFRASTRUKTUURI

Tässä luvussa käsitellään yleisesti elektroniikkatuotannon infrastruktuuria koskevia standardeja ja tukijärjestelmiä sekä niiden kehittämiseen tähtäviä meneillään olevia kansainvälisiä tutkimusprojekteja.

Tuotannon tietotekniseen infrastruktuuriin luetaan seuraavat asiat:

- arkkitehtuuri,
- komponentointi,
- hajautusalustat sekä
- komponenttien ja osajärjestelmien liityntätavat.

Valmiiksi kehitetyistä tukijärjestelmistä tarkastellaan SAP R/3:a ja sen tarjoamaa tukeaa joustavan tietoteknisen infrastruktuurin toteuttamiseksi. SAP R/3 on yritystason resursien suunnitteluun ja seurantaan tarkoitettu järjestelmä, jossa liiketoimintaintressit yhdistyvät varsinaiseen tuotannonohjaukseen.

Varsinaisista tuotannonohjausjärjestelmistä käsitellään SEMATECHin kehittämää CORBA (Common Object Request Broker Architecture) -arkkitehtuuriin perustuvaa kehysrakennetta CIM-sovellusten toteuttamiseksi. CORBA arkkitehtuuri on client-server-arkkitehtuuriin perustuva hajautettujen olioiden kehysjärjestelmä. Sen ytimenä toimivat arkkitehtuuritason liityntäkuvaukset (IDL), joita hyödynnetään ohjelmistoarkkitehtuurin suunnittelussa ja varsinaisessa toteutuksessa, sekä ohjelmistoväylä (ORB), jonka kautta hajautetut oliot kommunikoivat. Liityntäkielen tehtävänä on taata olioiden riippumattomuus fyysisestä paikasta, suoritusalueen tyypistä, käytetystä protokollasta ja kielestä eli tarjota puitteet ohjelmiston maksimaaliseen uudelleenkäyttöön ja joustavien ohjelmistojen toteuttamiseen.

2.1 Hajautetun tuotannonohjauksen infrastruktuuriin liittyviä standardeja

Standardien käytöllä pyritään helpottamaan järjestelmien integroitavuutta. Vaihtoehtoisia standardeja on useita. Standardin valinta vaikuttaa suuresti siihen, miten joustavasti alijärjestelmät voidaan kytkeä toisiinsa. Tuotannonohjausjärjestelmän kehittäjän täytyy sovittaa olemassa olevat ohjelmistot ja järjestelmät toisiinsa erillisten sovittimien avulla, koska tulevaisuudessakaan ei ole näkyvissä helpotusta standardien kirjavuuteen.

Seuraavassa on esitetty muutamia erilaisissa ohjausjärjestelmissä käytettäviä standardeja.

- CORBA OMG:n spesifikaatio läpinäkyvän olioväylän tarjoamiseksi hajautetuille järjestelmille. Ensimmäinen spesifikaatio ilmestyi 1991, nykyinen versio on 2.1. Tekeillä on POA (Portable Object Adapter), joka tehostaa CORBA-palvelinsovellusten siirrettävyyttä.
- DCOM Oliopohjainen hajautusalusta, jossa ORPC (Object Remote Procedure Call) -taso on toteutettu DCE RPC:n päälle. DCOM on Microsoftin vastine CORBA:lle.
- FDDI Fiber Distributed Data Interface, ANSI:n standardi, joka määrittelee fyysisen tason protokollan ja aseman konfiguroinnin, kun mediana käytetään optista kuitua (ISO:lla vastaava olemassa).
<http://www.alef0.cz/dokumentace/doccd2/data/doc/cintrnet/ito/55773.html>.
- FMS Fieldbus Message Specification per DIN 19 245. German National Standard. Käsittää alijoukon MMS -palvelujen ja -protokollan määrittelyistä.
- MMS Manufacturing Message Specification (ISO/IEC-9506). Kansainvälinen standardi, joka määrittelee sovellustason protokollan ja sisältyy UCA v. 1.0 ja 2.0 -spesifikaatioihin. MMS:n tavoitteena on tukea liiketoimintafunktioita kätkemällä sovellustason toiminnot MMS-palvelujen alle. MSS tarjoaa Virtual Manufacturing Device -mallin korkean tason yhteistoiminnalle laitteiden ja sovellusten välillä. [1],[2],[3].
- SMEMA Surface Mount Equipment Manufacturers Association. Kehitetty SRRFF (Standard Recipe File Format) on prosessinohjaustiedostojen luontia varten. <http://www.smema.com>.
- UCA Utility Communications Architecture. Perustuu ISO:n standardeihin. ICCP (Inter Control Centre Communications Protocol) määrittelee, kuinka MMS:ää käytetään UCA:n mukaisissa verkoissa valvomoiden, voimaloiden ja SCADA-asemien välillä. Pohjautuu TCP/IP-protokollaan (multicast/broadcast, connectionless/connection-oriented) tarjoten yleisen liittymän sähkö-, kaasu- ja vesivoiman ohjausjärjestelmille. [4]. EPRI (Electric Power Research Institute) on esittänyt luonnoksen uudesta UCA:sta [5].

2.2 Tuotannon tukijärjestelmien kaupallisia sovelluksia

Kaupallisten sovellusten tarjonta ja ominaisuudet vaikuttavat myös tuotannonohjausjärjestelmien kehittämiseen. Siksi seuraavassa on lueteltu muutamia tuotannonohjaukseen liittyviä kaupallisia tukijärjestelmiä käyttötarkoituksen mukaisesti luokiteltuina:

CAD/CAM/CAE Computer-Aided Design and Computer-Aided Manufacturing Systems

CADDS Computervision Corp. Tuotannonohjauksen suunnittelu ja hallinta. Yhteistyöympäristö EPD:lle. <http://www.cv.com>.

EUCLID
Quantum Matra Datavision Inc. CAD/CAM/CAE ja PDM -järjestelmä, joka pohjautuu oliotekniikkaan. Ominaisuuksiin perustuva valmistus- ja työvirtojen suunnittelun tuki. <http://www.matra-datavision.com>.

CMMS Computerized Maintenance Management Systems

MP2 Datastream Systems Inc. Integroitu ylläpidon tukijärjestelmä, joka perustuu tilastollisesti ennustettavaan kunnossapitoon (Statistical Predictive Maintenance).

MES Manufacturing Execution Systems

BASEstar Digital Equipment Corp. Open API, RTDB, RT event management, 4GL. Tukee tapahtumapohjaisia MES-sovelluksia ja transaktio-orientautuneita ERP-sovelluksia. Kolme sovellusta, esim. Process Data Acquisition System, joka täydentää SAP R/3 PP-PI moduulia.

TCM Time Critical Manufacturing. Effective Managment Systems. Yhdistetty tukiohjelmisto ERP- and MES- järjestelmille, sisältää PDM:n, MES:n, muutostenhallinnan, suunnittelun, rahoituksen, kirjanpidon ja hallinnon sekä päätöksentuen ja yritysjohdon työkalut. <http://www.ems.com>.

Work-Stream
DFS Consilium Inc. Hajautetun tuotannon työvirtojen ohjaus. Avoin MES-tuote: lattiatason Station Controller, reseptien hallintapalvelin, laatupalvelin (Quality Server). <http://www.consolium.com>.

MMI	Man-Machine Interface
InTouch	Wonderware Corp. Olio-orientoitunut graafinen MMI-sovellusgeneraattori automaatiojärjestelmien ja prosessien ohjaukseen sekä työnjohdon valvontaan. http://www.wonderware.com .
MRPII&ERP	Manufacturing Resource Planning / Enterprise Resource Planning
SAP R/3	SAP Systems Applications Products. Release 4.0 R/3: Komponentit liiketoimintakehysrakenteen arkkitehtuurille. http://www.sap.com .
PDM	Document / Product Data Management / sen laajennuksia
EPD	Electronic Product Definition. Connect: käyttäjien yhdistäminen sovelukseen, Manufacturing Engineer: CADDS 5:n päälle toteutettu integroitu tuotannosuunnittelun ja -ohjauksen työkalu. Product and Process Management-myyntioptio. Computer Vision, http://www.cv.com .
CIM/21	Computer Integrated Manufacturing -työkalu, Industrial Systems Inc. (ISI). CIM/21 tarjoaa yritystoimintaa tukevat ohjelmistopalvelut AViiON ja CLARiiON käyttäjille, jotka ovat Data General:n tuotteita. ISI on keskittynyt prosessinohjaukseen ja sen informaatiojärjestelmien tekemiseen. http://www.meteksan.com.tr/www.dg.com/solutions/html/cim_21.html

2.3 Tuotannonohjausjärjestelmien kehittämiseen liittyviä tutkimushankkeita

Tuotannonohjausjärjestelmien kehittämisessä tapahtuvia muutoksia voidaan ennakoida tarkastelemalla kansainvälisien tutkimusprojektien sisältöä. Seuraavassa on lueteltu muutama tutkimushanke, jotka kaikki käsittelevät jotain hajautettujen ohjausjärjestelmien kehittämisessä esiintyvää ongelma-aluetta tai kehittämiskohdetta.

CIME	Computer-Integrated Manufacturing and Engineering. Kohteena arkkitehtuurit, kommunikointi, hallinta ja suunnittelu mekatroniikan, robotiikan ja aistinnan alueella. http://albion.ncl.ac.uk/
------	---

- CNMA Communications Network for Manufacturing Applications. Aikaisemmat vaiheet 1 - 3 ovat käsitelleet CNMA:n spesifiointia, toteutusta ja validointia. Nykyinen ESPRIT-projekti kuuluu vaiheeseen 4 ja tähtää MMS, Directory Service, FTAM-palvelujen liittämiseen tuotantolinjan ohjaussovellusten kommunikointiprotokolliin. <http://albion.ncl.ac.uk/esp-syn/text/2617.html>.
- EIMS Enterprice Integration of Manufacturing Systems. Bostonin yliopiston PCMS laboratorion tutkimushanke. Tasoajatteluun perustuva järjestelmäarkkitehtuuri, jonka ositus perustuu fyysisiin tai toiminnallisiin kriteereihin. Toiminnallisen tiimin yhteistoiminnasta huolehtii ADA (Autonomous Decission Agent)-komponentti. Arkkitehtuuri tukee erilaisia toimintamoodeja: järjestelmä- ja komponenttitason konfigurointia, suunnittelun simulointia ja asynkronista toimintaa. <http://cad.bu.edu/pcms/ei/about/about.html>.
- ExSM Expert Systems in Manufacturing. University of Sunderland. Esimerkkejä AI-tekniikan käytöstä tuotannonohjausjärjestelmissä perustuen kansainvälisiin tutkimusprojekteihin. <http://www.bkng.demon.co.uk/tech/papers/es.html>.
- IT-CIM Integration Testing for Computer-Integrated Manufacturing. ESPRIT-projekti, jossa tavoitteena on kehittää tuotantoympäristöjen integroinnissa tarvittavia testausmenetelmiä ja apuvälineitä. Hajautuksessa käytetään LAN- ja kenttäväyliä.
- OMI OMI Software Architecture Forum (OSAF). Luokittelee sulautetuissa järjestelmissä olevat liittynät viiteen kategoriaan: sovelluskomponenttien väliset, järjestelmäkomponenttien väliset, protokollapohjaiset verkkoliittynät, hw/sw-liittynät ja hw-komponenttien liittynät [6].
- Solutions Architecutre Massachusetts Institute of Technology, Laboratory of Computer Science. Information Systems Achitecture: 4-tasoinen arkkitehtuuri sisältäen liike-elämä-, järjestelmä-, teknologia- ja tuotearkkitehtuuri-tasot. Ajattelun lähtökohtana mukautuvuus, laajennettavuus, skaalautuvuus, siirrettävyys, hajautettavuus ja jaettavuus rakenteellisesti stabiilissa arkkitehtuuriratkaisussa. <http://www.lcs.mit.edu/alex/arch/arch97/arch71.html>.

2.4 Yritystason tuotannonohjaus - SAP R/3

SAP R/3 ja R/2 ovat SAP AG, Systems, Applications and Products in Data Processing -yrityksen (Saksa) tuotemerkkejä. SAP R/3 perustuu 3-tasoiseen asiakas-palvelinarkkitehtuuriin, jossa ensimmäinen taso muodostuu GUI-asiakassovelluksista, toinen taso on sovellustaso ja kolmas taso muodostuu tietokantapalvelimesta. Lähestymistapa on puhtaasti informaatiojärjestelmille (Management Information Systems) ominainen ja soveltuu kaupallishallinnon sovelluksiin. Mallissa ei käsitellä reaaliaikaisuutta, mutta se sisältää on-line- tapahtuvat asynkroniset ja synkroniset toiminnot, esim. päivitykset tietokantaan. Tietokannan oletetaan olevan relaatiopohjainen, vaikka oliotekniikkaa on hyödynnetty järjestelmän määrittelyssä, esim. organisatoristen elementtien määrittelyissä, sovelluselaimissa ja käyttöliittymien ikkunoinnissa.

Tietokanta sisältää pysyväisluontoista ja muuttuvaa tietoa. Lähes muuttumattomia tietoja ovat sovellusten konfigurointi- ja kontrollitiedot sekä ns. master-tiedot, jotka sisältävät myyntiin, ostoon ja tuotantoon liittyviä pysyviä tietoja. Toimintojen tuottamat tiedot ovat muuttuvia ns. transaktiotietoja. CTS (Correction and Transport System) huolehtii järjestelmän muutosten hallinnasta, esim. tiedon konvertoinnista eri järjestelmien välillä tai päivityksen yhteydessä tuotekehityksen ja tuotantoversioiden välillä. ABAP/4 (Advanced Business Application Programming/4) on SAP:n 4GL -kieli dialogien ja tietokantanäkymien luontiin ja se mielletään osaksi R/3:n ohjelmistoalustaa (middleware), joka linkittää toisiinsa asiakas- ja palvelinsovellukset.

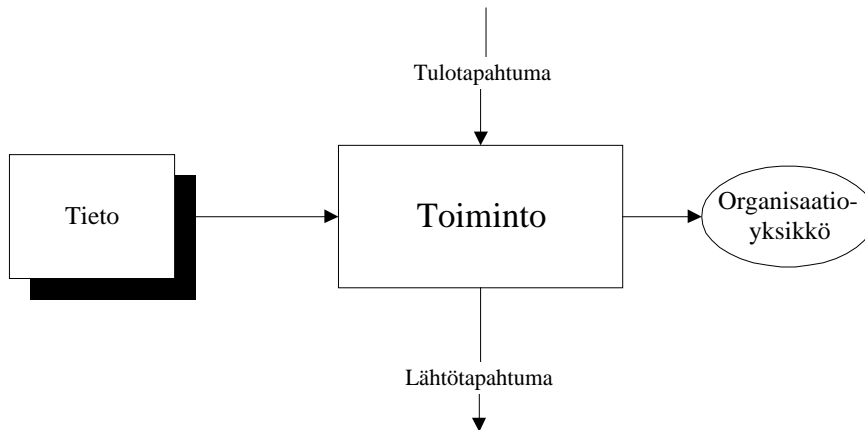
SAP:n sisältämät moduulit jaetaan seuraaviin neljään elementtiin:

- liiketoiminta,
- henkilöresurssit,
- tuotanto ja logistiikka sekä
- myynti ja jakelu.

Järjestelmien välistä integroitavuutta tuetaan ALE (Application Link Enabling) -liityntätekniikalla, joka toimii SAP:n hajautusalustana. Liiketoiminta on jaettu kolmeen pääkategoriaan: rahoitus, ohjaus (kontrolli) ja tukipalvelut. Henkilöresurssien hallinta sisältää tarvittavat palvelut esim. aikataulutukset ja palkanmaksun. Tuotanto ja logistiikka muodostavat järjestelmän monimutkaisimman osan sisältäen materiaalinhallinnan, tehtaan ylläpidon, tuotannon suunnittelun ja valvonnan, laadunvalvonnan sekä projektihallinnan. Myynnin ja jakelun tukijärjestelmämoduulit voidaan toteuttaa globaaleina, mikä

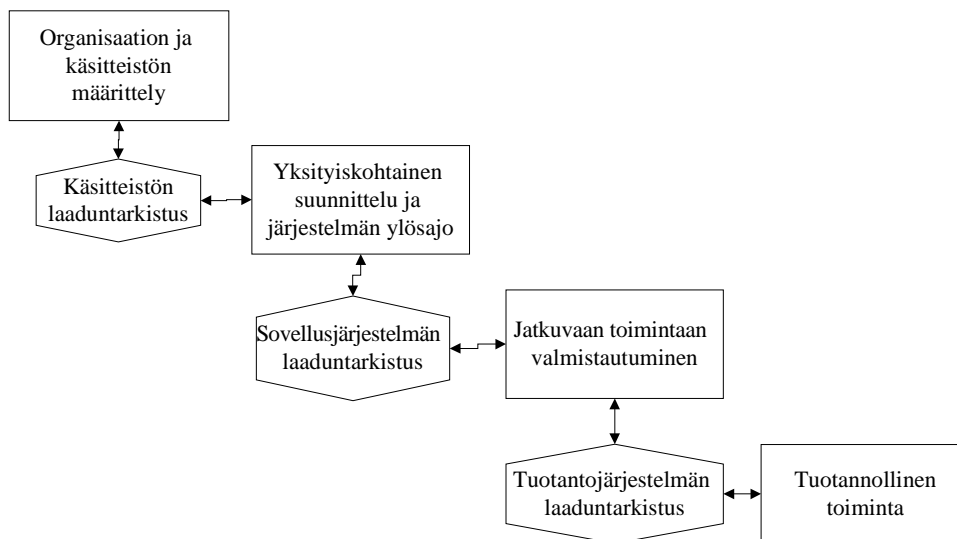
tukee asiakaspalvelun, tilaustenkäsittelyn, konfiguroinnin sekä jakelu-, vienti-, laivaus- ja kuljetuspalvelujen hallinnan yhteneväisyyttä.

SAP:n referenssimalli perustuu tapahtumajohdettuun prosessinmuutokseen, jossa määritellään toiminnot, toimintoon liittyvät tiedot ja organisatoriset yksiköt sekä toiminnon aktivoiva(t) tapahtuma(t) sekä toiminnosta edelleen generoitu/generoidut tapahtuma/tapahtumat (kuva 1).



Kuva 1. Tapahtumajohdetun prosessimuutoksen toimintaperiaate.

SAP määrittelee myös R/3:n kehittämiseen vesiputousmallia noudattavan nelivaiheisen suunnitteluprosessimallin, jossa kehitysprosessin laatua valvotaan vaiheiden välissä tapahtuvalla mittauksella (kuva 2). Prosessin tarkempi kuvaus on määritelty IMG:ssa (Implementation Management Guide).



Kuva 2. SAP:n kehittämisprosessin vaiheet ja tarkistuspisteet.

SAP sisältää yhtenä organisaation osa-alueena tuotannon ohjaukseen liittyvät asiat. SAP-järjestelmä on tarkoitettu suurien tehdaslaitosten informaatiojärjestelmäksi, vaikka sitä on viime aikoina pyritty soveltamaan myös pk-yritysten tarpeisiin NT-pohjaisena hajautettuna järjestelmänä. SAP-järjestelmä on pitkälle hierarkisoitu, ja järjestelmän joustavuus ja yhdistyvyys erilaisten järjestelmien kanssa on jätetty täysin huomiotta. Tämä on puute sovellettaessa SAP R/3:a joustavan elektroniikkatuotannon ohjaukseen.

Lisätietoa SAP R/3:sta voi hankkia Bancroftin kirjasta ja siinä mainituista viitteistä [7]. Kirja itse määrittelee käytetyt termit ja lyhenteet heikosti, mutta soveltuu SAP R/3:n perustietojen hankintaan.

2.5 Tuotannon suorituksen ohjaus - CIM-kehysrakenne

SEMATECH-konsortio, johon kuuluvat esimerkiksi Digital, Intel, Motorola ja National Semiconductor, on kehittänyt CIM -kehysrakenteen [8]. Kehysrakenne perustuu oliotekniikan ja CORBA-arkkitehtuurin hyödyntämiseen. Tässä raportissa ei käsitellä itse CORBAa, koska siitä löytyy erittäin paljon hyvää kirjallisuutta [11-13].

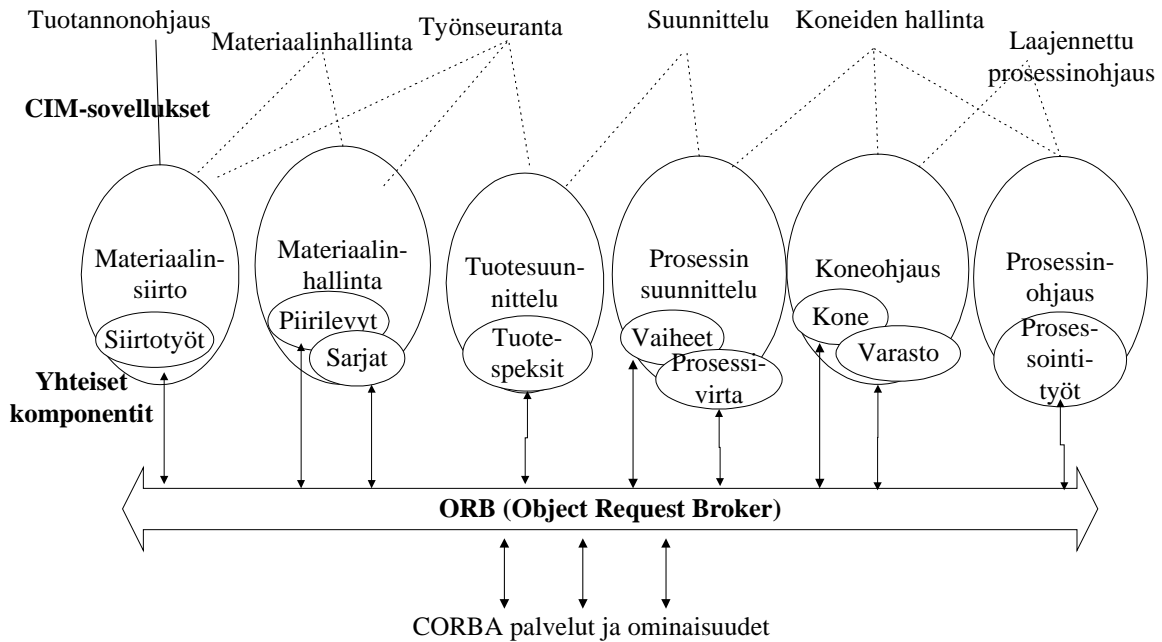
CIM-kehysrakenne käsittelee MES (Manufacturing Execution System) -tason tuotannonohjausta ja jättää tarkastelun ulkopuolelle yritystason tuotannonohjauksen, kuten esimerkiksi tilausten käsittelyn. Kehysrakenne ei myöskään käsittele itse työkalujen ohjausta, kuten käyttäjän ja koneen välistä vuorovaikutusta.

Sovellusoliot sisältävät kehysrakenteen määrittelemän liitynnän, fyysisen laitteen tai sovelluksen toimittajan määrittelemät ulkoiset ja sisäiset liitynnät sekä liityntöjen yhteensovittamisessa tarvittavat muunnokset, jotka ovat toimittajan vastuulla. CIM-kehysrakenteen käyttäjä eli soveltaja määrittelee käytettävän liityntätekniiikan, johon laitteiden tai alijärjestelmien toimittajien on sopeuduttava.

SEMI on aloittanut kehysrakenteen standardoimiseen tähtäävän työn vuoden 1997 alkupuoliskolla. Myös OMG Manufacturing Domain Task Force:n työn odotetaan kohdistuvan CIM-kehysrakenteen omaksumiseen.

Kehysrakenne perustuu CORBA-palveluiden käyttöön läpinäkyvän hajautusalustan toteuttamiseksi. Arkkitehtuuri muodostuu sovellusalueen yhteisistä komponenteista ja itse CIM-sovelluksista (kuva 3). Yhteiset komponentit luokitellaan erilaisiin ryhmiin toiminnallisen vastuualueensa perusteella. Sovellukset tai sovellusoliot hyödyntävät yhteisten komponenttien palveluja CIM-kehysrakenteen mukaisen liitynnän kautta, joka määritellään CORBAn IDL:n avulla. Reaalimaailman oliot ja niiden liitynnät upotetaan CIM-kehysrakenteen komponentteihin, joita CIM-järjestelmien toimittajat käyttävät

sovelluskomponenttien toteuttamisessa “rakennusmateriaalina”. Kehysrakenteen valmiit komponentit on ryhmitelty tuotepaketeiksi, jotka on suunniteltu joustaviksi laajennettavuuden ja optimoitavuuden suhteen.



Kuva 3. CIM-kehysrakenteen toiminnalliset ryhmät.

CIM-kehysrakenteen arkkitehtuuri perustuu resurssinhallintaohjelmien (resource manager) muodostamiseen ja integrointiin. Resurssinhallintaohjelmisto hallitsee resurssijoukon tilaa ja tarjoaa standardiliitännät, joiden kautta resursseihin voidaan kohdistaa toimintoja. Suora vaikuttaminen hallintaohjelman ohi ei ole sallittua. Hallintaoperaatiot sisältävät resurssien alustukset, poistamiset, jäljittämisen, konfiguroinnin ja suorituskyvyn hienosäädön. Resurssinhallinnalla on kolme eri tasoa [9]:

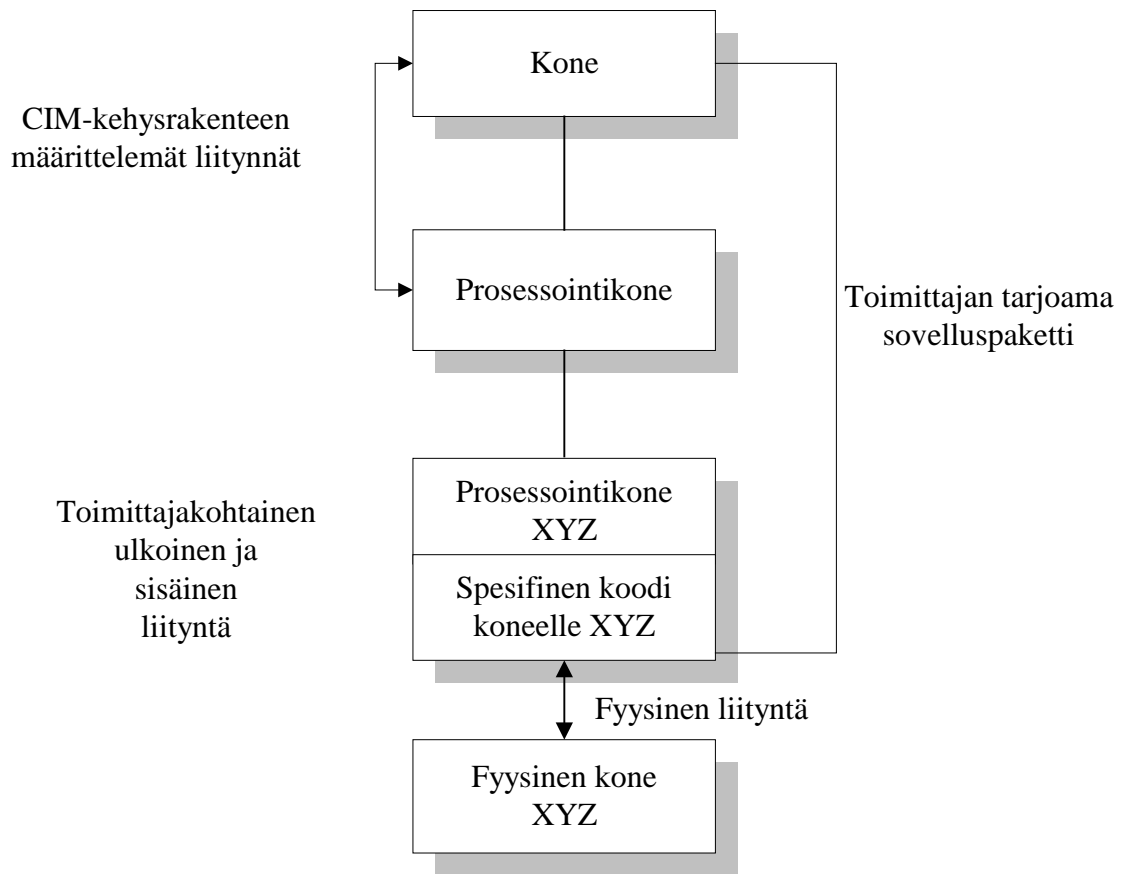
- liityntätaso, joka tarjoaa sovellusaluekohtaiset ohjelmointiliitännät (API)
- komponenttitaso, joka muodostuu joukosta samaan kohteeseen kuuluvia sovellusaliitintöitä (Domain Interface)
- pakettitaso, joka muodostuu joukosta yhteenliittyviä komponentti- ja sovellusliitintöitä (Application Interface).

Liityntätaso kuvaa rajapinnan, jonka sovellusalueen olio tarjoaa muiden olioiden käyttöön. Normaalisti nämä oliot ovat kohdistettavissa reaali maailman olioihin kuten tuoterään tai koneeseen. Ohjelmointikielessä liityntäkuvaus (IDL) kohdistetaan olioluokkaan.

Komponenttia ei CIM-kehysrakenteessa määritellä olioksi, vaan joukoksi liityntöjä, jotka eri oliot toteuttavat ja joita käyttämällä tietty toiminnallisuus saadaan aikaan. Siten komponentit ovat minimaalinen yksikkö, joka voidaan lisätä, poistaa tai korvata sovelluksessa. Komponentti käsitteenä vastannee perinteistä moduulin käsitettä, mutta komponentilla ei ole yhteistä fyysistä olemusta.

Sovelluspakettitaso edustaa joukkoa komponentteja tai toimittajakohtaista liityntäspesifikaatiota, joka voidaan toteuttaa alijärjestelmänä tai pakata yhteen siten, että se tarjoaa myös *käyttöönotto- ja korvattavuustuen*. Itse CIM-kehysrakenteessa toiminnalliset ryhmät (ks. kuva 3) toimivat perustana sovelluspakettien toteuttamisessa. Sovelluspakettitaso ei ole siis varsinainen tasoajatteluun perustuva arkkitehtuuritaso. Kuvassa 4 on esitetty esimerkki pakettitason käytöstä. Koneen valmistaja tai toimittaja liittää tuotteeseensa CIM-kehysrakenteen mukaisen rajapinnan, joka toimii integrointirajapintana tuotannonohjausjärjestelmän kehittäjälle.

Voisi sanoa yksinkertaisesti, että liityntätason tehtävänä on tukea ohjelmiston sisäistä rakennetta ja uudelleenkäyttöä tarjoamalla tuki tuotekehityksen nopeuttamiseen. Komponenttitaso taas tukee sovellusten rakentamista ja ylläpitoa tarjoamalla tuen järjestelmän inkrementaaliseen kehittämiseen ja laajennettavuudelle. Sovelluspakettitaso toimii järjestelmäsuunnittelijoiden, myyjien ja ostajien tukena järjestelmän tai osajärjestelmän ominaisuuksien määrittelyssä. Yleensä sovelluspaketti määrittelee oman ORB-käytävän ja tarjoaa IIOP-tuen. IIOP on ORB-käytävien välinen kommunikointiprotokolla, jonka kautta eri järjestelmissä käytettävien ORB-toteutusten välinen yhdistyvyys on toteutettavissa. Kehysrakenne ei kuitenkaan tarjoa kuvaustapaa komponenttien ja pakettien kuvaamiseksi, joten komponentti- ja pakettitaso jäävät arkkitehtuurikuvauksessa abstraktisiksi.



Kuva 4. Esimerkki CIM -kehysjärjestelmän mukaisesta sovelluspaketista.

Kehysrakenne perustuu seuraavien mikroarkkitehtuurien hyödyntämiseen:

- Infrastruktuuritasolla CIM-kehysrakenne käyttää CORBA-yhteensopivia palveluja. Kommunikointi perustuu TCP/IP-, NetBios- tai SNA-protokollaan, sen päällä on kerros, joka toteuttaa hajautettujen järjestelmätason palvelujen yhteisen siirto-semantiikan.
- Hajautettuja järjestelmäpalveluja ovat CORBAn kommunikointipalvelut (Messaging and Queueing), olioiden hallintapalvelut (ORB, Life Cycle) ja hajautusta tukevat palvelut, kuten hakemisto-, ajoitus-, turvallisuus- ja transaktiopalvelut.
- Sovellus- ja työryhmäpalvelut tarjoavat tuen tapahtumien, työvirtojen ja transaktioiden monitorointiin ja ovat CORBA-palveluja tai -lisäpalveluja (CORBAservices, CORBAfacilities).

- Esitystapa- ja tiedonsaantipalvelut hyödyntävät hajautettuja järjestelmäpalveluja tarjoamalla tulostus-, katselu- ja verkkoselaus (Web browsers) -palvelut sekä tietokantaliitännät ja pysyvien tiedostojen hallintapalvelut.

Jokaisella kehysrakenteen mukaisella oliolla tulee olla nimi ja viite, joiden avulla nimi- ja palvelukohtaiset sidonnat ovat mahdollisia (CORBA Naming service ja CORBA Trader service).

Laajennettavuus on komponentin kyky tulla paremmaksi erikoistumisen, muuntamisen, konfiguroinnin tai toiminnallisten laajennosten avulla. Periytyminen tukee erikoistumista liityntä- ja toteutusosalle ja muuntamista esimerkiksi johdettujen liityntöjen tai uudelleenmäärittelyn kautta (overriding). Nämä ovat toteutuskohtaisia asioita, joihin spesifikaatio ei puutu. Sitä vastoin kehysrakenne tukee plug-in-toimintoja, joiden avulla järjestelmä voidaan konfiguroida tuote kohtaisesti.

Kehysjärjestelmä tukee uudelleenkäyttöä kolmella tasolla: arkkitehtuuri-, suunnittelu- ja toteutusosalla. Arkkitehtuurin uudelleenkäytössä sovelletaan samoja periaatteita ja ohjeita [9], kun taas suunnittelun uudelleenkäytössä hyödynnetään CIM-kehysrakenteen spesifikaatiota kehittämällä siihen perustuvia järjestelmiä. Toteutusosalla uudelleenkäyttöä tuetaan vaihdettavuus- ja laajennettavuusominaisuuksien avulla.

Järjestelmän joustavuutta tuetaan komponenttiperusteisella arkkitehtuurilla, jota voidaan tarvittaessa muuttaa IR (Interface Repository) -tietokannan kautta. CORBAN Trader-palvelu tuo lisäjoustavuutta käyttöönottovaiheeseen, koska palvelut ovat haettavissa niiden tarjoamien ominaisuuksien perusteella. Suuressa järjestelmässä tämän palvelun käyttö nopeuttaa järjestelmän käyttöönottoa.

Tuotannonohjausjärjestelmät asettavat palvelujen laadulle ainakin seuraavat vaatimukset (QoS, Quality of Service):

1. Suorituskyky: transaktioiden tai viestien lukumäärä sekunnissa, vastausaikojen ennustettavuus reaaliaikavaatimusten täyttämiseksi.
2. Luotettavuus: virheiden väli (MTTF 'mean time to failure'), toipumisväli (MTTR 'mean time to repair').
3. Skaalautuvuus: suorituksessa olevien järjestelmien, olioiden tai prosessien enimmäislukumäärä, kun suorituskykyvaatimukset vielä täyttyvät.

4. Resurssien kulutus: muistin ja kovalevyn kulutus suoritettavien palvelujen tukemiseksi.

Viestinvälitystä (Messaging Service) tukevien liityntöjen on lisäksi tuettava seuraavia laatukriteerejä:

- Asiakas voi lähettää pyynnön ilman, että sen suoritus keskeytyy.
- Asiakas voi lähettää pyynnön, joka ei täyty asiakkaan elinajan puitteissa.
- Asiakas ja palvelin voivat määrittellä palvelun laadun, jota on käytettävä pyynnöissä ja vastauksissa.
- Palvelin voi kontrolloida, missä järjestyksessä pyynnöt palvellaan.

Käytetyn tapahtumapalvelun (Event Service) on taattava, että tapahtumat toimitetaan perille. Käytössä on kolme tasoa: ei takuuta, tapahtumien kuluttajan virheiltä suojautuminen sekä kuluttajan ja tapahtumapalvelun virheiltä suojautuminen. Kun suojaudutaan tapahtuman kuluttajan virheiltä, tilaukset (subscription) on tallennettava pysyvästi, ja tapahtumapalvelun täytyy varastoida tapahtumat väliaikaisesti (spool). Korkeimmassa laatutasossa viestit tallennetaan erikseen niin, että tapahtumapalvelun palattua jälleen toimintaan, tapahtumat voidaan toimittaa perille. Lisäksi tapahtumapalvelulta vaaditaan tuki viestien järjestyksen säilyttämiseksi. Sitä vastoin kehysjärjestelmä ei välttämättä vaadi tapahtumien suodatusta, priorisointia tai vanhentamista.

3. TUOTANNONOHJAUKSEN JOUSTAVUUSTARPEET

Nykyiset tuotannonohjausjärjestelmät voidaan yleisesti luokitella suuriin, keskitettyihin järjestelmiin, jotka on yhdistetty pieniin hajautettuihin automaatiojärjestelmiin. Näissä järjestelmissä esiintyy seuraavia ongelmia [9]:

- rajoittunut joustavuus,
- korkeat integrointikustannukset muihin järjestelmiin yhdistettäessä,
- korkeat kehitys-, laajennus-, ja ylläpitokustannukset sekä
- korkeat hankinta-, käyttöönotto-, ja toimintakustannukset.

Seuraavan sukupolven tuotannonohjausjärjestelmien täytyy ratkaista nämä ongelmat esimerkiksi korvaamalla kaupallisilla vaihtoehdoilla osia ohjelmistoista (komponentoitavuus), kehittämällä integrointitukea niin, että järjestelmät voidaan koota useiden toimittajien tuotteista (standardi liityntäteknikka) ja kehittämällä olemassa oleviin järjestelmiin uusia piirteitä niin, että vanha järjestelmä voidaan pitää toiminnassa samanaikaisesti (laajennettavuus ja skaalautuvuus).

Haastatteluissa esille tulleet käytännön ongelmat voidaan jakaa tuotannonohjausjärjestelmän kehittämisessä, käyttöönotossa, integroinnissa ja varsinaisessa käytössä esiintyviin ongelmiin. Kaksi ensimmäistä luokkaa kuvaavat ongelmia tuotannonohjausjärjestelmän kehittäjän näkökulmasta. Integrointivaiheessa esiintyy sekä järjestelmän kehittäjän että käyttäjän ongelmia. Viimeinen luokka kuvaa ongelmien heijastumista tuotantojärjestelmän loppukäyttäjälle.

3.1 Ohjausohjelmiston kehitys

Tuotannonohjausohjelmiston kehitysvaiheelta puuttuu kuvaustapa, joka tukisi järjestelmän tilaajan, suunnittelijan ja toteuttajan näkökulmaa. Suunnittelijoiden tuntemus integroitujen järjestelmien ongelmista on usein heikkoa ja työmääräarviot liian optimistisia.

Ohjelmiston siirrettävyydessä esiintyy ongelmia. Syynä tähän on siirrettävyyttä tukevien rajapintojen puuttuminen. Tämä johtuu siitä, ettei ohjelmiston uudelleenkäyttöä ole tarpeeksi huomioitu suunnitteluvaiheessa. Lisäongelmia tuottavat laiterajapintojen ja käyttöympäristön rajapintojen muutokset. Yleisen rajapinnan käyttö ei ole aina

mahdollista suorituskyvyn takia. Esimerkiksi ODBC:n hitaus vaatii usein siirtymisen tuotekohtaiseen kyselykieleen, jonka suoritusaika on nopeampi.

Toiminnassa olevan järjestelmän muuttaminen tai uusien piirteiden kehittäminen vanhaan järjestelmään on hankalaa ja kallista. Syynä tähän on inkrementaalista kehittämistä tukevien infrastruktuuripalvelujen puuttuminen. Infrastruktuuri on useimmiten tuotekohtaista: laitteisto pysyy samana, mutta ohjelmisto joudutaan kehittämään lähes täysin uusiksi. Syynä on yleisten rajapintojen määrittelyjen puuttuminen ja siksi jokainen tuote suunnitellaan täysin räätälöitynä.

Windows NT tuotannonohjausjärjestelmien suoritusympäristönä yleistyy. Sen tarjoama tuki tehtävien priorisoinnille on heikkoa eikä se täytä reaaliaikaiselle käyttöjärjestelmälle asetettavia vaatimuksia.

Laitteilta saatavasta tiedosta ainoastaan osa voidaan hyödyntää esim. tilatietojen monitoroinnissa tai vikojen havainnoinnissa. Turhaa tietoa joudutaan siirtämään, koska siirrettävää tietoa ei voi määrittellä tarpeen mukaiseksi. Tämä johtuu siitä, etteivät laiteoimittajat tarjoa yhdenmukaista konfiguroitavaa liityntää.

3.2 Ohjausjärjestelmän käyttöönotto

Järjestelmän integrointitestauksessa ja käyttöönottovaiheessa järjestelmään joudutaan usein tekemään periaatteellisia muutoksia. Tämä johtuu huonosta määrittelystä tai siitä, ettei ohjausjärjestelmän käyttäjä vielä tiedä, mitkä ominaisuudet ovat tarpeellisia tai mitkä ominaisuudet voidaan toteuttaa. Määrittelyjen täydellisyyttä parantaisi yhden kaikille osapuolille ymmärrettävän kuvaustavan käyttö. Rajapintojen muuttuminen aiheuttaa lisätyötä käyttöönottovaiheessa. Koska infrastruktuuri ei tue asteittaista testausta ja käyttöönottoa, käyttöönottovaiheesta muodostuu pitkä ja kallis.

Kun suoritetaan järjestelmän tehtävien uudelleenjakoa, järjestelmää skaalataan lisäämällä esimerkiksi palvelin. Tämä aiheuttaa tiedon monistusta ja kommunikoinnin lisääntymistä väylällä. Lisäksi ohjelmiston heikko siirrettävyys tuottaa lisäongelmia skaalautuvuudelle.

Käyttöönottovaiheessa esiintyy usein poikkeustilanteita, joiden käsittelyä on vaikeaa tai mahdotonta ennakoida, koska ohjausjärjestelmä suunnitellaan sen normaalia käyttäytymistä silmällä pitäen eikä mahdollisista poikkeustilanteista ole tarpeeksi tietoa saatavilla.

3.3 Järjestelmän integrointi

Laitetoimittajat jättävät usein liityntärajapinnan määrittelemättä. Liitynnän toiminta selvitetään erityisesti sitä varten kehitetyillä kuunteluohjelmilla. Vaikka selvä tietoformaatti olisikin olemassa, liitynnän käyttö vaatii erillisen konversio-ohjelman toteuttamista. Konversio-ohjelmat kertaantuvat jokaisen liitynnän kohdalla ja monimutkaistavat ja hidastavat järjestelmän toimintaa.

Tietokantojen lukitukset ovat ongelmana, mikäli käytetään useita erilaisia tietokantapalvelimia. Syynä on valmistuotteiden lukitusmekanismien erilainen tuki.

Kellojen synkronointi aiheuttaa ongelmia. Pulmien ratkaisemiseksi tarvittaisiin yleiset hajautetut ajoituspalvelut. Kellojen synkronointi suoritetaan yleensä määrävälein pitäen yhtä kelloa referenssikellona. Koska infrastruktuuri ei tarjoa yleistä palvelua globaaliin ajoitukseen, jokaisessa tuotteessa palvelu ratkaistaan erikseen. Tämä aiheuttaa lisätyötä järjestelmän integrointivaiheessa ja tuotannonohjauksen integroimisessa esim. yritystason järjestelmään.

Integrointivaiheessa rajapintoihin joudutaan tekemään toistuvasti muutoksia. Syynä tähän on rajapintojen heikko suunnittelu ja niiden puuttuminen täysin arkkitehtuurivaiheen kuvauksista.

Tuotannonohjausjärjestelmän tehtävänä on tuottaa tietoa järjestelmän erilaisille päätöksentekijöille, kuten työnjohtajille, osastoinsinöoreille, laatupäälliköille ja johdolle. Tämä tarkoittaa tiedon kontekstikohtaista esittämistä.

3.4 Järjestelmän käyttäminen

Loppukäyttäjän näkökulmasta tuotannonohjausjärjestelmä ei tue riittävästi työtehtävän suorittamista. Tiedon esitystapa eli käyttöliittymä tulisi pystyä säätämään hyödyntäjän tarpeen mukaisesti. Käyttöliittymän tulisi tarjota työn kannalta oleelliset asiat helposti omaksuttavassa muodossa.

Poikkeustilanteiden käsittely, kuten käsin tehtävät pienet erät, on hankalaa. Käytännössä tiedot syötetään käsin järjestelmään. Syynä poikkeustilanteiden käsittelyn hankaluuteen on järjestelmän jäykkyys. Esimerkiksi SAP R/3 tukee kuitenkin erittäin hyvin jäljitettävyyttä, mutta on toisaalta työläs mukauttaa erilaisiin tarpeisiin. Järjestelmän joustavuustarpeella on siten ainakin tässä tapauksessa ristikkäisvaikutus jäljitettävyyteen. Ristikkäisten vaatimusten tapauksessa kompromissien tekeminen on usein työlästä. Järjestelmän jäykkyys ilmenee myös tuotantosuunnitelmissa tehtävien

muutoksien aiheuttamissa ongelmissa materiaalin seurannassa. Järjestelmän jäykkyys haittaa myös virheiden korjausta, jonka tuki on puutteellinen.

Käyttäjää paremmin palvelevat toiminnot ovat usein vaikeita toteuttaa. Syynä saattaa olla käytetty verkkotopologia, koska tiedonsiirtotarpeen kasvaessa tietojen välittäminen esim. yhden palvelimen kautta voi muodostua pullonkaulaksi. Mikäli hajautettujen sovellusten kommunikointi ei ole läpinäkyvää, verkkotopologia heijastuu sovellukseen ja pullonkaulan poistaminen aiheuttaa paljon lisätyötä.

3.5 Käytetyt ratkaisumallit

Järjestelmän joustavuutta on lisätty seuraavilla erityisratkaisuilla:

- Tuotannossa tapahtuvien suurien poikkeamien, joilla tarkoitetaan järjestelmän sallimien normaalien poikkeamien ylittämistä, käsittelemiseksi on käytetty ylimääräisiä jonotusmekanismeja. Joustavuusmekanismeja ei ole tarpeeksi huomioitu kaupallisissa ratkaisuissa eikä järjestelmää voida konfiguroida tapauskohtaiseksi.
- Sama käyttöliittymä toimii eri näkymien välittäjänä. Käyttöliittymät luokitellaan työtehtävän mukaan: tuotannosuunnittelu, tuotannon aktivointi, tuotteen tunnistaminen, poikkeustilanteiden hoito ja pakkaus. Luokittelu tapahtuu siis työvirran ja käyttötoiminnon määrittelemän kontekstin mukaisesti.
- Suunnitelmien muuttuessa käyttäjä puuttuu järjestelmän toimintaan, joten poikkeustilanteiden käsittelemistä varten on toteutettu erikoiskäyttöliittymä. Poikkeustilanteet saattavat kuitenkin vaihdella ja uudenlaisia poikkeuksia myös ilmenee.

3.6 Kehittämiskohteet

Elektroniikkateollisuudelle suunnatun kyselytutkimuksen perusteella onnistuneen tuotannonohjauksen tärkeimpiä ominaisuuksia ovat järjestelmän avoimuus ja standardiperusteisuus sekä käyttäjäturvallisuus. Tärkeimpinä kehittämiskohteina pidettiin seuraavia [15]:

- materiaalin ja tuotteiden jäljitettävyyys,
- tuotantokapasiteetin käytön optimoinnin kehittäminen,
- materiaali- ja tietovirtojen reaaliaikaisen hallinnan kehittäminen,

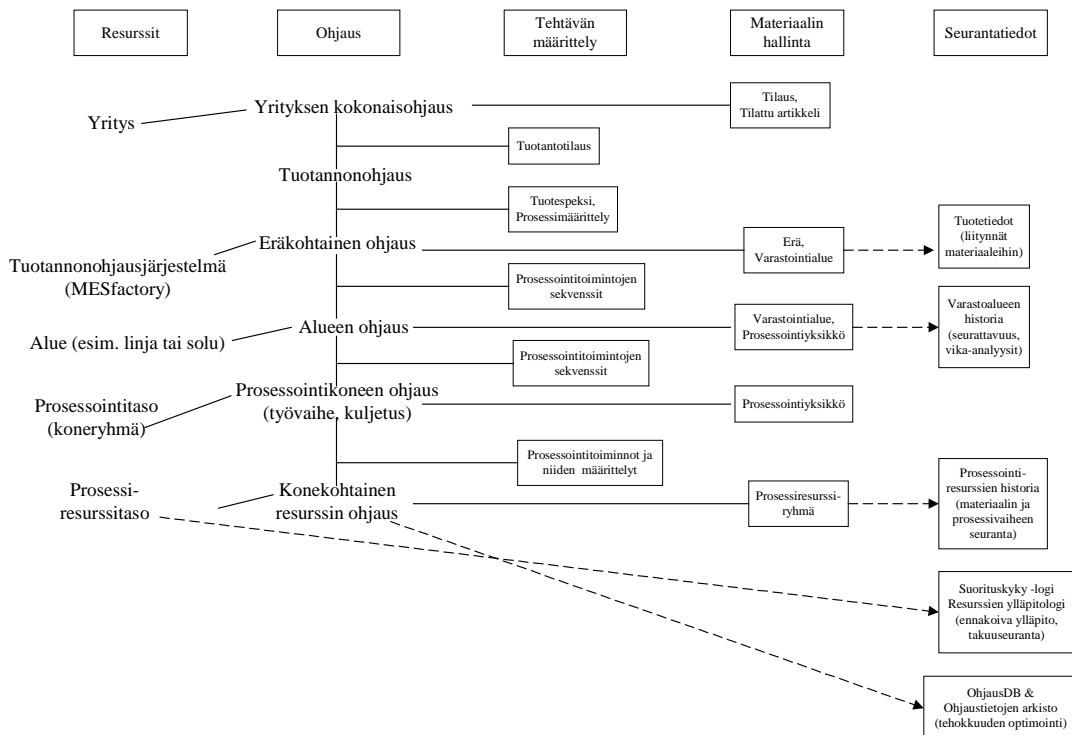
- asetusajkojen lyhentäminen,
- laadun ja prosessin valvonta sekä
- saannon parantaminen.

Neljä ensimmäistä kohtaa tähtäävät järjestelmän toimintavarmuuden lisäämisen kehittämiseen, ja edistävät kahden viimeksi mainitun kehittämiskohteen saavuttamista.

4. CIM-KEHYSRAKENTEN TARJOAMA TUKI OHJELMISTON JOUSTAVUDELLE

4.1 Jäljitettävyys ja optimointi

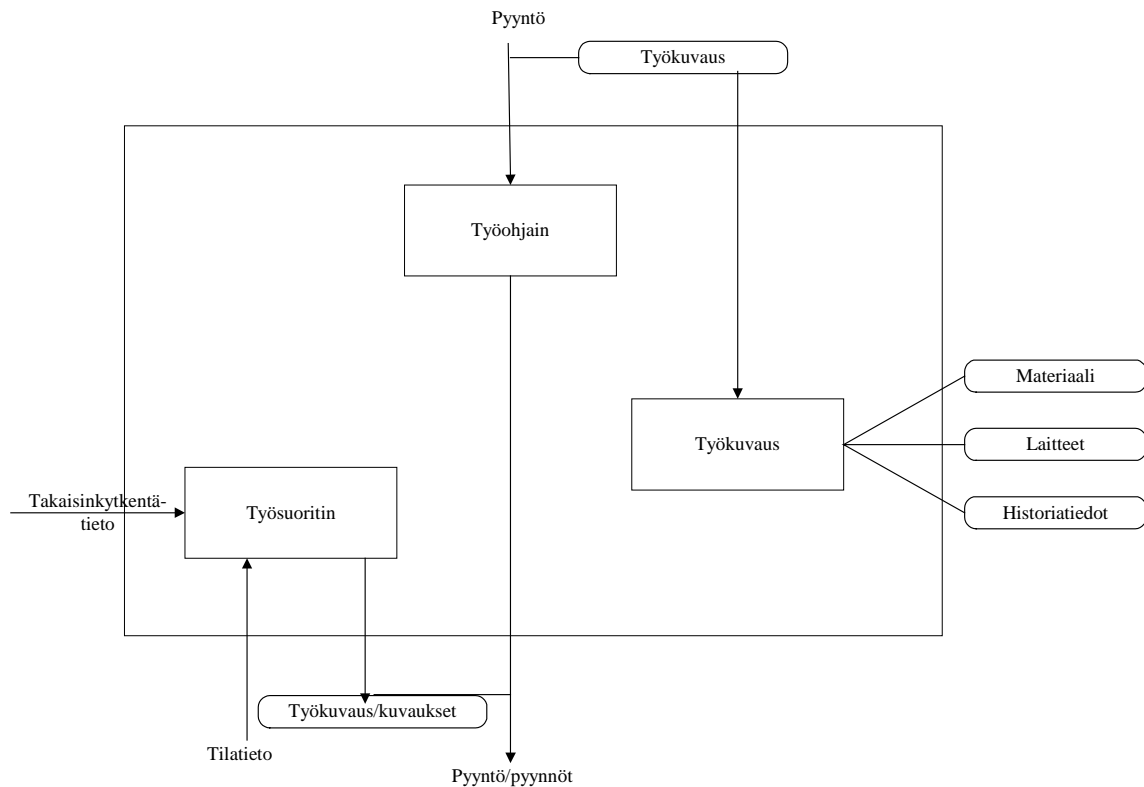
CIM-kehysrakenne perustuu hierarkkisten työtehtävien määrittelyyn ja suoritukseen. Kuvassa 5 on esitetty yleiskuva ohjauksen eri tasoista. Resurssit määrittelevät työn suorittamiseen tarvittavat fyysiset laitteet tai laiteryhvät. Tuotantolinjan ohjausjärjestelmä muodostuu konekohtaisten resurssien ohjauksesta, prosessointiyksikön ja linjan ohjauksesta. Tehtävä määritellään jokaisella tasolla käyttäen siihen tarkoitettua kuvaustapaa. Eräkohtaisesta ohjauksesta päätetään tuotannonohjaustasolla (MESfactory). Materiaalin hallinta liittyy jokaiseen ohjaushierarkiaan. Seurantatietoja kerätään eräkohtaisesti tuotteista ja niihin käytetyistä materiaaleista. Materiaalien käytön seuraaminen ja vika-analyysien teko tapahtuu linjoittain. Tarkemmin materiaaleja ja prosessia seurataan yksittäisen koneen tasolla. Tämä mahdollistaa prosessi- ja materiaaliressurssien käytön tarkemman synkronoinnin ja optimoinnin. Mikäli halutaan ennakoivaa ylläpitoa, siitä huolehtii koneohjain. Ohjaustietokannan ja ohjaustietojen arkistointi mahdollistaa prosessin tehokkuuden optimoinnin resurssikohtaisesti.



Kuva 5. Hierarkkiset työtehtävät ja niiden kytkennät materiaalinhallintaan ja seurantatietoihin.

Tehtävät (Task) kuvataan hierarkkisina tehtävärakenteina: tehtäväproseduureina, työvirtana tai prosessikuvauksena. Jokainen tehtävä kuvataan tehtäväkuvauksen (Task Specification) avulla ja tehtävä aloitetaan aloitustapahtuman (StartEvent) toimesta ja lopetetaan tehtävälle määritellyn päätöstapahtumaan (EndEvent). Tehtävä voidaan jakaa pienempiin osatehtäviin, jotka kaikki noudattavat samaa määrittely- ja käsittelytapaa. Kun tehtäväkuvaukseen kiinnitetään käytettävä materiaali ja työn suorittavat fyysiset resurssit eli koneet, määrittelystä muodostuu työkuvaus (Job), jonka suoritusta työkontrollerit (Job Controls) ohjaavat tallentaen ohjaukseen liittyvät historiatiedot resursseista, ohjauksesta ja käytetystä materiaalista. Työkontrollerit ovat hierarkkisia, ja kommunikointi eri tasojen välillä tapahtuu JobRequest - JobStatus -viestien välityksellä.

Työohjaimet muodostuvat seuraavista osista: työohjaus (Job Supervisor), työsuoritin (Job Executor) ja työkuvaus (kuva 6). Työohjain ottaa työpyynnön (JobRequest) vastaan, jakaa sen eri vaiheisiin, valitsee materiaalit ja laitteet jokaiselle vaiheelle skedulointialgoritmia hyödyntäen, koordinoi ja jakaa tehtävät pyyntöinä alemmalle ohjaustasolle. Työsuoritin tarkkailee työn etenemistä alemmalla ohjaustasolla ja muuttaa nykyisen tai myöhempien vaiheiden määrittelyjä, mikäli se on tarpeen. Työtiedot itsessään sisältävät nykytilan ja historiatiedot pyydetyistä ja delegoiduista työtehtävistä sekä allokoituista materiaaleista ja resursseista sekä töiden statustiedoista ja tapahtumista. Työohjaimen toteutustapa saattaa vaihdella eri hierarkiatasoilla, mutta jokainen työohjain sisältää silti tehtävien osituksen ja allokoinnin, suorituksen ohjauksen ja seurannan sekä tilannetietojen taltioinnin. Kuvassa 6 on esitetty vaihtoehto työn ohjauskomponentin toiminnalliseksi rakenteeksi.



Kuva 6. Esimerkki työohjaimen toiminnallisesta rakenteesta.

Tuotantolaitostasolla tehtävien ajoittamisesta huolehtivat tuotannon työohjain (Factory Job Supervisor), joka käyttää skedulerin palveluja työtehtävien optimoimiseksi ja jakelijan (dispatcher) palveluja tehtävien suorittamiseksi oikeassa järjestyksessä. Mikäli erillistä skeduleria ei käytetä, jakelija huolehtii myös päättelystä käyttäen omia sääntöjään oikean jakelulistan muodostamiseksi. Erillistä skeduleria ei välttämättä tarvita, mutta jakelija on välttämätön. Töiden jakelu voidaan toteuttaa jakelijan toimesta esimerkiksi priorisoituna tehtävälistanä, joka välitetään operaattorin, useimmiten työjohtajan, hyväksyttäväksi. Interaktiiviseen tuotannonohjaukseen voidaan sitoa myös ylemmän tason optimointi, mikäli jakelija pystyy ottamaan huomioon tiedot tuotantotieristä, niihin tarvittavista materiaaleista ja materiaalien sijainnista.

CIM-kehysrakenne määrittelee erän (Lot) materiaaliksi ja operaatioiksi, jotka tarvitaan tuotteen tekemiseksi. Työohjain (Job Supervisor) suorittaa eräkohtaisesti työn hallinnan ja lattiason ohjauksen työkuvausten perusteella, mutta ei tee materiaalin ja tuotteen varsinaista seuranta. Erän prosessivirta määritellään prosessispesifikaation avulla, ja erän nykytilaa seurataan prosessivirran kontekstin (ProcessFlowContext) avulla merkitsemällä erän sijainti suhteessa prosessivirtaan. Toisin sanoen merkintä osoittaa toimintospesifikaation, joka kulloinkin on aktiivisena.

4.2 Yleiset rajapinnat ja palvelut

CIM-kehysrakenne määrittelee CORBA IDL:n avulla yleiset tyyppi- ja poikkeusmäärittelyt, joista tässä mainitaan vain muutama prosessin ja laitteiden tilaan liittyvä määrittely:

- olion tilat elinkaaren aikana,
- koneiden ja koneresurssien tilat,
- historiatietojen kirjaustapa,
- historiatietojen jatkokäsittelyä tarkentavien suodattimien määrittely ja
- suorituskontekstin määrittely.

CIM-kehysrakenteen ytimen muodostavat kuitenkin komponenttimäärittelyt ja abstraktit liityntäryhmien määrittelyt, joiden toteuttaminen tai periytyminen kehysrakenteen oliolle toteuttaa yhteensopivuuden, joka vaaditaan vaihdettavien sovellusten kehittämisessä. Komponentti määritellään joukoksi liityntöjä, jotka yhdessä toteuttavat tietyn toiminnallisuuden. Komponentin liitynnät toteutuvat oliolle periytymisen tai erikoistumisen kautta.

Abstraktien liityntäryhmien tavoitteena on auttaa ymmärtämään, toteuttamaan ja käyttämään CIM-kehysrakennetta. Nämä liityntäryhmät muistuttavat läheisemmin perinteisiä liityntäkirjastoja kuin komponentteja, ja ne toimivat abstrakteina liityntöinä toimittajakohtaisille liitynnöille. Abstraktit liitynnät jaetaan kahteen luokkaan: komponenttihallinta- ja historialiityntä ovat esimerkkejä yleisistä liitynnöistä, kone- ja materiaali-liityntä sovellusaluekohtaisista liitynnöistä.

Perusliitynnät tarjoavat joukon liityntöjä ja palveluja, joita käytetään läpi koko CIM-kehysrakenteen:

- *Resurssien abstrakti liityntäryhmä* sisältää liitynnät nimettyjen entiteettien käsittelemiseen, entiteettien omistajan määrittelemiseen, aktiivisten entiteettien eli resurssien käsittelyyn sekä komponenttien hallinnan, joka tarjoaa liitynnät komponenttien liityntöjen rekisteröintiin, aktivointiin ja komponentin sisältämien resurssien hallintaan.
- *Työnohjauksen abstrakti liityntäryhmä* tarjoaa abstraktit palvelut työn luontiin, suorittamiseen ja hallintaan. Liityntäryhmä muodostuu työn pyytäjän (Job Requestor),

työn ohjaajan (Job Supervisor) ja työmäärittelyn (Job) liitynnöistä. Käyttäjä määrittelee halutun työn, joka aktivoituu pyytäjän ja ohjaajan tekemän sopimuksen (request-reply) välityksellä. Työnohjaustasot ovat hierarkkisia ja ylemmän tason ohjaus voidaan jakaa pienempiin osiin alemman tason suoritettavaksi. Työn pyytäjä voi rekisteröityä myös työn (Job) tilatietojen käyttäjäksi, mikäli pyytäjä haluaa väliaikatietoja työn edistymisestä.

- *Tehdastason palvelut* tarjoavat tehdastason toimintaa tukevat palvelut kuten dokumenttien hallinnan, versionhallinnan, ajoituspalvelut, tapahtumien välittäminen ja historiatietojen hallinnan. Dokumenttien hallintaa tarvitaan tuotespesifikaatioiden hallinnan tueksi. Mikäli muutostietoja halutaan myös hallita, tarvitaan lisäksi versionhallinnan palvelut. Ajoituspalvelut toteutetaan CORBAn Time-palvelulla. Tapahtumien välittämiseen käytetään tulevaisuudessa CORBAn Trader-palvelua. Historiatietojen hallintapalvelut sallivat historiatietoihin sisällytettävän myös aikaleimatut tapahtumat, mikä mahdollistaa aikaohjattujen tapahtumien ja niiden rakenteiden keräilyä, hallinnan ja navigoinnin diagnostiikkatarkoituksiin. Historiatiedot voidaan toteuttaa myös hierarkkisinä. Historiatietojen tulisi olla kaikkien olioiden nähtävissä ja käyttäjäksi rekisteröinti tapahtuu CORBA Trader-palvelun avulla. Kuitenkin historiakokoelmien hallinta, olioiden välisten relaatioiden määrittely ja kyselyjen teko relaatioihin perustuen tulisi olla CORBA-palveluiden tehtävä, mutta käytännön toteutukset puuttuvat vielä useilta osin.

4.3 Toiminnalliset ryhmät ja komponentit

CIM-kehysrakenne tukee perinteisiä toiminnallisia kokonaisuuksia ryhmittelemällä loogisesti samaa toimintoa tukevat komponentit yhteen. Seuraavat toiminnalliset ryhmät eivät ole arkkitehtuuriyksiköitä, mutta auttavat ongelmakentän jäsentämisessä:

- Tehtaanhallinta (Factory Management)
- Materiaalinhallinta (Material Management)
- Prosessikuvausten hallinta (Process Specification Management)
- Materiaalinsiirto (Material Movement)
- Koneenohjaus (Machine Control)
- Laajennettu prosessinohjaus (Advanced Process Control)

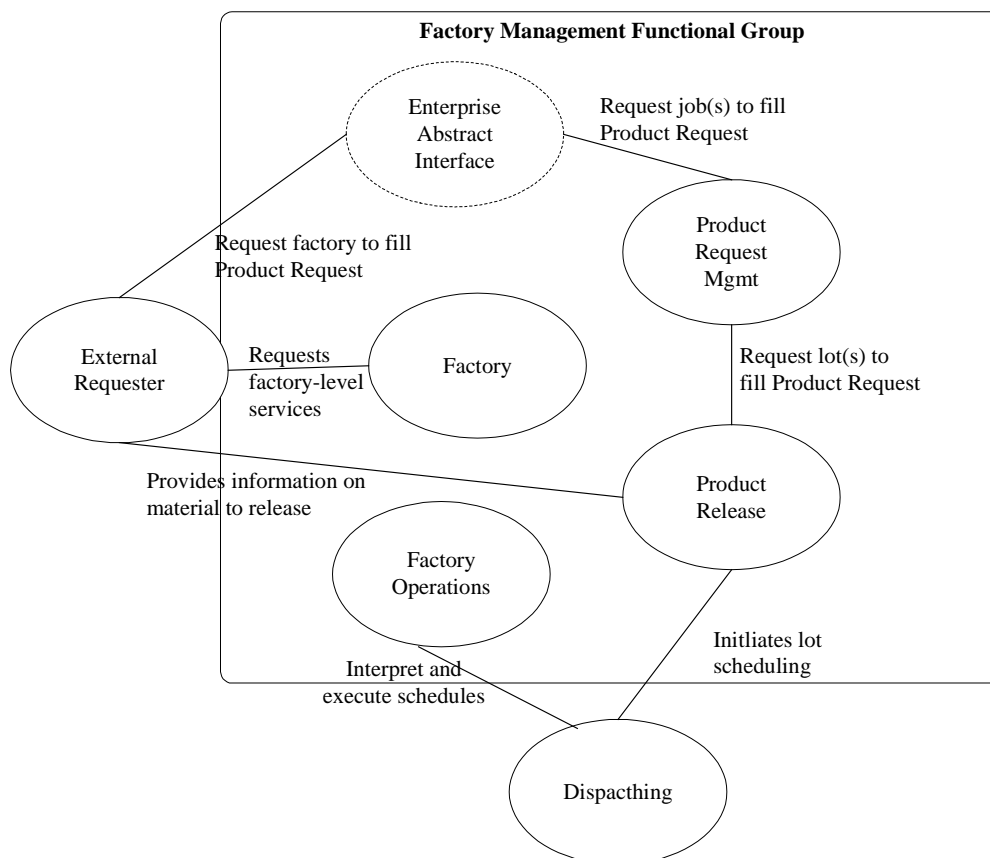
- Aikataulutus (Scheduling Management)
- Työntekijät (Factory Labor).

4.3.1 Tehtaanhallinta

Komponentti sisältyy tehtaanhallintaryhmään, mikäli se tarjoaa tehtaantason palveluja kuten tehdastason konfigurointia tai toimintaa tukevia palveluja tai tuotantoerien seurantaan tukevia palveluja. Ryhmä sisältää seuraavat komponentit ja abstraktit liitynnät (kuva 7):

- Factory-komponentti tarjoaa palvelut konfiguroinnille, monitoroinnille, ohjaukselle, resurssien ryhmittelylle, kalenteri- ja kyselypalveluille.
- FactoryOperations -komponentti tulkitsee tuotantoaikataulut ja alustaa tarvittavat tehdastason aktiviteetit. Komponentti muodostaa tehdasjärjestelmän hermokeskuksen ja on JobSupervision-komponentista johdettu tehdastason ohjain. Automatisoidussa ohjauksessa tämä komponentti tekee päätökset, mitä tuotetta tehdään, skedulerin ja jakelijan avustuksella. Ylimmän tason optimointitehtävät tulisi sijoittaa tähän komponenttiin.
- Enterprise Abstract Interface -liityntä tarjoaa perusliitynnät yritystason informaatiojärjestelmiin, jotka lähettävät tuotantopyynnöt tehdastason järjestelmälle (Factory system). Näiden liityntöjen avulla haetaan tuotantopyyntöön liittyvä tilaus, tuotantospesifikaatio ja muut tarvittavat tuotantopyyntöön liittyvät tiedot. Tuotantospesifikaatioon ei voi tehdä muutoksia tässä vaiheessa. Muutokset on tehtävä ylemmällä tasolla eli yritystasolla.
- Product Request Management -komponentti hoitaa yritys- ja tuotantotason välistä toimintaa hyväksymällä tuotantopyynnöt ja jakamalla ne tuotantoeriin, jotka annetaan tuotannolle optimoituina tuotantosuunnitelmina. Tämä komponentti sisältää kaksi pääaluetta: se määrittelee tuote-erät siten, ettei yli- tai alikuormituksia tapahdu ja kiinnittää tuotantotilaukset eriin. Linjojen kuormituksen optimointi tulisi tapahtua tässä komponentissa.

- Product Release -komponentti luo tuotantoerät suunniteltujen ja hyväksytyjen tuotantopyyntöjen perusteella ja vapauttaa erät varsinaiseen tuotantovaiheeseen. Viimekäden optimointi suoritetaan tässä vaiheessa yhdistämällä tai jakamalla tuotantoeriä linjojen senhetkisen tilanteen mukaan. Uudelleenryhmittely on kuitenkin tapahduttava ennen tuotantoerän luontia. Komponentti on osa tilaus - suunnittelu - tuoteversio-sekvenssiä, joka ottaa asiakkaan tilauksen ja luo Tilaus-Tilattu_tuote-Tuotantotilaus-Erä-olion.



Kuva 7. Tehdastason ohjauskomponentit ja niiden väliset suhteet.

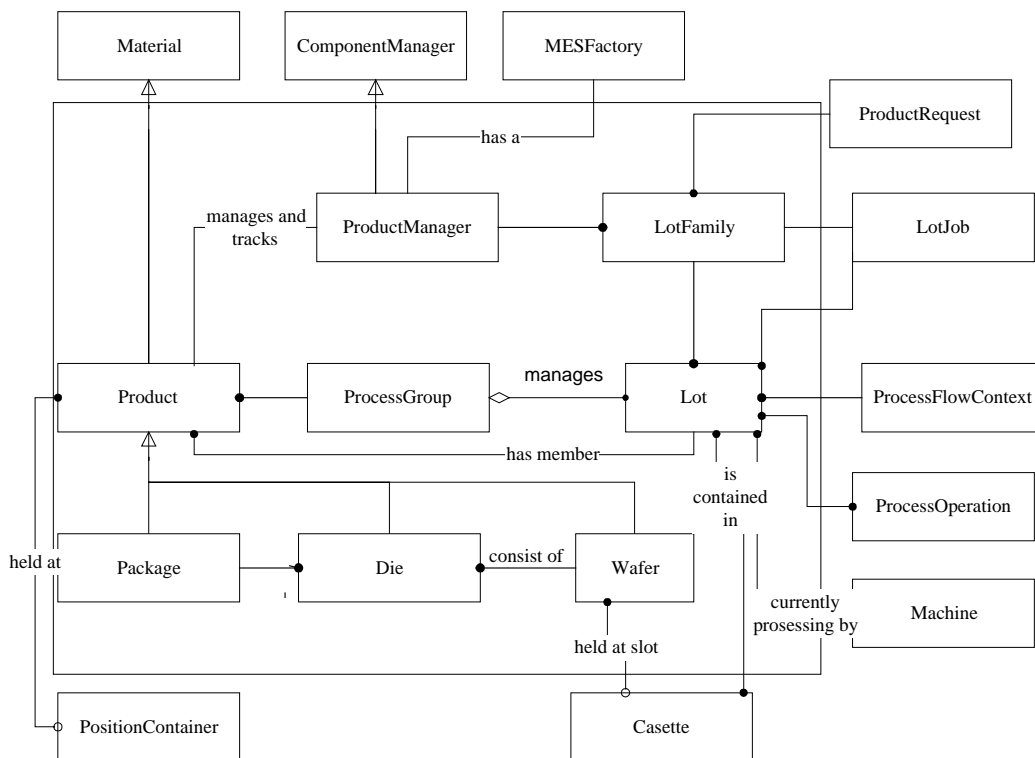
4.3.2 Materiaalinhallinta

Materiaalinhallintaryhmä sisältää materiaalinhallinnan abstraktit liitynnät sekä komponentit, jotka määrittelevät ja hallitsevat tuotteiden, erien, pitkäikäistavaroiden ja kulutustavaroiden ohjelmistolliset esitystavat. Tämä ryhmä ohjaa materiaalien elinkaarta ja ryhmään kuuluvat seuraavat komponentit:

- Product Management -komponentti vastaa tehdastason tuotetietojen hallinnasta.

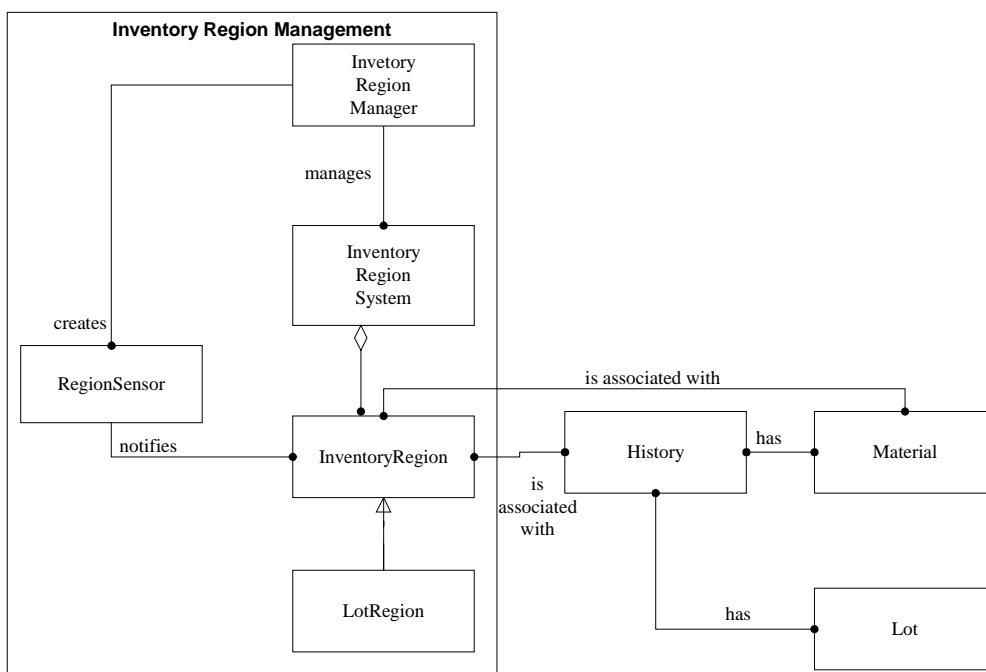
- Durable Management -komponentti vastaa tuotannossa tarvittavien pitkäikäisten kappaleiden, kuten kasettien, kiinnittimien ja hiusristikkojen hallinnasta.
- Consumable Management -komponentti tarjoaa palvelut kuluvien tarvikkeiden kuten kiinteiden ja nestemäisten aineiden seurantaan.
- Inventory Region Management -komponentti huolehtii materiaali-prosessin kulusta ja siitä, mitä materiaalia missäkin osaa tuotantoa sijaitsee.

Materiaalihallinnan abstraktit liittynät palvelevat kuljetusjärjestelmää, koneiden prosessointia ja materiaalin seurantaan. Kuva 8 esittää suorakaiteen sisällä Product Management -komponentin tarjoamat liittynät ja kytkennät muihin komponentin ulkopuolisiin liittyntöihin. ProductManager hoitaa tuotteisiin ja tuote-eriin liittyvien olioiden elinkaareen ja koordinointiin liittyvät hallintatehtävät ja on siten olennaisinta materiaali- ja erävirtojen seurannassa. CIM-kehysrakenne määrittelee myös signaalit, jotka aiheutuvat, mikäli tuote-erään liittyviä tietoja ei löydy tai tiedot eivät vastaa määrittelyjä. Product Management -komponentti muodostaa seurantajärjestelmän ytimen.



Kuva 8. Product Management Komponentin liittynät ja suhteet komponentin ulkopuolelle.

Komponenttien tarkemmat määrittelyt samoin kuin muista materiaalin hallintaan liittyvistä komponenteista on saatavissa viitteestä [10]. Muista materiaalin hallintakomponenteista kannattaa käsitellä tarkemmin Inventory Region -komponentti (kuva 9). Komponentti on joukko integroitavia alueita, jotka kattavat yhdessä täydellisesti kaikki kiinnostavat materiaalit valituissa asiayhteyksissä. Asiayhteydet voivat olla fyysisiä, kuten koneita, tai loogisia, kuten prosessivirran segmentteja. Kukin seurattava materiaali määritellään vain yhteen asiayhteyteen kuuluvaksi. Tätä komponenttia hyödynnetään, kun halutaan mitata ja seurata tuote-erän edistymistä prosessivirrassa käyttäen LotRegion -liittynän palveluja. RegionSensor -liityntä taas tarjoaa keinot, joilla seurattavat asiat kiinnitetään seurantajärjestelmään (InventoryRegions).



Kuva 9. Inventory Region Management -komponentin informaatiomalli.

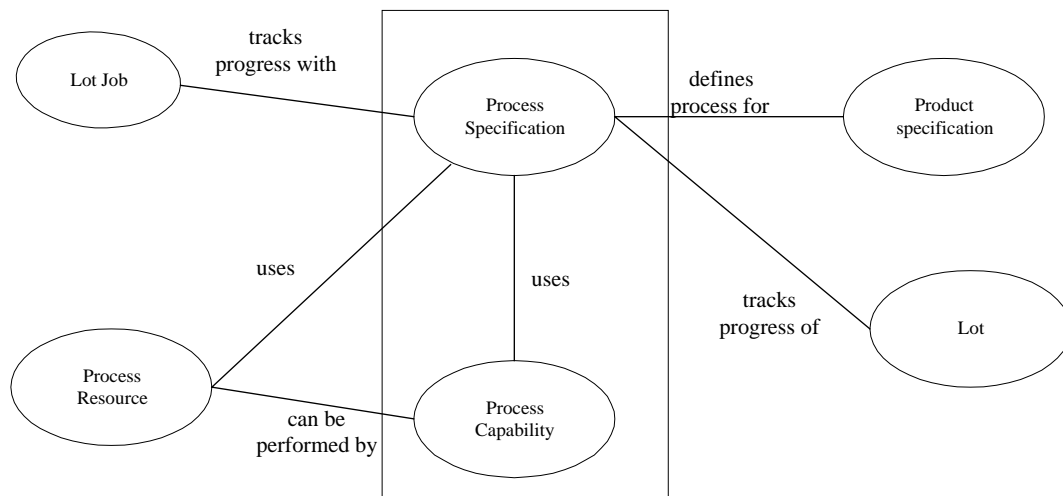
Tuotteeseen ja tuote-eriin liittyvät läheisesti materiaalinmäärittely- (Bill of Material Specification) ja tuotemäärittelykomponentit. Kun BOM-spesifikaatio on kerran luotu, se linkitetään tuotespesifikaatioihin, jotka sitä käyttävät. Molemmat spesifikaatiot ovat staattisia MESFactoryn kannalta. Tuotespesifikaatiota käytetään prosessivirtakontekstin (ProcessFlowContext) luomiseksi tuote-erälle.

4.3.3 Prosessikuvausten hallinta

Tähän ryhmään kuuluvat komponentit tuottavat arkkitehtuurin, jolla CIM-järjestelmän prosessispesifikaatioita hallitaan ja käytetään. Komponentit mahdollistavat prosessien ajonaikaisen käytön ja manipuloinnin tukemalla koneriippumattomien prosessikuvausten tekemistä, jotka ovat kuitenkin koneiden luettavissa. Ryhmä muodostuu seuraavista komponenteista:

- Process Specification Management -komponentti tarjoaa liittynät, joiden avulla on mahdollista tarkastella prosessitoimintojen sekvenssia ja seurata tuotantoerän nykyistä prosessitoimintoa.
- Process Capability Management -komponentti luo keinot määrittellä, mitä toimintoja kukin prosessiresurssi voi käyttää.

Spesifikaatio ei määrittele komponentteja tai liittynöjä, jotka tarvitaan prosessikuvausten tekemiseksi, vaan ainoastaan rajoitetut liittynät etukäteen määriteltyjen prosessikuvausten käyttämiseksi (kuva 10).



Kuva 10. Prosessikuvausten hallinnan komponenttisuhteen.

Prosessin kuvauskomponenttiin (Process Specification) sisältyvän ProcessOperation -liittynän tehtävänä on antaa prosessoitavan tuotteen ajonaikaista seurantatietoa. Dynaaminen informaatio, joka on määritelty prosessioperaatiokuvauksessa, saadaan selville prosessivirtakontekstin avulla. Informaatiota voidaan käyttää prosessoinnin aikana tai tallentaa historiatietoihin.

Process Capability Management -komponenttia käytetään tunnistamaan resurssien yleiset kyvyt ja hallitsemaan niiden resurssien väliset linkit, jotka pystyvät suorittamaan tietyn toiminnon.

4.3.4 Materiaalinsiirto

Materiaalinsiirtopalvelut muodostuvat Material Movement Management -komponentista ja materiaalin siirtoa palvelevasta abstraktista liityntäryhmästä. Tehdastason muut liitynnät käyttävät ainostaan Material MovementManager -liityntää hyväkseen. Abstrakteista liitynnöistä käytetään MaterialMovementController-, TransportJob-Supervisor- ja TransportJob -liityntöjä. Asiakasolio käyttää ohjauskomponenttia pyytääkseen, seuratakseen tai ohjatakseen materiaalin siirtoa. Movement-Controller hyväksyy pyynnön, aikatauluttaa ja suorittaa siirrot. Yksittäinen siirto suoritetaan ja monitoroidaan TransportJob -liitynnän avulla.

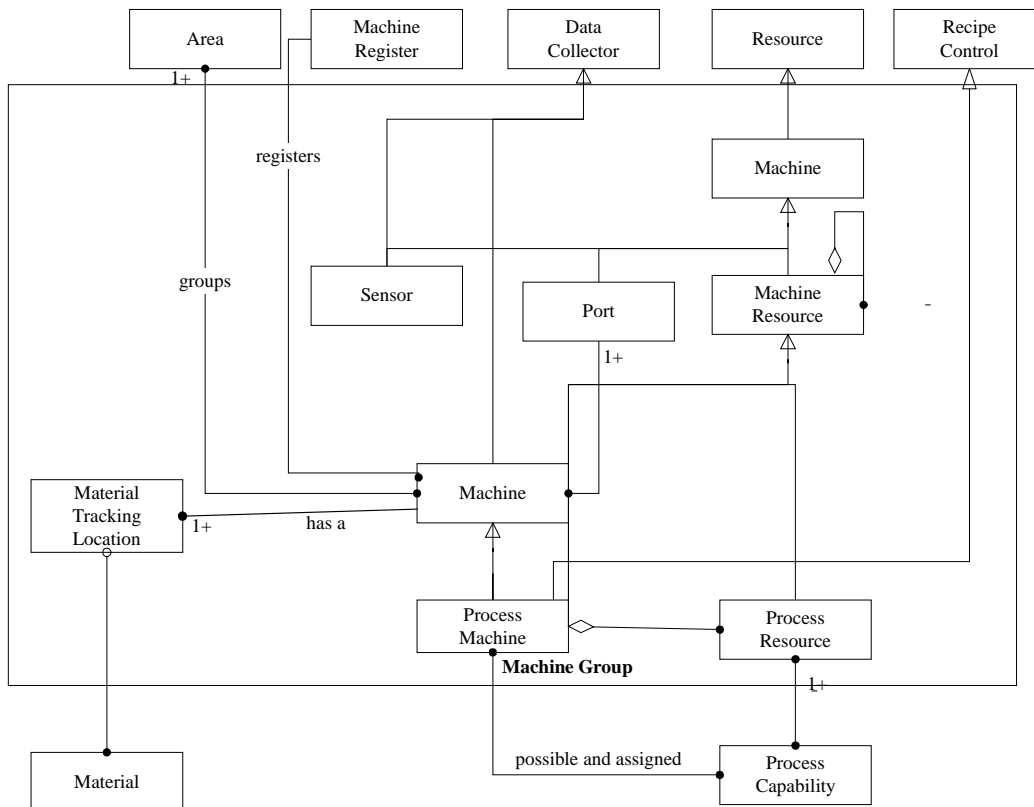
4.3.5 Koneenohjaus

Tämä komponenttiryhmä tuottaa perusarkkitehtuurin koneen tai laitteen ja sitä vastaavien tukiresurssien, kuten reseptien, esittämiseksi. Koneenohjausryhmään kuuluvat seuraavat komponentit:

- MachineGroup sisältää liitynnät koneiden ja resurssien hallintaan ja työprosessin ohjaukseen sekä materiaalin seurantaan.
- RecipeManagement tarjoaa reseptien hallinnan ja taltioinnin prosessiresursseille.
- Resource Tracking on tarkoitettu resurssien ylläpidon tueksi.

MachineGroup ja Resource Tracking liittyvät osittain materiaalin tai resurssien seurantaan ja historiatietojen tallentamiseen, joten koneenohjausryhmästä käsitellään ainoastaan näiden komponenttien toimintaa.

CIM-kehysrakenne kuvaa koneen informaatiomallin yleisellä tasolla ja prosessointikoneen mallin erikoistapauksena. Koneiden määrittelemiseksi tarvitaan kuvassa 11 esitetyt liitynnät.



Kuva 11. Machine Group -komponentin informaatiomalli.

Koneella on paikka-attribuutti, joka vastaa fyysisen laitteen paikkaa tehtaassa. Sillä on myös jokaista materiaalia kohti yksi materiaalin jäljityspaikka (MaterialTracking-Location) jota kone käyttää, säilyttää tai tarvitsee. Tämän jäljityspaikkaliittynän kautta voidaan kysellä, mitä materiaalia koneelle tarvitaan tai mikä on tietyn materiaalin tilannetieto.

ResourceTracking -komponentin avulla määritellään, aikataulutetaan ja jäljitetään tuki-resurssien tilanne ja huolehditaan niiden ylläpidosta eli komponentti monitoroi resurssien tilaa ja ilmoittaa tarvittavista huolloista PM (Preventive Maintenance) vaatimusten mukaisesti. Monitoroinnissa ei käytetä antureita resurssien kunnon havainnointiin, vaan tarkkailu perustuu ennalta suunniteltujen tarkastus- ja huoltomäärittelyjen tarkastamiseen.

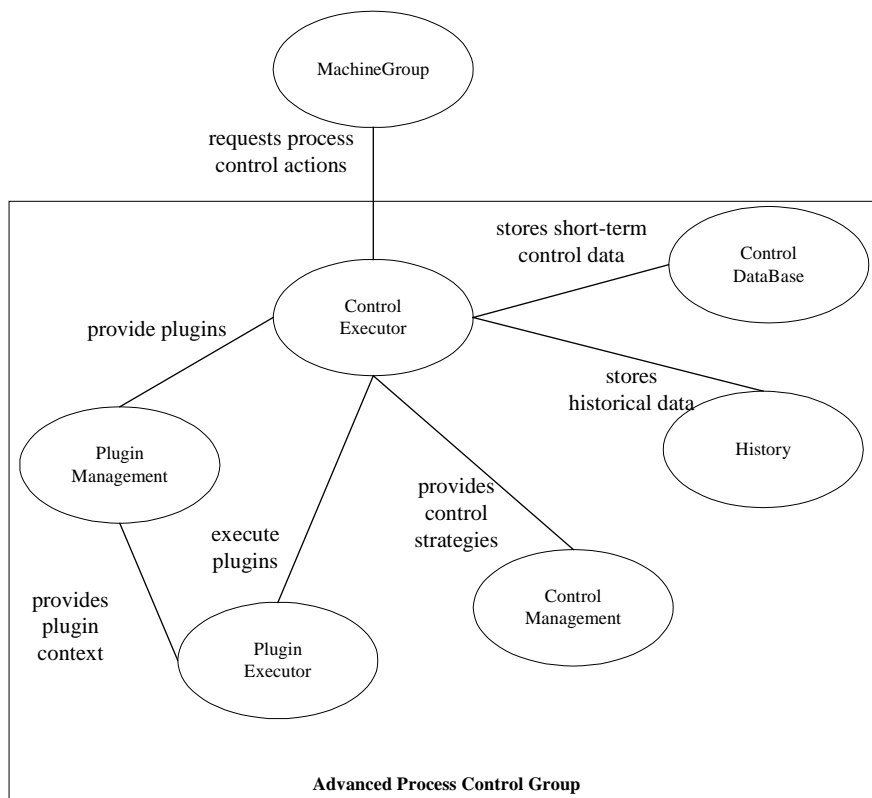
4.3.6 Laajennettu prosessinohjaus

Laajennetun prosessinohjauksen avulla ohjausjärjestelmään voidaan integroida ylemmän tason ohjaus, jossa vikojen havainnointi ja luokittelu sekä jatkuva-aikainen (run-to-run) takaisinkytkentä- tai myötäkytkentäohjaus leviää koskemaan useampaa prosessointilaitetta tai -konetta (kuva 12). Ryhmän palvelut aktivoidaan lähettämällä

työpyyntö (JobRequest) ControlExecutionManagerille, jonka liityntä on erikoistettu JobSupervisor -liityntä.

Spesifikaatio olettaa, että OMG:n Trading-palvelu on käytettävissä. PluginExecutor julkistaa palvelun ja sitä vastaavan PlugInExecutor -komponentin, joka valmistelee suoritussympäristön plug-in-komponentin vaatimaan muotoon ja kontrolloi plug-in-komponentin suoritusta. Tämä mahdollistaa uusien komponenttien liittämisen olemassa olevaan järjestelmään.

ControlManagement ja PluginManagement -komponentit huolehtivat dokumenttien hallinnasta ja versioiduista olioista. ControlManagement -komponentti sisältää ja hallitsee oliot, jotka kuvaavat ja ohjaavat prosessin kulkua. Operaatiokuvaukset on kapseloitu ControlStrategy -olioihin, joiden määrittelemiä tietoja RunningStrategy käyttää jäljittääkseen ControlExecutor -komponentin toimintaa.



Kuva 12. Laajennetun prosessinohjauksen komponentit.

Data Collection Plan Management -komponentti tarjoaa liittynät, joiden avulla määritellään, mitä tietoa työkalut tai anturit tuottavat historiatietoja varten. Määrittelyssä kuvataan tuotettavat muuttujat, mittaustaajuus, aikaväli, jolloin näytteet kerätään ja jolloin mitatuista suureista tuotetaan raportit niitä tarvitseville. Tämä komponentti

voidaan toteuttaa myös erillään muusta laajennetun prosessinohjausryhmän komponenteista.

ControlDatabase -komponentti tarjoaa nopean ja keveän pysyväistallennuksen ohjaustiedolle, joka täytyy jakaa usean eri prosessin kesken, jotta myötäkylä- ja takaisinkykyohjaus sekä virheiden havainnointi on mahdollista. ControlDatabase -komponenttiin tehtävät muutokset pyritään tekemään osana SEMIn standardoimistyötä ja tarkoituksena on hyödyntää HistoryManagement -ehdotusta komponentin liityntöjen määrittelyssä.

Viimeiset kaksi ryhmää muodostuvat ajastustoiminnoista (Scheduler, Dispatcher) ja henkilöresurssien määrittelystä, ohjauksesta ja seurannasta, jotka tuotannonohjauksen joustavuuden kannalta merkitsevät vähemmän ja jäävät siksi sivuutetuiksi tässä raportissa.

5. Yhteenveto

CORBA-pohjainen CIM-kehysrakenne tuo käyttökelpoisia ratkaisumalleja useimpiin tämän hetken käytännön ongelmiin. Se määrittelee yhtenäisen arkkitehtuurimallin ja rajapintakuvausten esitystavan eli IDL:n. CORBA-arkkitehtuuriin perehtyminen vaatii kuitenkin vähintään puolen vuoden opiskelupanoksen, ennen kuin voidaan täysin ymmärtää CIM-kehysrakenteen tarjoamat ratkaisut järjestelmän rakentamisessa. Lisäksi kehittäjiltä edellytetään oliotekniikan ja sovellusalueen hallintaa.

CIM-kehysrakenteen soveltamista tulisi tapahtua askeleittain, esimerkiksi kehittämällä toimiviin ratkaisuihin kehysrakenteen mukainen liityntäteknikka. Käytännössä tämä tarkoittaa myös ohjelmistoarkkitehtuurin uudelleensuunnittelua. Vaihtoehtoinen arkkitehtuurin ratkaisumalli tuotannonohjausjärjestelmän ongelmien ratkaisemiseksi on Solutions Architecture, jossa arkkitehtuuria tarkastellaan neljästä eri näkökulmasta: liike-elämä-, järjestelmä-, teknologia- ja tuotetasosta. Tämä malli ei kuitenkaan tuo yhtenäistä ratkaisua rajapintaongelmiin, mutta se palvelee ehkä paremmin eri intressiryhmien - tilaajan, kehittäjän, loppukäyttäjän - näkökulmia, kuin pelkästään teknisen näkökulman tarjoava CIM-kehysrakenne. Komponenttien ryhmittely toiminnallisuuden perusteella helpottaa kuitenkin kehysrakenteen ymmärtämistä, koska sillä kehittäjä voi sitoa uuden suunnittelutavan vanhaan tietoonsa eli perinteiseen toiminnalliseen arkkitehtuuriin.

Kehysrakenteen spesifikaation kuvauskielinä käytettiin useita käytössä olevia kuvaustapoja. Komponenttien välisten suhteiden kuvaamiseen oli kehitetty oma kuvaustapansa, Component Relationship Model, joka suuressa määrin muistuttaa UML:n käyttötapauskuvausta. Komponenttien välistä vuorovaikutusta kuvattiin viestisekvenssikaaviolla (MSC). Komponentin sisäinen informaatiomalli kuvattiin OMT-menetelmällä. Dynaamisen käyttäytymisen esittämiseen käytettiin Harelin tilakone-esitystä sekä kahta taulukkoesitystä, jossa esitettiin olion tila ja siihen kohdistuvat kyselyt sekä olion tilasiirtymät tilasta toiseen ja siirtymän aktivoimat liipaisimet. Käytetty menetelmä ei ole sinänsä tärkeä, vaan kuinka se soveltuu ongelmien kuvaamiseen. Esimerkiksi UML-kuvaustapa on käyttökelpoinen, koska kaikki tarvittavat kuvaukset löytyvät sen notaatiosta.

IDL-kuvaukset poistavat ohjelmistokomponenttien yhteensopivuusongelmat tai ainakin vähentävät niitä, koska rajapinnat kuvataan arkkitehtuuritasolla ja samoja kuvauksia käytetään varsinaisen ohjelmiston toteuttamisessa. Siten IDL toimii linkkinä suunnittelun ja toteutuksen välillä. Tämä ominaisuus palvelee erityisesti integraattoreita, joiden suurimpana ongelmana on ohjelmistojen ja järjestelmien yhteensovittaminen. Yhdessä kaupallisen ORB-tuotteen kanssa tämä ominaisuus tukee myös ohjelmistojen

siirrettävyyttä ja laajennettavuutta, koska tällöin ORB tarjoaa läpinäkyvän rajapinnan sovelluksille.

Kehysrakenteeseen määritellyt plug-in-toiminnot tukevat järjestelmän inkrementaalista kehittämistä. Tämä mahdollistaa, että toiminnassa olevaan järjestelmään voidaan kehittää uusia palveluja ilman häiriöitä tai vähäisin häiriöin, mikä vuorostaan lyhentää järjestelmän käyttöönotto- ja kehittämisäikää. Kehysrakenne ei kuitenkaan tuo ratkaisua inkrementaalisisä kehittämisessä tarvittaviin testaus- ja verifiointitekniikoihin, joihin saattaa löytyä ratkaisumalleja IT-CIM -projektin tulosten valmistuttua.

Ohjelmiston komponentointi ja rajapintamäärittelyt auttavat myös muutostilanteissa, koska muutosten aiheuttamat ongelmat pystytään paremmin ennakoimaan ja rajaamaan. Koska tuotannonohjauksessa tapahtuvat poikkeustilanteet harvoin voidaan ennakoida, järjestelmän tulee olla myös muunneltavissa mahdollisimman pienellä työmäärällä. Tähän ongelmaan tuo osittaisen ratkaisun uudenlainen, asteittainen kehittämistapa. Mitä varhaisemmassa vaiheessa loppukäyttäjät osallistuvat järjestelmän testaukseen, sitä todennäköisempää on, että myös eri poikkeustilanteita esiintyy ja nämä tilanteet voidaan ottaa huomioon seuraavassa suunnitteluvaiheessa.

CIM-kehysrakenne ei tarjoa suoranaista tukea käyttöliittymien konfiguroitavuuteen käyttäjän työn tai ominaisuuksien mukaan. Siihen voitaisiin soveltaa laajennetun prosessinohjauksen komponenteissa käytettyä periaatetta, jossa toiminnot kuvataan erillisillä strategiakuvauksilla, joita varsinainen suoritin käyttää halutun tiedon esitystavan tai käyttölogiikan toteuttamiseksi. Tämän tyyppinen ratkaisu vaatisi oman jatkotutkimuksensa ja ympäristön, jossa myös muu osa järjestelmää perustuisi kaupallisten CORBA-tuotteiden käyttöön ja CIM-kehysrakenteen toteutukseen.

Vaikka CIM-kehysrakenteen omaksuminen ja hyödyntäminen kaupallisissa tuotteissa vie varovaisestikin arvioiden muutamana vuoden, se tarjoaa epäilemättä kaikista laajimman tuen integroitavien järjestelmien tuotekehittämiselle ja ylläpidolle. Samalla se yhtenäistää eri tukiohjelmistojen toteuttamisperiaatteet ja tarjoaa yhtenäisen ajattelutavan ohjausjärjestelmien ja informaatiojärjestelmien kehittämiseksi. Mikäli kehysrakenne sisällytetään osaksi OMG:n sovellusaluekohtaisia standardeja, siitä tulee todennäköisesti ratkaisumalli, jolla on takanaan laajin yritysconsortio, tällä hetkellä noin 700 yritystä. Toisaalta Microsoftin DCOM taistelee markkinaosuuksista informaatiojärjestelmien alueella ja on siirtymässä myös ohjausjärjestelmien alueelle OPC:n (OLE for Process Control) avulla [16]. Tämä ei kuitenkaan välttämättä ole ylipääsemätön ongelma, koska jo nyt useat kaupalliset CORBA-toteutukset tarjoavat sillan Microsoftin ActiveX -komponentteihin.

Lähteet

- [1] Falk, H. 1996. EPRI Sponsored Benchmark of FMS and MMS. Preliminary Test Metrologies, Topologies and Documentation.
- [2] Falk, H. & Robbins, J. 1995. An Explanation of the Architecture of the MMS Standard. 18 p.
- [3] Sisco (Systems Integration Specialists Company Inc). MMS-Ease.
<http://www.siconet.com>.
- [4] UCA Working Group. 1996. IEC 870-6-503: TASE.2 Services and Protocols. v. 6.1. 123 p. IEC 870-6-802. TASE.2 Object Models. v. 6.1 69 p. IEC 879-6-702. TASE.2 Profiles. V. 6.1. 24 s.
- [5] EPRI (Electric Power Research Institute). 1997. Draft Input for the Utility Communications Architecture v. 2.0. Introduction + Profiles. 16 + 138 s.
- [6] Kung, A. & Lake, T. 1996. Rationale for an Open Software Reference Model for Embedded Systems. OSAF Architecture Forum. 8 s.
- [7] Bancroft, N., Seip, H., & Sprengel, A. 1997. Implementing SAP R/3. 2nd edition. Greenwich: Manning Publications Co. 310 s. ISBN 0-13-889213-X.
- [8] Doscher, D. & Hodges, R. 1997. SEMATECH's experiences with the CIM framework. Communications of the ACM. Vol. 40, No. 10. s. 82-84.
- [9] SEMATECH. 1997. CIM framework architecture guide. v. 1.0. 66 s.
- [10] SEMATECH. 1997. CIM framework specification 2.0. 396 s.
- [11] OMG. 1997. Common Object Request Broker: Architecture and Specifications, Version 2.1.
- [12] Siegel, J. 1996. CORBA: Fundamentals and Programming. New York: John Wiley & Sons, Inc., 693 s.
- [13] Baker, S. 1997. CORBA Distributed Objects Using Orbix. Addison Wesley, 518 s.
- [14] Grimes, R. 1997. Professional DCOM Programming. Birmingham: Wrox Press Ltd. ISBN 1-861000-6-X.
- [15] Pöyry, A. 1998. Tuotannon kokonaisohjaus liittyen yrityksen informaatiojärjestelmiin. Elektroniikkatehdas 2005-seminaari 28.11.1997. Oulun yliopisto, VTT Elektroniikka, VTT Automaatio.
- [16] Santori, M. 1997. OPC: OLE for Process Control. Real-time Magazine nro 4, s. 78-81.