

KYMENLAAKSON AMMATTIKORKEAKOULU

Automaatiotekniikka

Markku Nurminen

AUTOMAATIOJÄRJESTELMÄN OHJELMISTON ELINKAAREN HALLINTA

Opinnäytetyö 2013

TIIVISTELMÄ

KYMENLAAKSON AMMATTIKORKEAKOULU

Automaatiotekniikan koulutusohjelma

NURMINEN, MARKKU	Automaatiojärjestelmän ohjelmiston elinkaaren hallinta
Insinöörityö	48 sivua + 5 liitesivua
Työn ohjaaja	Lehtori Vesa Kankkunen
Toimeksiantaja	Kymenlaakson ammattikorkeakoulu
Elokuu 2013	
Avainsanat	MetsoDNA, ETHERNET, Windows, tietoverkot

Moderni automaatiojärjestelmä on dynaaminen sovellusverkko, joka on suunniteltu prosessiteollisuuden tarpeisiin. MetsoDNA-järjestelmä on tapa ohjata prosesseja ja järjestelmä perustuu tietämykseen ja informaation verkottamiseen, ohjausautomaatiikkaan ja erilaisiin sulautettuihin kenttäohjauksiin. MetsoDNA on verkko, jossa sekä ohjelmisto- että laitesovellukset toimivat yhdessä ja joka tuo joustavuutta valita tarvittavat automaatio- ja informaationhallintasovellukset tuotantolaitoksissa.

Automaatiojärjestelmän elinkaaren hallintaan liittyy niin laite- kuin ohjelmistojen hallintaa. Tähän liittyy moninaisia ongelmia automaatiojärjestelmän kunnossapidon ja järjestelmän päivittämistarpeen ollessa ajankohtaista. Automaatiojärjestelmän elinkaarta tarkastellaan niin suunnittelijan kuin ylläpitäjän suunnasta ja pohditaan mitä voitaisiin tehdä. Yhteistyö on tärkeää, koska järjestelmän rakentamiseen liittyy eri alan ihmisiä ja heidän omaa erikoisosaamistaan. Tavoitteena on havainnollistaa näiden asioiden monimutkaisuutta ja niihin liittyviä vaikeuksia, ja sitä kautta löytää ratkaisuja tulevaisuudessa eteen tuleviin ongelmiin ja antaa kokonaiskuva elinkaaren hallinnasta. Lähteinä käytetään alan kirjallisuutta, Metso Automation Oy:n materiaalia ja Kymenlaakson ammattikorkeakoulun automaatiolaboratorion käsikirjoja. Lopputuloksena todetaan yhteistyön tärkeys, suunnittelun ratkaiseva osuus kokonaisuudessa ja asiakas – toimittaja – ylläpitäjä-yhteys.

ABSTRACT

KYMENLAAKSON AMMATTIKORKEAKOULU

University of Applied Sciences

Automation Engineering

NURMINEN, MARKKU

Life Cycle and Management of Automation System
Software

Bachelor's Thesis

48 pages + 5 pages of appendices

Supervisor

Vesa Kankkunen, Senior Lecturer

Commissioned by

Kymenlaakson ammattikorkeakoulu University of Applied
Sciences

August 2013

Keywords

MetsoDNA, ETHERNET, Windows, information networks

A modern automation system is a dynamic application network that is designed for needs of process industries. One way to control processes is established by MetsoDNA system, which is based on networking of knowledge and information as well on control automation and a variety of embedded field control applications. MetsoDNA is a network in which both the software and hardware applications work together, bringing flexibility into the choice of the necessary automation and information management applications in production plants.

The life cycle of an automation system involves both hardware and software management. This entails a multitude of problems that surface when the automation system requires maintenance and updates. The life-cycle of an automation is examined from both the designer's as the maintenance personnel's perspective with the purpose of finding solutions to these problems. Cooperation is importance, because the system construction involves people from various fields of expertise. The aim of this paper is to illustrate the complexity and related difficulties but also to find solutions to future problems and provide an overview of the life cycle in order to facilitate the management of it.

Sources are the literature of the industry, material of Metso Automation Oy as well as manuals in the material automation laboratory of Kymenlaakso UAS. The result emphasizes the importance of co-operation the decisive role of design and the customer-supplier-administrator connection.

SISÄLLYS

ABSTRACT

KÄSITTEITÄ	6
1. JOHDANTO	8
2. METSODNA	8
2.1. Historia	8
2.2. MetsoDNA nyt	10
3. ETHERNET	10
3.1. Ethernetin historia	10
3.2. Ethernetin periaate	11
3.3. Ethernetin toiminta	12
4. KÄYTTÖJÄRJESTELMÄ	13
4.1. Verkkokäyttöjärjestelmä	13
4.2. Yhteensopivuus	14
4.3. Tietokonejärjestelmä osana kokonaisjärjestelmää	14
4.4. Rinnakkaisuus	15
4.5. Eri järjestelmien yhdistäminen	15
4.6. Käyttöjärjestelmän rakenne	16
4.7. OSI-malli	16
5. PC AUTOMAATIOSSA	18
5.1. Automaatiojärjestelmän yleisrakenne	19
5.2. MetsoDNA-automaatiojärjestelmän rakenne	21
5.3. Aktiviteetit	22
5.4. Verkonrakenne MetsoDNA:ssa yleisesti	23
5.5. EA suunnittelu- ja ylläpitoympäristö	24
5.5.1. Function Explorer	26
5.5.2. Käyttöliittymä	27

6. ELINKAARI KÄYTTÖJÄRJESTELMISSÄ JA OHJELMISTOISSA	28
6.1. AutoCAD-ohjelmiston ja käyttöjärjestelmien elinkaari	31
6.1.1. Microsoftin tarjoama tuki eri käyttöjärjestelmille	31
6.1.2. MetsoDNA:n ja Microsoft Windows-järjestelmien historia	32
6.1.3. Windows-alustat, joita tuetaan MetsoDNA:ssa	32
6.1.4. Kokonaisjärjestelmän päivittäminen	33
6.2. Ohjelmistotekniikka	34
6.2.1. Ohjelmiston elinkaari	35
6.2.2. Elinkaarikustannukset	36
6.2.3. Tuotteenhallinta	36
6.2.4. Ohjelmointiin liittyvät erityisongelmat	37
7. YLLÄPIDON ONGELMAT	37
7.1. Järjestelmän siirto ylläpitohenkilöstölle	39
7.2. Järjestelmien erilaiset toteutukset	39
7.3. Ylläpidon kehittäminen	40
7.3.1. Sovellukset	40
7.3.2. Vanhat järjestelmät	41
7.4. Automaatio- ja toimistojärjestelmien eroja	41
7.4.1. Koventaminen	42
8. PÄÄTELMÄ	43
LÄHTEET	45
LIITTEET	
LIITE 1. Metson AutoCAD - ja käyttöjärjestelmätuki	
LIITE 2. Metson automaatiojärjestelmätekniikan elinkaari	
LIITE 3. Token Ring fyysinen topologia	
LIITE 4. Automaatiojärjestelmän kehitys Metso	

KÄSITTEITÄ

EAS - Engineering and Maintenance Activity Server, suunnittelupalvelin, jossa kaikki järjestelmän konfiguroinnit.

EAC – Engineering and Maintenance Activity Client, suunnittelutyökalu, jolla tehdään sovellusohjelmointi ja käyttöliittymäsuunnittelu.

ETHERNET – Yleisin, laajasti hyväksytty lähiverkkotekniikka.

CSMA/CD – Carrier Sence Multiple Access With Collision Detection, tietoliikenteen siirtotien varausmenetelmään perustuva, jonka tehtävänä on jakaa siirtotietä usealle tietokoneelle.

VME-SOLMU – Plug in pistoyksikkö, joka koostuu erilaisista keskusyksiköistä (CPU), verkkoliityntäkorteista (NCU, NCU2), kenttäväyläohjaimista (FBC), muistiyksiköistä (DMU).

AIDB – AutoInfoDataBase, sisältää suunnitteluun tarvittavat oliot.

OPS – Operator Station, operointiasema.

PCS – Process Control Station, prosessiasema.

ALP – Alarm Processor, hälytysasema.

DIA – Diagnostic Station, diagnostiikka-asema.

RTS – Router Station, reititysasema.

BU – Backup Station, varmennusasema.

TOKEN RING – Valtuudenvälitysverkko, jossa oikeus lähettää dataa verkossa, hallitaan valtuutusta kierrättämällä. Verkko on fyysisesti tähtimäinen vaikka rengas loogisesti.

1 JOHDANTO

Tämä opinnäytetyödokumentti tutustuttaa MetsoDNA-automaatiojärjestelmän toimintaan ja rakenteeseen ja toteutuvassa modernisoinnissa oleviin suunnittelun ja ylläpidon haasteisiin. Modernisointi sisältää erilaisia laitteistoja ja ohjelmistoja, joiden kanssa esiintyy yhteensopivuusongelmia. Nämä ongelmat ovat kiusallisia järjestelmän toimittajalle ja asiakkaalle. Ne aiheuttavat lisätyötä ja tuottavat ongelmia. Tutkimustapoina käytetään alan kirjallisuutta, Metso Automation Oy:ltä ja Kyamk:lta saatua materiaalia. MetsoDNA:n järjestämällä kurssilla perehdyttiin automaatiojärjestelmän suunnitteluun ja on käytetty näitä tietoja hyväksi opinnäytetyössä. Alunperin tarkoituksena oli tutustua erään tehdaslaitoksen modernisoitavaan keittolaitoksen automaatiojärjestelmän ohjelmistoihin ja osallistua samalla työharjoitteluun. Mutta erinäisten sattumien vuoksi tämä ei toteutunut aivan suunnitellulla tavalla. Siksi päädyn ratkaisuun, jossa tutkimuskohteita ei tutkita siinä laajuudessa, jota aiemmin suunniteltiin.

Mutta kaiken kaikkiaan saan tutustua MetsoDNA:n järjestelmään niin järjestelmätasolla kuin siihen liittyvien ohjelmistojen tasolla. Tarkoituksena on tutustua automaatiojärjestelmään kokonaisuutena ja siihen liittyviin haasteisiin elinkaaren aikana sekä eri osapuolten yhteistyön hankaluuteen ja asenteisiin. Käyn läpi myös kustannusasioita ja miten niihin päästään vaikuttamaan. Lisäksi tässä työssä puhutaan kommunikoinnista käyttäjän ja suunnittelijan/ylläpitäjän välillä. Ongelmat ilmenevät kommunikoinnissa asiantuntijan ja käyttäjän välillä aiheuttaen väärinkäsityksiä.

2 METSODNA

2.1. Historia

Automaatiossa on kyse jonkin fyysisen tapahtuman mittaamisesta ja mittauksen perusteella tehtävästä ohjauksesta. Automaation alkuaikoina ohjattavaa prosessia ohjattiin mekaanisesti, hydraulisesti, pneumaattisesti tai elektronisesti. 1950-luvulla alkoi tietokoneen käyttö prosessien ohjauksessa. Tietokoneohjaus sai automaatiossa kehityksen nopeaan nousuun. Tästä olemme tulleet nykyisiin hajautettuihin digitaalisiin automaa-

tiojärjestelmiin. (Ylikunnari 2003.)

Kehityksen vaiheita:

- pioneeriaika noin 1955
 - suora digitaalinen ohjaus noin 1962
 - minitietokoneet noin 1967
 - mikrotietokoneet noin 1972
 - digitaalisen ohjauksen yleistyminen noin 1980
 - hajautettu ohjaus noin 1990
- automaation kehitys jatkui tehtaiden integroitua, kun järjestelmäkarttaa katsotaan vaakasuoraan koneet, laitteet ja prosessit yhdistyivät toisiinsa ja pystysuoraan prosessin automaatiojärjestelmien yhdistyessä muihin automaatio- ja hallintojärjestelmiin lopulta kehittyäkseen yhdeksi ohjattavaksi kokonaisuudeksi

Digitaalisia automaatiojärjestelmiä eli tietokoneohjausjärjestelmää käytetään yleisesti tuotantoprosessien ohjaukseen. Prosessit ovat usein monimutkaisia, laajoja ja voivat olla jopa sisältää ihmiselle vaarallisia prosessiosuuksia esimerkiksi kemianteollisuudessa. Automatisointi on hyvä ratkaisu tällaisiin prosesseihin tuomaan turvallisuutta. Automaatio on myös ratkaisu toistuvaan niin sanottuihin liukuhihnatyyppeihin prosesseihin, koska se tuo luotettavuutta ja laatua yksitoikkoiseen työhön verrattaessa ihmiseen. Tällä tarkoitan, että kone ei väsy ja menetä tarkkaavaisuutta niin kuin ihminen väsyessään. Automaatio on vallannut muitakin osa-alueita kuten kappaletavateollisuuden.

Nykyinen hajautettu automaatiojärjestelmä DCS (Distributed Control System) on tietoverkko, jossa tietokoneet kommunikoivat ohjaus-, säätö-, mittaus- ja asetuservotiedoilla. Nämä kommunikoivat laitteet on hajautettu ohjattavan prosessin laajuudelle. Hajautettujen automaatiojärjestelmien kehittyminen alkoi Honeywell TDC 2000:sta vuonna 1976. Tässä järjestelmässä hajautus oli toiminnallisella tasolla. Eli yksittäisille tietokoneille oli jaettu omat erikoistehtävät, jotka ”keskustelivat” keskenään (Ylikunnari 2003). Honeywell TDC 2000:ta seurasi Valmet Automation ja Metso Automation Damatic Classic 1979, Damatic XD 1988, Damatic XDi 1996, MetsoDNA 2000, MetsoDNA CR 2006 ja nykyinen Metso DNA 2011. (Kosonen 2012.) (Liite 4)

2.2. MetsoDNA nyt

Nykypäivän MetsoDNA-järjestelmä kattaa sellu- ja paperiteollisuuden lisäksi voimantuotantoa (turbiinit) ja on valloittamassa uusia alueita kuten kaivos- ja maanrakennusala. MetsoDNA-järjestelmä on kehittynyt tuottavuudeltaan ja ympäristöystävällisyydeltään, mikä on nykypäivänä erittäin tärkeä markkinoinnin myyntivaltti. Kehitystyö on ollut käyttäjälähtöistä ja kehityssuunta on käyttäjää (operaattori) helpottava työn hallinta. MetsoDNA-järjestelmään on saatavissa sisäänrakennettu niin sanottu mekaaninen kunnon- ja käynninvalvonta. Mekaaninen kunnonvalvonta tarkoittaa ennakoivaa huollontarpeen valvontaa mekaanisille osille, kuten laakerit ja muut sellaiset ja tarvittava tieto on saatavilla helposti prosessinohjauksesta. MetsoDNA hallitsee sellu-, paperi- ja massateollisuutta kokonaisuudessaan, tehtaan puukentiltä aina pituusleikkurille saakka, josta valmis tuote saadaan. Laatu paranee ja tuottavuus tehostuu nykypäivän MetsoDNA:n ohjausjärjestelmän liittämällä prosessinohjaukseen. Käyttäjä saa prosessista otettua tietoa helposti napinpainalluksella reaaliajassa tai takautuvasti eli tapahtunutta historiaa. (Paperi ja puu 14.5.2012.)

3. ETHERNET

3.1. Ethernetin historia

Ideana Ethernetin syntymiselle on ollut jaettu siirtotie. Projekti ALOHA, eli radioverkko, jota Norman Abramson ja hänen työryhmänsä kehittivät 1960-luvulla, oli Ethernetin kaltaisen verkon ensiaskel, idea jaetusta siirtotiestä. Järjestelmässä oli kaksi kanavaa, lähtevälle ja tuleville liikenteelle. ALOHA:ssa niin sanotulta pääkoneelta oli lähetyskanavalla suora yhteys etäpisteiden koneisiin. Lähetettävä liikenne pääkoneelta suoritettiin yhteislähetysperiaatteella. Jotta lähetys saavutti oikean osoitteen, niin lähetysten otsaketietoihin oli laitettu kohdeosoite, johon lähetys haluttiin toimittaa. Näin myös vastaanottoasema tiesi lähetysten kuuluvan sinne. Etäpisteistä pääkoneelle tulevat lähetykset käyttivät hieman erikoisempaa tapaa estää lähetysten katoaminen, satunnaista uudelleenlähetystä, koska lähettäjiä voi olla useampia kuin yksi samaan aikaan pääkoneen suuntaan. Käyttäjän painaessa enter-näppäintä etäasema hylkäsi sa-

noman. Tämän jälkeen lähettävä asema jäi odottamaan lähetyksen kuittausta, joka piti tapahtua tietyn ajan sisällä (200 – 1500ns). Jos näin ei tapahtunut, niin sanoma tulkittiin vioittuneeksi eli jokin muu asema oli lähettänyt samanaikaisesti ja oli tapahtunut törmäys. Tästä seurasi vioittuneen sanoman todenneiden asemien siirtyminen satunnaiseen uudelleenlähetytilaan. Satunnaisesti ne lähettivät uudestaan, kunnes sanoma saavutti määränpään. Tämän tyyppistä verkko, jossa lähettävät asemat kilpailevat lähetyvuoroista samalla kanavalla kutsutaan kilpavarausverkoksi. Ensimmäinen Ethernet sai alkunsa 1972 heinäkuussa. Tuohon aikaan Xerox kaappasi herran nimeltä Robert Metcalf PARC:lle verkkotekniseksi asiantuntijaksi Xerox PARC:n tutkimuskeskukseen. Hänen haluttiin liittää Xeroxin ALTO-tietokone ARPANET-verkkoon, joka oli Internetin edeltäjä. Metcalf tutki Abramsonin aikaisempia tutkimustuloksia ALOHA-järjestelmän parissa. ALTO ALOHA -verkosta tuli maailman ensimmäinen lähiverkko. Se oli toteutettu mikroilla ja toimi 22.7.1973. Tuolloin ensimmäisen kerran sana Ethernet ilmestyi verkon nimenä. (Jaakohuhta 2000, 9 - 12.)

Ethernetin merkittävin parannus aikaisempaan ALOHA-verkkoon oli törmäyksentunnistus- eli carrier sense -ominaisuus. Lähettävä asema kuuntelee, onko siirtotie vapaa ennenkuin lähettää datan. 1977 Metcalf työryhmineen sai patentin ”Multipoint Data Communication System with Collision Detection”. Tästä hetkestä alkaen monilähetystekniikka on tunnettu ”Carrier Sense Multiple Access with Collision Detection- CSMA/CD” eli Ethernet syntyi. (Jaakohuhta 2000, 12 - 13.) Ethernet 1 oli alkujaan 1970-luvun DIX - projektista, jonka muodostivat Dec, Intel ja Xerox. Ethernet 2, julkaistiin 1983 ja IEEE standardoi sen 1985 (ISO 8802-3). (Pahlama 2009.) Tästä eteenpäin Ethernetin kehitys on ollut nousujohtoista ja on nykyään yleisin, laajasti hyväksytty lähiverkkotekniikka.

3.2. Ethernetin toimintaperiaate

MAC eli Media Access Control -protokollaa käytetään Ethernetissä. Mainittakoon Token Ring -verkko (Liite 3), koska sitä on käytetty automaatiassa ja siinä toimintamenetelmänä on TOKEN -protokolla. Verkot sisältävät omat tiedonvälitysmenetelmänsä, miten dataa käsitellään jaetussa mediassa. Ethernet on se toimintamenetelmä, jota käsitellään tässä työssä. (Jaakohuhta 2000, 85.)

Sanomia Ethernet-verkossa kutsutaan kehyksiksi. Näitä kehyksiä on kolmea tyyppiä.

Unicast-kehys, jolla on yksilöllinen lähde- ja kohdeosoite. Unicast-kehykset, datapaketit kulkevat vain ja ainoastaan lähettäjän ja vastaanottajan välillä.

Multicast-kehys, jossa lähettäjä lähettää datapaketin monelle vastaanottajalle esimerkiksi tietylle ryhmälle ja kehystyyppiä käytetään työryhmäohjelmissa. Sanoman lähettäminen on huomattavasti tehokkaampaa kuin yksitellen lähetettävä sanoma.

Broadcast-kehys, jossa lähettäjän lähettämä datapaketti leviää kaikille laitteille koko levitysalueella ja Broadcast-viestejä on eri tarkoituksiin.

Spanning tree -menettely, jossa lähiverkon hierarkiasta kehyks tarvitaan levittämään tietoa kaikille silloille ja kytkimille ja menetelmää käytetään verkkokäyttöjärjestelmissä olemassa olon mainostamiseen. Palvelimet ja työasemat käyttävät sitä levitysviesteissään ja verkkoprotokollat lähettäessään tietoa osoitteistaan, reitittimisestä ja palvelimistaan. Sillä reititystietojen välitykseen tarvittavalla tiedolla pystytään selvittämään paras mahdollinen reitti IP-verkossa tai yhdistämään IP- ja MAC-osoite. (Jaakohuhta 2000, 85-87.) MAC-osoite ei ole hierarkkinen, joten se on hankalampi isoissa verkkoratkaisuissa (hidas reitittää). IP-osoite on hierarkkinen, koska siinä on verkko-osa ja laite-osa erikseen (nopea reitittää). (Pahlama 2009.)

3.3. Ethernetin toiminta

Ethernet-lähiverkkoa on väylä- ja tähtityyppistä ja kummassakin tyyppissä päätelaitteet ovat kytketty eri periaatteella toistimilla väylälle, joista muodostuu levitysviestialue. Ethernet-verkko perustuu CSMA/CD-menettelyyn. Laite, joka aikoo lähettää dataa, kuuntelee ensin, että kanava on vapaa ennen kuin alkaa lähettää sanomaa (CSMA). Jos kuitenkin kaksi laitetta lähettää samaan aikaan dataa tapahtuu törmäys (collision). Törmäys havaitaan ja arvotaan uudelleen lähetysjärjestys (CD). Uudelleen lähetykset lisäävät muun liikenteen ohella väylän liikennekuormaa, mutta ruuhkan keventämiseen on kyllä keinoja. Ensinnäkin, nopeuden nosto väylässä auttaa, koska tietoa (dataa) siirtyy lyhyemmässä ajassa enemmän kuin hitaammalla nopeudella. Nopeuden kasvattaminen Ethernetissä on mahdollista kolmelle eri nopeudelle: 10-, 100- ja 1000 -

Mbps. Lähiverkkokytkin mahdollistaa törmäysalueiden jakamisen helposti ja sillä saadaan jokaista porttia kohden jaettua tietty nimellishopeus. Toinen tapa liikennekuorman keventämiseksi väylällä on eristäminen, millä samalla väylällä olevia asemia jaetaan pienempiin ryhmiin. Lähiverkon toiminta on edelleen samanlainen, vaikka se jaetaan pienempiin törmäysalueisiin ja siltaus on menetelmä, jota käytetään jakamaan lähiverkko eri osa-alueisiin eli törmäysalueita pienentämään. (Jaakohuhta 2000, 91-94.)

Mainitaan myös CSMA/CA (Carrier Sence Multiple Access with Collision Avoidance), jonka lähetystavassa lähetysaikeiset laitteet varoittavat levitysviestillä muita laitteita lähetysaikeestaan. Sillä pyritään välttämään dataviestien törmäykset mutta se hidastaa verkon toimintaa. Eli Ethernet tai lähiverkko koostuvat väylistä, työasemista, kytkimistä, reitittimistä, palvelimista, silloista ja muista sellaisista standardoiduista laitteista. (Pahlama 2009.)

4. KÄYTTÖJÄRJESTELMÄ

Mikä käyttöjärjestelmä on? Vastaus tähän kysymykseen on löytyy helpommin kuin kysymykseen, mikä käyttöjärjestelmän tehtävä on. Käyttöjärjestelmä (operating system) tarkoittaa ohjelmistoa, joka on liityntä sovellusohjelmiin ja laitteisiin ja se on rajapinta käyttäjän ja laitteiden välillä, jossa ne kommunikoivat keskenään. Eri ohjelmia suoritetaan usein samaan aikaan ja käyttöjärjestelmä huolehtii niin laiteresurssien jakamisesta kuin yleisesti toiminnan sujuvuuden ylläpitämisestä. Käyttöjärjestelmiä on esimerkiksi Unix- ja Linux-verkkokäyttöjärjestelmät ja työasemien omat Windows-, Linux- ja AppleOS-käyttöjärjestelmät. (Haikala & Järvinen 2003, 11-12.) Työasemien käyttöjärjestelmät ovat yhden käyttäjän laitteita, jotka ovat rajapintana käyttäjän ja järjestelmän välillä.

4.1. Verkkokäyttöjärjestelmä

Tietoliikenteen toimivuuteen tarvitaan aina ohjelmistoja, jotta tietoliikennettä yleensä voi tapahtua. Ohjelmistojen tarkoitus on asettaa tietoliikenteelle niin sanotut liikennesäännöt ja mainituista ohjelmistoista käytetään nimitystä verkkokäyttöjärjestelmä (network operating system). Verkkokäyttöjärjestelmä luo verkkoyhteydet työasemaverkkoon ja sen tarkoitus on antaa yhteisiä palveluja verkossa eri työasemille. Kom-

munikointi työasemien välillä mahdollistuu ja palveluina ovat esimerkiksi tulostimien jakaminen useammalle työasemalle, yhteisien tietokantojen käyttö, sähköpostit ja kaikkea muuta sellaista, mitä huomaamattamme käytämme melkein joka päivä. Esimerkkeinä verkkokäyttöjärjestelmistä mainittakoon Novell NetWare, Microsoft Windows NT (nykyisin Windows 2000), UNIX ja Open VMS. (Jaakohuhta 2000, 5.)

4.2. Yhteensopivuus

Käyttöjärjestelmät asettavat tietynlaisia vaatimuksia laitteistolle, joissa se operoi. Ei riitä, että on käyttöjärjestelmä ja laitteisto, vaan niiden täytyy yhteensopia keskenään. Usein yhteensopivuusongelmia kohdataan, kun uusia versioita tulee ja vanhat versiot toimivat rinnalla. Uusia päivityksiä tarvitaan, jotta uusia ominaisuuksia voidaan ottaa käyttöön ja niitä tarvitaan korjaamaan aiempien versioiden esiintyviä puutteita tai soveltamaan vanhojen järjestelmien yhteensopivuutta uusien versioiden ominaisuuksien käyttämisen mahdollistamiseksi. Esimerkiksi Windows 2000 -käyttöjärjestelmissä on käytetty tekniikkaa, jolla pyritään parantamaan käyttöjärjestelmän siirrettävyyttä laitteistoalustojen välillä ja samalla pyritään minimoimaan laitteistojen muutoksista johtuvia ongelmia. Tämä toteutetaan ohjelmallisesti, joka toimii käyttöjärjestelmän ja laitteiston välissä osiossa, HAL (Hardware Abstraction Layer) eli laitteistoabstraktiokerros. (Haikala & Järvinen 2003, 12, 14.) Jokainen laite tarvitsee ajurin (driver) eli sovelluksen, joka koostuu tietorakenteista (laiteohjaus), rutiineista (alustaminen) eli laitteelta luku-, kirjoittaminen- ja keskeytyskäsitteilytoimintarutiini. Ajurin perusidea on sovittaa laite muuhun laitteistoon ja toiminnan ongelmattoman toiminnan varmistamiseksi. (Haikala & Järvinen 2003, 81.)

4.3. Tietokonejärjestelmä osana kokonaisjärjestelmää

Tietokonejärjestelmää on joka päivässä elämässä edessämme, esimerkkeinä voidaan mainita auton ABS-jarrujärjestelmä, teollisuudessa prosessinohjausjärjestelmät ja kassapäätejärjestelmät. Tietokonejärjestelmä on niin sanotusti sulautettu järjestelmään (embedded system) eli on osana suurempaa kokonaisuutta. Niissä reaaliaikainen ohjelmisto saa useita ärsykeitä eri lähteistä ja epämääräisessä järjestyksessä, mutta siitä huolimatta ohjelman pitää pystyä suorittamaan siltä vaaditut tehtävät. Tietokonetekniikka valtaa jatkuvasti enemmän ja enemmän aloja, joihin liittyy ohjausta, hallintaa ja

tietorekisterin ylläpitämistä. (Haikala & Järvinen 2003, 82.)

4.4. Rinnakkaisuus

Ohjelmissa rinnakkain tapahtuvat toiminnot kommunikoivat keskenään yhteisiä tietorakenteita käsiteltäessä tai keskustelevat sanomien välityksellä muusta liikenteestä. Moniajojärjestelmät toimivat kuvatulla tavalla ja rinnakkaisuus voidaan jakaa kahteen tyyppiin:

- yksittäiseen ohjelmaan, jonka sisällä suoritetaan samaan aikaan eteneviä prosesseja (multitasking)
- järjestelmään, jossa suoritetaan samaan aikaan eteneviä toisistaan riippumattomia prosesseja (multiprogramming)

Rinnakkaisohjelmien käyttöön on monia syitä. NOPEUS tarkoittaa, että useammalla suorittimella voidaan nopeuttaa ohjelmien toimintaa. HAJAUTUS eli ohjelma, jossa useammat prosessit kommunikoivat keskenään, hajautetaan tietokoneverkkoon (tiedostojärjestelmät). ONGELMA OHJELMA, jotkin ohjelmointiongelmat eivät toimi ilman rinnakkaisia prosesseja tai ainakin vaikeuttavat ohjelmointia. MODULARISOINTI tarkoittaa, että modularisoidaan ohjelma jakamalla prosesseihin ja niiden välillä sanomia vaihdetaan (kommunikointi). (Haikala & Järvinen 2003, 89-91.)

Käyttöjärjestelmä tapauksessa NOPEUS eli monisuoritinkäyttöjärjestelmä (multiprocessing), HAJAUTUS eli ohjelma verkkokäyttöjärjestelmään, ONGELMA OHJELMA ja MODULARISOINTI saatetaan moniajokäyttöjärjestelmän toteutukseen. Rinnakkaisohjelma koostuu useista prosesseista, jotka vaikuttavat toinen toisensa etenemiseen. Niin sanotusti suoraan eli yhteisien tietorakenteiden käyttö, lähettäminen, luominen ja hävittäminen, sekä epäsuorasti, resursseista kilpaileminen eli suoritin, muisti yms. (Haikala & Järvinen 2003, 89-91.)

4.5. Eri järjestelmien yhdistäminen

Lähiverkkotekniikka (Ethernet, Token ring), joka on kehittynyt ajan myötä, tuo mahdollisuuden suhteellisen helposti ja vähin kustannuksin yhdistää eri järjestelmiä. Esi-

merkkinä kirjoitinpalvelu, jossa verkon laite tarjoaa palveluita verkon muille laitteille. Samoin tärkeä tiedostojärjestelmä hajautetaan verkkoon muiden saataville ja käyttäjältä se saattaa näyttää keskitetyltä vaikka todellisuudessa se on hajautettu. Koska siirtotienä useimmiten käytetään suuritehoista tiedonsiirtokanavaa kuten edellä mainittiin (Ethernet), niin yhdistetyissä järjestelmissä se on käyttöjärjestelmäpainotteinen. Tämä siitä syystä, että pullonkaulana ei ole siirtotie vaan protokollaohjelmien vaatima käsittelyaika. (Haikala & Järvinen 2003, 218-219.)

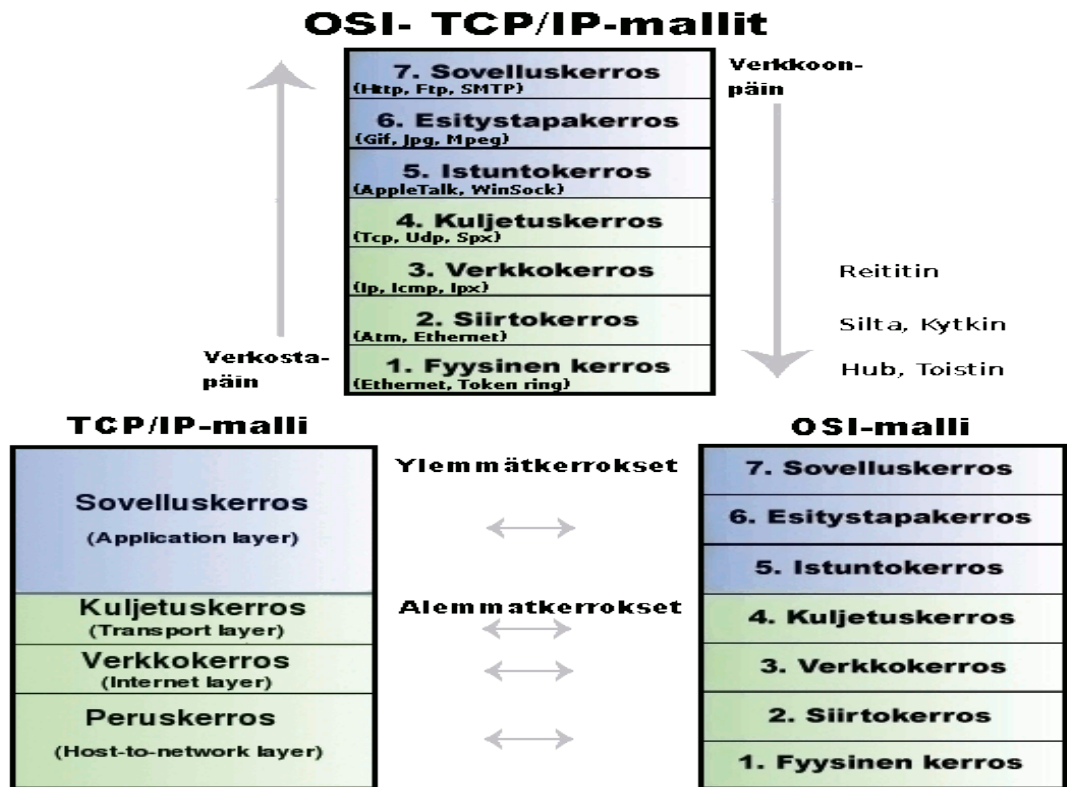
4.6. Käyttöjärjestelmän rakenne

Hajautettu tiedostojärjestelmä ei ole monimutkainen toteuttaa, koska tiedostojärjestelmän päälle rakennetaan kerros, jonka tehtävä on ohjata muissa koneissa olevien tiedostojen operaatiot oikeisiin osoitteisiin. Hajautettu käyttöjärjestelmä on hankalampi toteuttaa siksi, että tällaisessa käyttöjärjestelmässä huolehtii se automaattisesti ohjelman eri osien välisestä kommunikaatiosta ja kuorman jakamisesta. Käyttöjärjestelmä ei voi edeltä käsin tietää, miten ohjelman eri osat kommunikoivat verrattuna käyttäjään. Suunnittelun kannalta ajateltuna joudutaan paneutumaan seuraaviin seikkoihin: kommunikointitapa (eri osien kommunikointi), nimien hallinta ja suojaus (objektien kuvaus verkko-osoitteiksi), resurssien hallinta (kuorman jakaminen), turvallisuus (yhteyksien ja tietojen suojaus), palvelutaso (vikasietoisuus, suoritusteho), palvelut (kirjoitin, tiedostot). (Haikala & Järvinen 2003, 225-226.)

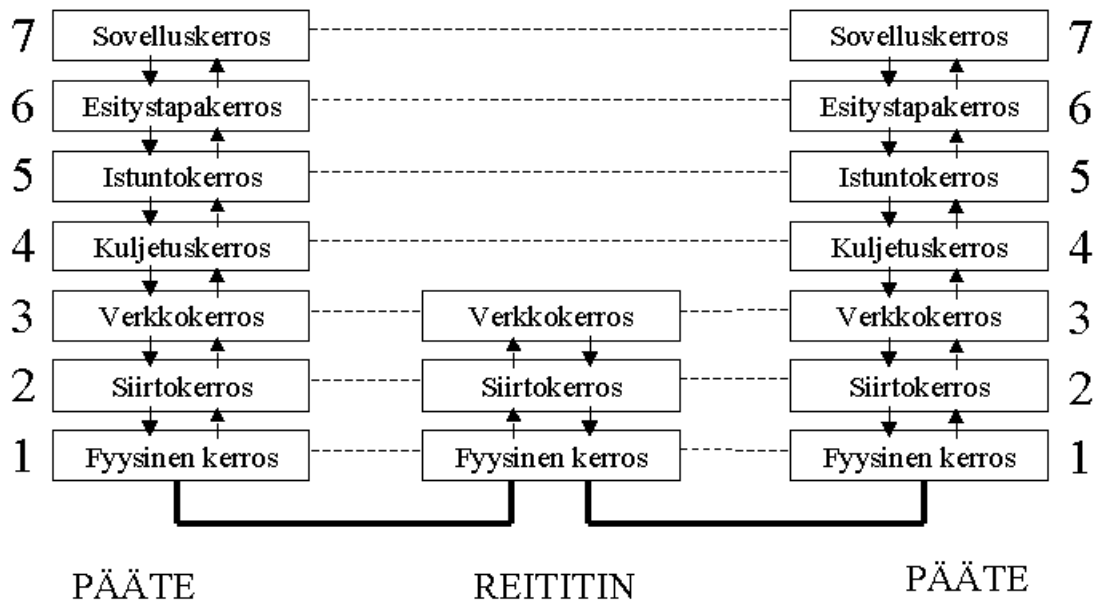
Käyttöjärjestelmien kehitystä ohjasi Microsoft-painotteisuus PC-koneissa ja www:n (World Wide Web) kehittyminen. Tästä johtuen se vakiinnutti työasemiin ja palvelimiin käyttöjärjestelmiksi Windows- ja Unix-käyttöjärjestelmät eri versioin ja www:n ideana oli mahdollistaa tiedon hakeminen verkosta helposti. (Haikala & Järvinen 2003, 234-235.)

4.7. OSI-malli

OSI-malli on 7-kerroksinen ISO standardoitu viitemalli ja hierarkkinen protokollapino, jossa aina ylempi kerros käyttää alemman palveluja (pystysuunnassa) ja protokollat (vaakasuoraan) keskenään. (Kuva 1 ja 2)



Kuva 1. OSI-malli. Kommunikointi pystysuunnassa. (krimaka.net, viitattu:14.5.2013.)



Kuva 2. Protokollat kommunikointi vaakasuoraan. (Tietoliikenneohjelmistojen ja multimedian laboratorio, viitattu 14.5.2013.)

OSI-kerrokset lyhyesti esiteltynä alemmasta ylöspäin ovat:

1. Fyysinen kerros (Physical layer) muuttaa informaation signaaleiksi, avaa ja sulkee yhteyden.
2. Siirtoyhteys kerros (Data Link layer) kehystää sanoman Ethernet-kehukseen, joka aiotaan lähettää. Siirtokerroksessa kaksi alakerrosta, LLC (Logical link control) siirtoyhteyden looginen ohjaus ja MAC (Media access control) median varaus, lähetysvuoron varaus, alempi kerros. Toteutus joko ohjelmallisesti (SW) tai elektroniikalla (HW).
3. Verkkokerros (Network layer) muodostaa päätelaitteista tietoverkon ja on IP - protokollaa käyttävä, globaalinen ja hierarkkinen. Suurissa verkoissa varsinainen- ja aliverkkokerros ja esimerkiksi reititin toimii verkkokerroksessa. Tehtävänä on datapaketin siirto.
4. Kuljetuskerros (Transport layer) vastaa tiedonsiirto-osapuolien välillä tapahtuvasta datan siirron virheettömyydestä. Esimerkkinä TCP (Transmission Control Protocol).
5. Yhteysjaksokerros (Session layer) epästabiliin verkkojen yhteyden ylläpito. Esimerkiksi istunnoissa toteutuvien keskustelujen synkronointi.
6. Esitystapakerros (Presentation layer) tiedon esitystavan muodostaminen ja muunnokset. Koodimuunnokset, pakkaus ja salaus.
7. Sovelluskerros (Application layer) sisältää sovellukset (protokollat), esimerkkeinä HTTP web-selaus, SMTP sähköposti ja FTP tiedostojen siirto. (Pahlama 2009, 1.)

5. PC AUTOMAATIOSSA

Ohjelmistotarjonta PC-pohjaisessa automaatioissa on parantunut 80 – 90-luvun vaihteesta lähtien. Dos-käyttäjärjestelmiin pohjautuvat saivat rinnalleen muita kuten Unix-, OS/2- ja Windows-pohjaiset versiot. Ne ovat siirtyneet aikaisemmasta tiedonkeräyksestä valvomo-ohjelmistoiksi ja jopa suoriin toteutuksiin eli PC ohjaa suoraan toimilaitetta. Tiedonkeräämisessä tieto kulkee prosessista PC:hen päin ja valvomotoiminnassa tietoa siirtyy myös toiseen suuntaan eli prosessiin päin. Tieto on lähinnä yksittäisiä ohjauksia ja asetusarvotietoja, koska erillisyksiköt toteuttavat säädöt ja ohjaukset. (Ylikunnari 2003, 15.)

Suurissa PC-säädöissä liityntäkorteilla (prosessi-I/O:lla) tapahtuu ohjauksen toteutus ja siihen liittyy valvomo-ohjelmisto, josta voidaan konfiguroida säätöalgoritmia ja

loogisia operaatioita. Se on halpa ratkaisu mutta sopii vain lähinnä pieniin automatisoitaviin kohteisiin. Teollisuudessa PC:t jaetaan toimisto- ja teollisuus-PC:ksi. Olenainen ero on, että teollisuus-PC:t kestävät kovempia olosuhteita kuin toimisto-PC:t ja toinen huomattava ero on, että teollisuus-PC:t ovat modulaarisia, jotta vioittuvien korttien ja kovalevyjen vaihto on mahdollista, helppoa ja nopeaa. Automaatiossa vaadittavat piirteet käyttöjärjestelmältä ovat moniajo, suojaus ja priorisointi, jotta on mahdollista ajaa monia ohjelmia yhtä aikaa. Ohjelmat eivät saa häiritä toisiaan tai käyttää samaa muistialuetta, koska tärkeysjärjestys ohjelmien kuin muistin käytön jaossa ohjelmien kesken mahdollistaa toimivuuden. (Ylikunnari, 2003, 16-17.)

Tyypillisesti ohjelmat ovat suunnittelutyökaluja, joilla suunnitellaan erilaisia kaavioita joko käskylistan muodossa tai graafisesti toteutettavalla lohko-ohjelmoinnilla. Kokonaisautomaattioratkaisu koostuu standardielektroniikkakorteista, joilla oma erikoistehävänsä (ala-asemat), nopea tietoliikenneverkko, prosessi-I/O:n elektroniikka ja valvomolaitteet. Varusohjelmisto on olennainen osa tätä ratkaisua, koska niissä on kattava ohjelmisto automaation suunnitteluun, operointiin, valvontaan ja sovellusohjelmoinnin tukemiseen ja integrointiin muihin ympäristöihin ja järjestelmiin tarvittavia työkaluja. Suomessa hallitseva järjestelmä on Valmetin Damatic Classic, XD/Xdi ja Metson DNA CR. Kenttäväylien kehityksestä johtuen automaattioratkaisut ovat yksinkertaistuneet, selkeytyneet ja standardoinnin edistymisen väyläratkaisuissa on auttanut yhteensopivuutta laitteiden osalla. Kehitys kenttäväyläteknologiassa on edes auttanut merkittävästi automaattiototeutuksia vähentäen liityntäelektroniikkaa ja kaapelointia. Älykkäät laitteet ovat saaneet enemmän tehtäviä, kuten suunnittelu-, asennus- ja ylläpitotoimintojen selkeytymisenä, joka on johtanut PC:n käyttöön yleisesti valvomoratkaisuna. (Ylikunnari 2003, 17-18.) Varusohjelmilla tarkoitetaan ohjelmiston osia, joita järjestelmätoimittaja on toimittanut vakiokokoonpanossa tai optiona ja ne on erikseen hankittava. Varusohjelmat koostuvat käyttöjärjestelmästä, tietoliikenneohjelmistosta, ajureista, liityntäohjelmista. (Ylikunnari 2003, 28-29.)

5.1. Automaatiojärjestelmän yleisrakenne

Perinteiseen automaatiokokonaisuuteen erotellaan seuraavasti:

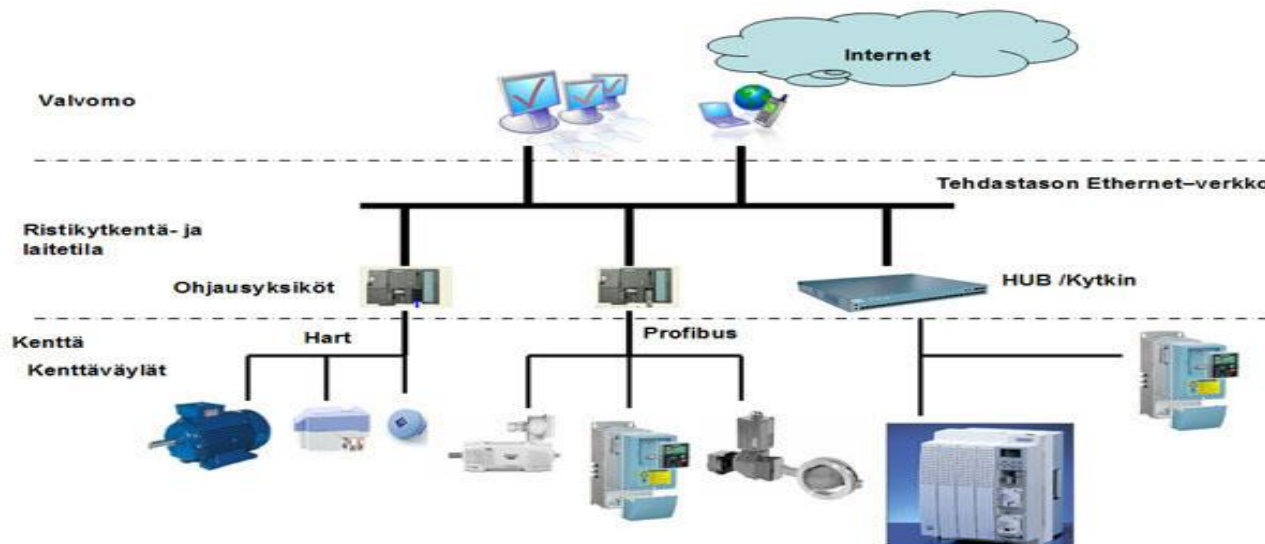
- kenttälaitteet eli mitta- ja toimilaitteet
- keskusyksiköt eli viestien vastaanotto- ja lähetinlaitteet, muuntimet, säätimet.
- valvomolaitteet eli näytöt, piirturit.

Digitaalisessa automaatiojärjestelmässä se voidaan jakaa kahteen osa-alueeseen:

- kenttälaitetaso
- järjestelmätaso

Näitä kahta tasoa erottavana rajana pidetään ristikytkentää (XL- ja JL-liittimet). Tämä raja hieman hämärtyy älykkäiden kenttälaitteiden ja digitaalisen kenttäväylien käyttöönotettaessa eli voidaan jo sanoa, että automaatiojärjestelmä ulottuu kentälle saakka. Älykkyyttä on siirtynyt kenttälaitetasolle, jolla tarkoitetaan mittaustietoa prosessista jalostetaan jo kentällä ja se antaa varsinaiselle järjestelmälle mahdollisuuden vapauttaa laskentakapasiteettia muuhun käyttöön. (Tuomi 2006.)

Kehityssuunta on tehostuva tietoliikenne tehdasväylissä ja -liitynnöissä tavanomaisiin toimistolähiverkkoihin ja sen ansiosta voidaan integroitua perusautomaation erityyppisiin toimintoihin. Informaatiojärjestelmiä kuten kunnonvalvonta-, tuotannon-ohjaus- ja yritysjohtonjärjestelmät on mahdollista yhdistää kokonaisautomaatioon. Puhumattaakaan ylemmän tason optimoivista säädöistä, joilla haetaan parhaimpia tuotannon kokonaistoimintaa kehittäviä malleja. Laitteiden muistikapasiteetin ja laskentatehokkuuden kehityskulun kasvaessa ei kasvulle näy olevan näköpiirissä rajaa, joka mahdollistaa yhä haastavimpien menetelmien käytön (hardware). Ratkaisevin asia, mikä vaikuttaa tähän, on automaation suunnittelu- ja toteutusympäristöjen muuttuminen kokonaisuudeksi. Järjestelmän laitekehitys sai aikaan laitteiden varusohjelmien kehityksen ja niiden kehitys johtaa aina vain laajempiin, haastavampiin ja ennen kaikkea monimutkaisempiin ratkaisuihin. Ohjelmistoilla haetaan tehokkuutta ja helppokäyttöisyyttä niin sovellussuunnittelijalle kuin loppukäyttäjälle. Ohjelmoitavuus käyttäjäystävällisempi, virhe herkkyyks vähenee suunnittelijan näkökulmasta ja operaattorin näkökulmasta tehokkaampi, helpompi käyttö ja kunnossapidon näkökulmasta parempi itsediagnostiikkaominaisuudet apuna. (Tuomi 2006.)



Kuva 3. Tyypillinen tehdastason automaatiojärjestelmä. (Opetushallitus. Automaatiojärjestelmä. Viitattu 23.3.2013.)

5.2. MetsoDNA-automaatiojärjestelmän rakenne

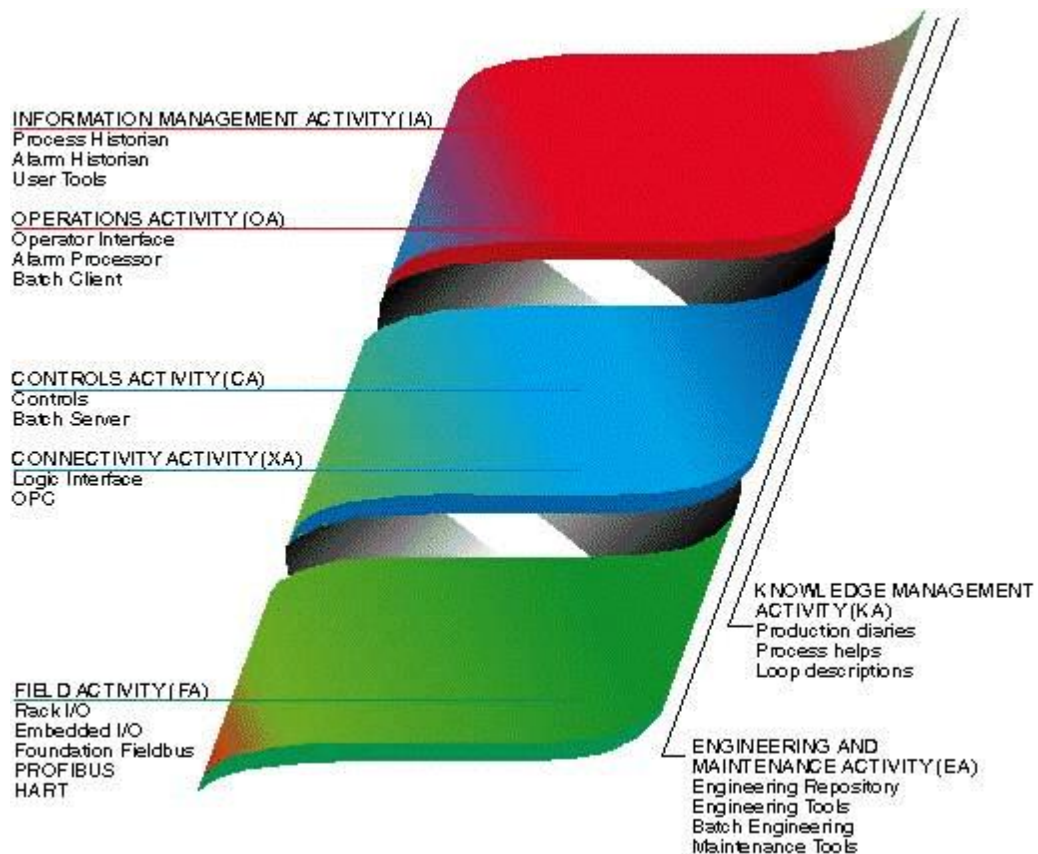
Tyypillisesti laitoksilla on eri versioita automaatiojärjestelmistä, koska käytännössä koko automaatiojärjestelmää ei uusita aina kun uusin versio julkaistaan. Se tulee ajan-kohtaiseksi vasta aikaisemman version vaatiessa laajempaa päivitystä. Uusin Metson tuotekonsepti on MetsoDNA, joka on dynaaminen sovellusverkko ja se perustuu tietämykseen ja informaation vapaaseen verkottamiseen, ohjausautomaatiikkaan ja sulautettuihin kenttäohjauksiin (DNA= Dynamic Network of Applications). Kahdennettu verkko, jossa sovellukset eri laitteiden ja ohjelmistojen kesken toimivat kokonaisuutena. MetsoDNA:ssa on aktiviteetteja, jotka liittyvät toisiinsa verkkotoiminnoilla ja ne voivat ulottua koko verkkoon tai osaan laitteistossa. Valvomoverkko, yhdistää komponentit yhteen MetsoDNA:a käyttöliittymän ja verkon, joka perustuu Ethernet-tekniikkaan. (Ylikunnari 2003, 30.) (Liite5)

Prosessiverkko kokoaa prosessin ohjaukseen tarvittavat komponentit ja prosessiohjauspalvelimet (kytketty valvomoverkkoon) mahdollistavat kommunikoinnin käyttöliittymän kanssa. Kenttäväylät yhdistävät I/O:t, kenttälaitteet ja prosessiohjauspalvelimet kokonaisuudeksi. MetsoDNA:aan ratkaisuun kuuluu myös toimistoverkko, joten käyttöliittymäkomponentit voivat olla liitettynä toimisto- tai valvomoverkkoon. Mutta

valvomoverkko pitää erottaa tietoturvan takaamiseksi toimistoverkosta., sillä tavoin saadaan samalla kevennettyä toimistoverkon taakkaa. (Ylikunnari 2003, 31.) (Kuva 3)

5.3. Aktiviteetit

Aktiviteettejä yhdistävää verkkoa MetsoDNA:ssa voidaan pitää ytimenä. Aktiviteeteilla tarkoitetaan eri järjestelmien toimintoja sisältäviä kokonaisuuksia, joilla kullakin on oma tarkoitus. Eri aktiviteetteja ovat tietämyshallinta-, informaationhallinta-, operointi-, säätö-, kenttä-, liityntä-, suunnittelu- ja ylläpitoaktiviteetit ja lisäksi on eDNA, jonka avulla voidaan tarjota online-tukipalveluita asiakkaalle. (Kuva 4) (Ylikunnari 2003, 31-32.)



Kuva 4. Aktiviteetit MetsoDNA (Ylikunnari 2003, 33).

Metso DNA:ssa ovat uusimmat aktiviteetit kuten User Interaction- , Automated Process- ja Secured lifecycle -aktiviteetit. User Interaction –aktiviteetti on alusta, jossa

yhdistyy työkalut tulostavoitteellisuuden saavuttamiseen tuotanto-organisaatiossa ja siinä on työkalut valvomotyöntekijästä yritysjohtoon saakka saman halutun tiedon saamiseksi. Siihen liittyy Metso DNA Report Diary (tuotannon päiväkirja), jolla välitetään kaikille samat tosiasiat ja mielipiteet samassa paikassa. (Metso Oyj. Feel the ease. User Interface. 2011)

Automated Process –aktiviteetissä on säätöratkaisut kaikkiin prosessikohteisiin ja samalla teknologialla voidaan toteuttaa hajautetut kenttäratkaisut, laatusäädöt ja erilaiset kenttäväylästandardit. Se koostuu ratkaisuista prosessin säätöön ja optimointiin, sisältäen prosessiohjaimet niin liitynnät kentälle kuin muihin järjestelmiin. (Kosonen 2012.)

Secured Life Cycle –aktiviteetti on yleisesti järjestelmän hallintaan tarkoitettu, jossa on työkalut automaatiosuunnitteluun, järjestelmän dokumentointiin ja kentälaitteiden hallintaan. Erilaisia työkaluja voidaan käyttää keskitetysti samassa käyttöliittymän kautta ja toimintojen ydin on DNA Explorer, jossa on käyttöliittymä työkaluille ja sovellustietokannoille. Älykkäille kentälaitteille konfigurointiin ja kunnonvalvontaan tarkoitettu työkalu on DNA Field Device Configurator, jossa on helposti saatavilla operaattoreille ja kunnossapitohenkilöstölle tarvittavat konfiguraatio- ja kunnonvalvontatiedot. Sillä mahdollistetaan nopea pääsy tarvittavaan tietoon ja löytämään ratkaisut nopeasti häiriötilanteissa. (Metso Oyj. The hard facts for a good feeling. 2011.)

5.4. Verkon rakenne MetsoDNA:ssa yleisesti

Metso DNA on avoimen verkon sisältävä järjestelmä, johon on mahdollisuus kytkeä muidenkin valmistajien järjestelmiä, prosessitietokoneita ja logiikoita. Kytkeytyminen on mahdollista sarjaliitynnällä, Ethernet-väylän kautta tai OPC-linkillä. Kenttälaiteväylät, jotka voidaan liittää järjestelmään ovat Profibus DP ja PA, Foundation Fieldbus ja AS-Interface. Tällä yhdellä järjestelmällä pyritään tarjoamaan kaikki, mitä tarvitaan erilaisissa prosessiohjauksissa. Liitynnät, säätö prosessiohjaukseen, koneohjauksiin ja sähkökäyttöille, että mekaanisen valvonnan sovellukset kuuluvat Metso DNA:n konseptiin. Nykypäivänä aina tärkeämmäksi tuleva turvallisuus on huomioitu Metso DNA:ssa tavalla, joka ilmaistaan kirjain yhdistelmällä TLJ (turvallisuuteen liittyvä järjestelmä). TLJ integroituna Metso DNA:han tuo mainittavia etuja, kuten kaikille yhtei-

nen käyttöliittymä, hälytysten käsittely, yksi historiatietokanta ja sovellusten toteutus yhdellä työkalulla, joten kunnossapitohenkilökuntaa ei tarvitse kouluttaa moniin erilaisiin järjestelmiin. (Metso Oyj. Feel the ease of one system. 2011.)

Metso DNA:ssa on tietenkin tehdasstandardit täyttävät komponentit, joilla varmistetaan turvallinen käytettävyys ja varaosa- ja päivystyspalvelu kuuluvat standardiin sekä järjestelmän kahdennettavuus. Kahdennettavuudella varmistetaan järjestelmän toiminta vikatapauksissa, joissa jokin väylistä vioittuu. Kahdennusta voidaan käyttää Ethernet -verkossa, I/O-väylissä, Profibus DP ja Foundation Fieldbus HSE:ssä, kiintolevyissä, jännitesyötöissä, liityntä- ja ohjaussolmuissa, operaattorin käyttöliittymissä, hälytys-, tapahtuma- ja historiakäsittelyssä. Tietoturvallisuus on toteutettu niin, että kaikki Microsoft Windows-pohjaiset käyttöliittymät (solmut) ovat tietoturvakovennettuja ja virusohjelmilla varustettuja. Palomuuuri on erottamassa toimistoverkko automaatiojärjestelmästä ja niin sanottu etäyhteys suojattu käyttäjätunnuksilla ja salasanoilla, jota kutsutaan nimellä Metso SCS (Secured Connection Solution). (Metso Oyj. High availability. 2011.) Metso DNA:lla voidaan toteuttaa niin analogiset kuin digitaaliset ohjaukset, joita ovat mittaukset, PID-säädöt, moottorien ja venttiilien ohjaukset. Ylemmän tason ohjauksiin, joita toteutetaan ovat sumeat säädöt, neuroverkot, monimuuttujasäädöt ja Java-ohjelmat. (Metso Oyj. True scalability. 2011.)

5.5. EA suunnittelu- ja ylläpitoympäristö

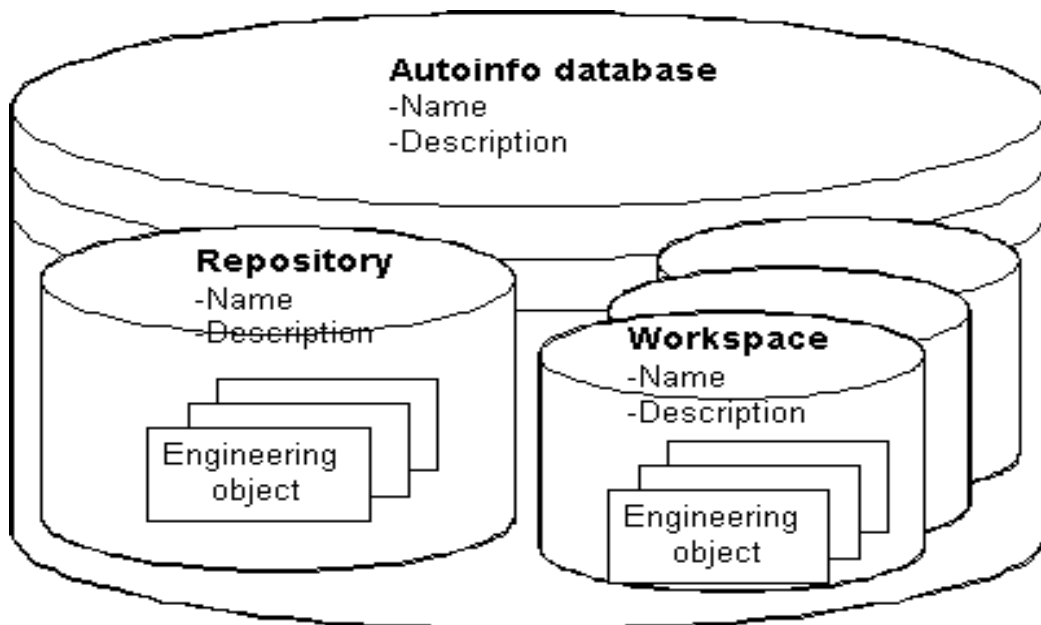
Metso DNA Explorer kaiken ydin suunnittelu- ja ylläpitoympäristössä. Metso DNA:ssa, sillä hallitaan järjestelmän konfigurointia, ylläpitoa ja testausta. Se on työkalu, jolla hallitaan dokumentaatiota, kenttäväyliä, kenttälaitteita ja ohjaussovelluksia. Siinä on mahdollista jopa tehdä 3D-mallinnuksia prosessinkomponenteista, kuten säiliöistä ja putkilinjoista. Järjestelmän komponentteina ovat esimerkiksi:

- Metso ACN PO, käytetään hälytysten käsittely-, varmistus- ja käyttöliittymäasemana
- ACN RT, ohjaussolmuna eli prosessiasemana
- ACN AS, teollisuuspalvelimena pienissä ja keskisuurissa sovelluksissa, jota käytetään trendi- ja hälytysarkistona, kunnonvalvonnassa ja suunnittelusolmuna (ACN kaappi)
- ACN WS, teollisuuslaadun Windows-työasema, käytetään kuin AS:ssä mutta pöytä

malli

- ACN CS, teollisuustietokone kenttäkoteloon, kaappiin tai hyllylle
- ACN SR1, prosessinohjauspalvelin ACN:n I/O-tuotteille. Operoi itsenäisesti tai osana sovellusverkkoa (Metso Automation Oy 2010) (Ylinen 2011)

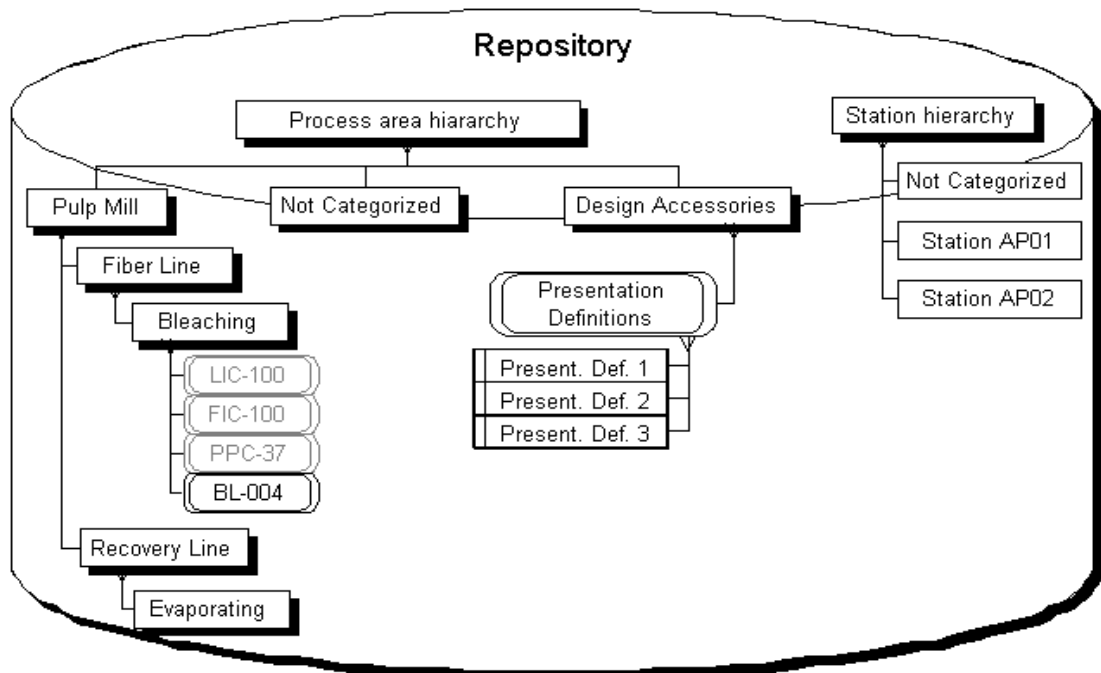
EAC-laitteisiin on asennettu niin sanottu Client-työkalut, jotka mahdollistavat EAS-laitteen autoinfotietokannan käytön. Autoinfotietokantahan on yhteinen suunnittelutietokanta suunnittelujärjestelmän sovelluksille ja sinne tallentuu autoinfotietomallinnuksen tuloksena syntyvä malli. Autoinfotietokanta muodostuu erillisistä tietovarastoista, makasiinista ja työtiloista. Työtiloissa mallinnusta työstedään ja valmiiksi saadut mallit integroidaan makasiiniin. (Ylikunnari 2003, 46.) (Kuva 6)



Kuva 6. Autoinfotietokanta, jossa on neljä tietovarastoa, yksi makasiini ja kolme työtilaa. (Ylikunnari 2003, 46.)

Tietovarastoissa on tallennettu kaikki suunnitteluun tarvittavat oliot, joita tarvitaan autoinfotietomalliin (autoinfotietotoiminnot, prosessialueet, asemat) ja eri tietovarastoja ovat makasiini (Repository) ja työtilat (Workspace). Erilaisia tietovarastoja on, koska suunnittelutyönvaiheet jaotellaan suunnittelun eri vaiheissa tapahtuviksi kokonaisuuk- siksi. Makasiini on tietovarasto, jossa on suunnitteluolioita ja se toimii loppukäyttäjän yhteisenä tietokantana. Työtila on tietovarasto, josta löytyvät suunnitteluoliot ja työtilat ja siellä työstedään mallia ja työtiloja. Autoinfotietokannassa niitä voi olla useampi

toisin kuin makasiinia. (Ylikunnari 2003, 46-47.) (Kuva 7)



Kuva 7. Repository. (Ylikunnari 2003, 47.)

5.5.1. DNA Explorer

Toimintohelä eli DNA Explorer (entinen Function Explorer) on suunnittelun työkalu ja se on tarkoitettu lähinnä sovellussuunnittelun ja ylläpidon avuksi. EA (Engineering and Maintenance Activity) eli MetsoDNA:aan suunnittelu- ja ylläpitoaktiiviteetti tukee automaatiosuunnittelussa eri osa-alueita kuten automaation instrumentointi, automaation sähköistys, automaatioverkko ja sovellus. Toimintoheläällä suunnitellaan sovellusmalli suunnitteluympäristön autointietokannassa ja tuotettu malli ladataan MetsoDNA:aan ajoympäristöön, missä malli liittyy ajoympäristön kautta todelliseen prosessiin. Valmis tuotettu malli sisältää ajoympäristöön tarvittavat sovellukset eri aktiiviteeteille esimerkiksi säätö-, operointi- ja informaationhallinta-aktiiviteetit ja muut sellaiset. (Metso DNA Manuals 2011.)

Graafiset työkalut kuten FbCAD (toimilohkokaaviot), SeqCAD (sekvenssikaaviot), Picture Designer (kuvat), FF Explorer (FF –väyläkonfiguraatiolle), PROFIBUS Configurator (Profibus –väyläkonfiguraatiolle), Control Diagram CAD (säätökaaviolle), Field CAD (kenttäkaaviolle), Hardware CAD (laitteistokaaviolle), Loop Circuit CAD (piirikaaviolle), Motor CAD (piiri- ja johdotuskaaviolle), Logic CAD (logiikkakaavi-

olle) ja tekstieditori (automaatiokielisille konfigurointitoiminnolle) ja toimintose-
laimesta ovat nämä graafiset suunnittelutyökalut käynnistettävissä halutulle suunnit-
teluoliolle. (Metso DNA Manuals 2011.)

5.5.2. Käyttöliittymä

Autoinfotietokannan sisältöä käsiteltäessä toimintose-
laimen pääikkuna sisältää jäsen-
telijäikkunoita, jotka ovat lista- (List Hierarchy), paketti- (Package Hierarchy) ja pro-
sessialuejäsentelijä (Process Area Hierarchy) ja suunnitteluvaiheessa jo toteutetut mal-
lit löytyvät näistä jäsentelijöistä. Listajäsentelijästä sisältää kaikki yhdellä listalla vali-
tun tietovaraston autoinfotietotoiminnot, pakettijäsentelijä esittää valitun paketin tietoa-
varaston autoinfotietotoiminnot ja prosessijäsentelijä näyttää valitun prosessialueen mu-
kaan jäseneltynä olevan tietovaraston autoinfotietotoiminnot. Tällä käyttöliittymällä
voidaan käsitellä suunniteltua mallia tai kokonaisuutta ja muokata sitä. (Metso DNA
Manuals 2011.)

Tärkeimmät suunnittelutyökalut MetsoDNA:ssa ovat FbCAD, Picture Designer,
HwCAD ja muut sellaiset. FbCAD -työkalu on EA Server-palvelin ja EA Client ympä-
ristössä ja se voi toimia osittaisilla toiminnoilla Windows NT-pohjaisessa työase-
massa. FbCADilla suunnitellaan toimilohkokaavioita, jotka MetsoDNA:aan ohjaamat
ja säätämät prosessit tarvitsevat ja se sisältää toimilohkokaavion erilaiset konfiguroin-
titoiminnot prosessin ohjaukseen ja säätöön. Toimilohkokaaviot tallennetaan EA Ser-
ver -palvelimella autoinfotietokantaan, jotta niitä voidaan myöhemmin tutkia tai muo-
kata sekä FbCADilla suunniteltu toimilohkokaavio on graafinen dokumentti ja ajoym-
päristöön ladattava sovellus. SeqCAD-työkalu on prosessiaseman sekvenssiohjelmien
suunnitteluun ja dokumentointiin tarkoitettu työkalu. (Metso DNA Manuals 2011.)

Picture Designer-työkalu on kuin FbCAD ja samassa ympäristössä. Työkalu tunnistaa
kaavionäyttötoimilohkot ja sillä voidaan muokata toimilohkoja ja grafiikkaa. Lisäksi
kaikki konfiguraatiot dokumentoidaan, mikä varmistaa ohjelmistojen ja dokumenttien
pysymisen ajan tasalla ja lopuksi tallennus EA Server-palvelimen makasiiniin.

HwCAD-työkalulla suunnitellaan niin kokoonpanoja kuin layouteja ja se on dokumen-
tointiohjelma, jolla ylläpidetään laitteistojen dokumentteja. Tallennus EA Server-
palvelimeen tai EA Client -työaseman levyille mutta huomioitavaa on, että tallennusta
ei voi tehdä makasiiniin tai työtiloihin. (Metso DNA Manuals 2011.)

Control Diagram CAD -työkalulla suunnitellaan ja dokumentoidaan loogisia säätökaavioita standardin mukaisilla symboleilla. Logic CAD -työkalulla suunnitellaan ja dokumentoidaan loogisia toimintakaavioita ja tietokantojen käyttö suunnittelussa edesauttaa työtä. Usein projekteissa toteutetaan samalla toimintaperiaatteella toimivia piirejä, joista puuttuu vain jokin pieni osa. Ne muuttamalla saadaan piiri toteutettua helpommin ja näin hyödynnetään jo aiemmin samankaltaista piiriä pienin muutoksin. (Metso DNA Manuals 2011.)

6. ELINKAARI KÄYTTÖJÄRJESTELMISSÄ JA OHJELMISTOISSA

Ohjelmistojen elinkaari tarkoittaa ajanjaksoa, jossa sovelluksen kehityksen alkaa aina käytöstä poistamiseen ja sen välisenä aikana tapahtuu paljon asioita. Asiakas on yleensä motivaattori kehittämään uusia ja parempia ohjelmistoja nykyhetken tarpeisiin. Käytännössä se tapahtuu siten, että asiakas haluaa jotakin ja toimittaja alkaa tutkia ja kehittää asiakkaan toivomuksen mukaista ohjelmistoa. Usein aikaa on liian vähän kehittämiseen tai asiakas ei ole tyytyväinen saatuun tuotteeseen, joten muutoksia joudutaan tekemään. Näitä niin sanottuja päivityksiä tulee sitä mukaan kuin korjattavaa tai parannettavaa on ja alkuperäinen ohjelma muokkautuu hiljalleen ongelmien ilmaantuessa ja niiden poistamisessa. Ohjelmistot testataan, mutta usein tosi toimissa oleva ohjelmisto näyttää puutteet ja tämä johtaa niin sanottujen päivityksien kehittämiseen. Ohjelmiston elinkaari kulkee tiettyä mallia myöten: vaatimukset ja esitutkinta, määrittely, suunnittelu, toteutus, testaus (integrointi), käyttöönotto ja ylläpito. (Cordata 1983, 232-233)

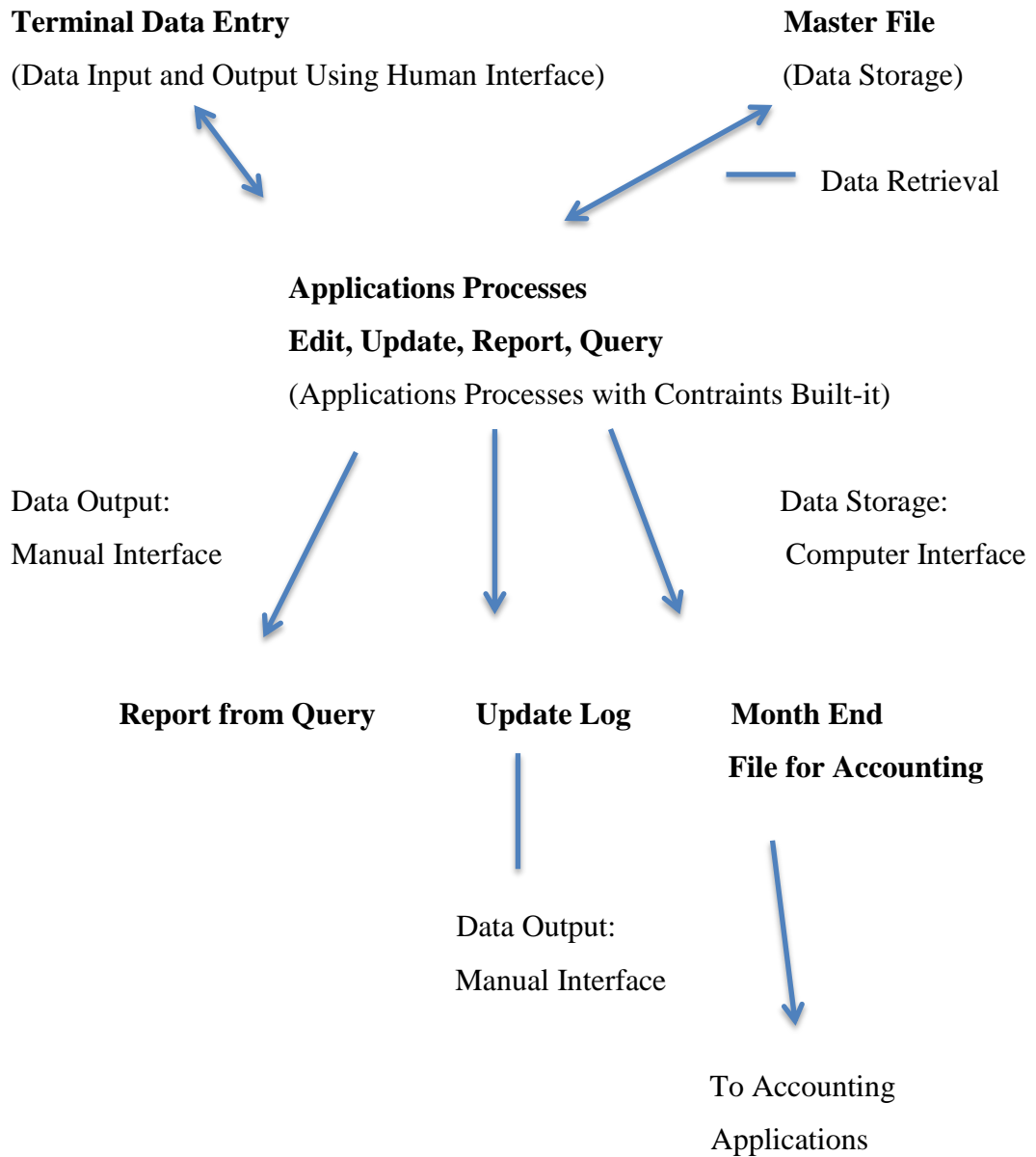
Ohjelmistoissa esiintyvien ongelmien tai käyttöä vaikeuttavien monimutkaisuuksien ratkaisujen löytäminen noudattaa usein seuraavaa kaavaa. Ongelmasta ilmoitetaan atk-tukeen tai hallintoon ja tekninen henkilöstö tutkii ongelmaa sekä ehdottaa ratkaisua. Kysytään uusia vaatimuksia ongelman suhteen ja muutetaan lopulliseen järjestelmään niin, että se toimii halutulla tavalla. Ja lopuksi selvitetään, aiheuttavatko muutokset muuta ongelmaa muualle järjestelmään. Usein asiat eivät ole näin yksinkertaisia, koska muutokset yleensä vaikuttavat moneen asiaan puhuttaessa tietotekniikasta. Ilmoittaja saattaa olla tietämättömyyttään aiheuttanut ongelman, jonka ammattitaitoinen atk-tukihenkilö huomaa keskustellessaan ilmoittajan kanssa ongelmasta, jota tukihenkilö

pyrkii kartoittamaan.

Ohjelmistot saattavat toimia useita vuosia hyvin, mutta kuitenkin törmätään vääjäämättä kahteen ongelmaan. Ohjelmisto ei vastaa enää nykypäivän kasvaviin vaatimuksiin ollen hidas, mutta siltä vaadittaisiin nopeampaa reaktiota ongelman selvittämiseen ja ratkaisemiseen. Toiseksi laaja määrä ohjelmistoja tekee tukihenkilön kalliiksi, koska tarvitaan tukihenkilön apua, joka ei ole saatavilla tai tarpeeksi osaavaa ja koulutettua henkilöä. Tämä saattaa olla pitemmällä aikavälillä huono ja kallis tosiasia, koska uusia, tehokkaampia tapoja elinkaassa tullaan aina tarvitsemaan. Niin sanottujen ohjelmistojen ja järjestelmien ylläpito asettaa nykypäivänä aikamoisen haasteen asiaan vihkiytyneille, mutta kuitenkin haasteisiin vastataan esimerkiksi niin, että tukihenkilöt ovat tavoitettavissa 24/7, vaikkakin se lisää kustannuksia. (Cordata 1983, 232-233).

Teollisuudessa kaikki häiriöt ovat ongelmia prosesseissa ja ne pitää saada nopeasti korjattua, ettei aiheudu lisäkustannuksia. Tähän ongelmaan on ratkaisu ja sitä kutsutaan varmuuskopioksi, niiden avulla voidaan asettaa (konfiguroida) uudet oikeat asetusrvot ja näin ei ole kadotettu lopullisesti häiriössä kadonneita tai tuhoutuneita asetusrvoja. Seuraava kuva 8 esittää yleiskuvan ohjelmistosuunnittelun haasteista.

Overview of Software Engineering



Kuva 8. Ohjelmistosuunnittelijan haasteet. (Conger 1994, 328)

6.1. AutoCAD-ohjelmiston ja käyttöjärjestelmien elinkaari

AutoCAD-suunnittelutyökalu on automaatiojärjestelmän suunnitteluun ja toteutukseen tarkoitettuja komponentteja sisältävä työkalu. Näillä komponenteilla rakennetaan erilaisia säätö-, mittaus-, ohjaus- ja muita sellaisia –piirejä. Kuitenkin tämä AutoCAD -ohjelmisto ajan myötä vanhenee, joten ohjelmistoa täytyy päivittää. Kehitetään uusia innovaatioita ja parannetaan niin sanottua vanhaa versiota, koska on käytännön työssä todettu, että jokin ei toimi halutulla tavalla tai voitaisiin tehdä yksinkertaisemmin. Päivityksillä yleisesti haetaan helppokäyttöisyyttä, parannuksia ja ominaisuuksia verrattuna vanhempaan versioon.

Nykypäivänä kustannustehokkuus määrittää monessa asiassa vaatimukset niin ohjelmistoille kuin laitteille. Liitteessä 1 havainnollistetaan kuvalla suunnitteluohjelmiston kehitystä Metson järjestelmissä. Seuranta otos on Metso-versioista vuodesta 2000 vuoteen 2011 ja versioita on kuvassa esitetty vuosittain 2000 - 2011 versiot 1.1-14.x välillä. CAD-versiot ovat 2000 - 2009 AutoCAD-nimellä, mutta vuodesta 2010 eteenpäin BricsCAD-nimellä ja kuvassa oikealla on käyttöjärjestelmä versiot esitettynä aikana 2000 - 2011. Käyttöjärjestelminä Metso käyttää Windows-pohjaisia järjestelmiä ja havaitsemme kuvasta, että eri ohjelmien pitää sopia keskenään yhteen. Tämä ongelma kohdataan useasti käytännön työssä, jossa käyttömukavuus kärsii ja aiheuttaa työskentelytehokkuuteen laskua. Uusia ominaisuuksia ei voida hyödyntää aina, koska ne eivät toimi toisen ohjelmiston kanssa, ennenkuin se on päivitetty ja tämä kilpajuoksu antaa haastetta ohjelmistojen suunnittelijoille. Kuva osoittaa, että ei pelkkä ohjelmiston rakentaminen riitä, vaan sitä tuetaan eli kehitetään matkan varrella. Metson periaatteisiin kuuluu ensiksi testata versiot sisäisesti ennenkuin se lasketaan tuotantoon ja tämä siksi, että Metso säilyttää laatutason, joka on kaikkien osapuolien kannalta hyvä. (Metson PowerPoint-esitys 3.11.2009)

6.1.1. Microsoftin tarjoama tuki eri käyttöjärjestelmille

- Windows NT with Service Pack 6a, ei tueta 31.12.2004 jälkeen
- Windows 2000 with Service Pack 4, ei tueta 10.7.2010 jälkeen
- Windows XP Service Pack 2 - Service Pack 2:n tuki loppui 13.7.2010, mutta Ser-

vice Pack 3 tukeminen jatkuu 8.4.2014 saakka.

- Windows Server 2003 Service Pack 1, Service Pack 1 on loppunut, mutta Service Pack 2 tuetaan vielä 14.7.2015 saakka.

Microsoft tuki sisältää: tuotanto-, ylläpito- ja lopetusvaiheen tuotteilleen. Eri tuotteiden elinkaaret ovat aina hieman päällekkäin, jolla tarjotaan päivityksiin liikkumatilaa.

6.1.2. MetsoDNA:n ja Microsoft Windows-järjestelmien historiasta

- C2001 and C2002 sisältää Windows NT and Windows NT Server
- C2003 – C2005 sisältää Windows 2000 Professional and Server
- C2003 – C2010 sisältää Windows XP Professional and Windows 2003 Server
Windows Vista käyttöjärjestelmä jätettiin väliin.
- C2011 => sisältää Windows 7 and Windows 2008 Server R2
Windows 8 on jo suunnitelmissa 2014 ja Win 8 Server 2015.

MetsoDNA:n ja Microsoft Windows-tuotteidenkin noudattaa samanlaista kaavaa kuin edellisessä kappaleessa mainittiin. Uusimmat tuotteet MetsoDNA:lta ja Microsoft Windows:lta kulkevat rinta rinnan ja yhteensopivat parhaiten uusien ominaisuuksin.

6.1.3. Windows-alustat, joita tuetaan MetsoDNA:ssa

Windows XP Service Pack 3 Windows Server 2003 Service Pack 2:

MetsoDNA C2007, jossa tuetaan pidennetty, MetsoDNA C2008, C2009, C2010.

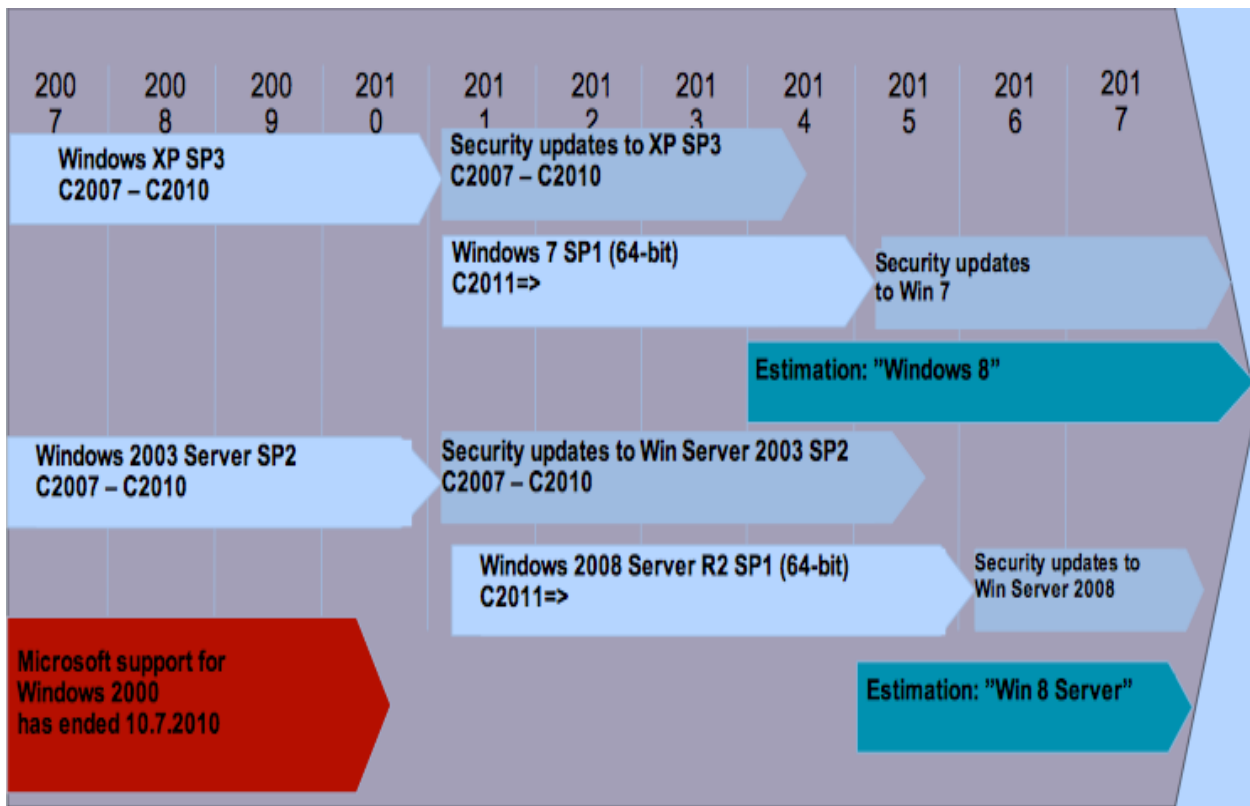
Windows 7 SP1 (64-bit) ja Windows 2008 Server R2 SP1

Windows 7 (32-bit) tuetaan työasemissa

Windows XP SP 3 tuetaan myös työasemissa (Kiviniemi 2011.)

Kuten huomataan Metso käyttää järjestelmien nimityksessä collection nimeä C2001, C2002, C2003 - 2005, C2003 - 2010 ja C2011. Se koostuu Metson automaatiojärjestelmästä, johon kuuluu suunnittelu- ja prosessinohjaukseen tarvittava ohjelmisto kokonaispakettina. Käyttöjärjestelmä ohjelmistona hallitseva on ollut Windows XP Professional ja on todettu varmatoimiseksi, josta johtuen Windows Vista on tämän ohitettu eli jäänyt pois käyttöjärjestelmä puolella. XP:n jälkeen on siirrytty suoraan Win-

dows 7:n.



Kuva 9. Käyttöjärjestelmien elinkaari. (Metson PowerPoint-esitys 3.11.2009)

6.1.4 Kokonaisjärjestelmän päivittäminen

Aikaisempi tai olemassa oleva automaatiojärjestelmä määrittää mahdollisuudet, miten iso päivitys on järkevää tehdä. Jos asiakkaalla on vanhinta versiota Windows NT tai Windows 2000, olisi suositeltavaa päivittää suoraan uusimpaan C2011:een, missä on Windows 7 käyttöjärjestelmä ja Windows Server 2008 R2. Mikäli asiakkaalla on C2006 tai vanhempi, niin riittää päivitystarve C2007:een, koska siinä on pidennetty version tuki. Mikäli asiakkaalla on C2007 tai uudempi versio riittää Service Pack tason korotus. (Oksanen 2009.)

Yleisesti MetsoDNA:n tuki on Windows 7 ja Win 2008:sta tuetaan vain C2011:tä eikä uudet Windowsit tue vanhempia versioita. AutoCAD 2006 ja GdCAD combo tuetaan vielä C2011:tä, mutta lisenssin saatavuus voi olla ongelma. MS Terminal Server ja Viivissä parhaimpaan tulokseen päädytään, jos valitaan uusin versio. Koska C2011:ssä

on Viivi-yhteensopivuus, niin käyttöjärjestelmä on Win 7 tai Win XP, vaikkakin vaihtoehtoja on muitakin. Esimerkiksi, jos vanhan softan odotetaan yhteensopivan, toimitetaan asiakkaalle palvelin, jossa on Windows 2003 Server SP2-käyttöjärjestelmä ja siihen asennetaan Windows Terminal Services. Käyttäjät toimivat Viivin kanssa, joten sitä käytetään MS Remote Desktop Clientilla Terminal Services palvelimelta, eikä sitä asenneta paikalliselle koneelle, mutta pitää muistaa tämän tavan olevan vain niin sanottua tekohengitystä. Esimerkiksi Metso DNA demot toimivat pilvessä Remote Desktopilla. (Kiviniemi 2011.)

6.2. Ohjelmistotekniikka

Ohjelmistojä toteutettaessa sen suunnittelu ei ole niin yksinkertaista, koska ohjelmistotuotannon kehitykseen liittyy monia asioita. Kehitys noudattaa tiettyä kaavaa. Esimerkiksi aluksi riittää, kun saadaan jokin toimiva tapa ongelmaan ja siihen löytyy ratkaisuja, jotka toimivat useammassa tapauksissa. Siitä seuraa se, että tietotaito kehittyy systemaattiseksi ja työskentelymenetelmiksi. Seuraava askel on menetelmien kehittyminen riittävästi tukeakseen malleja ja teorioita. Uusien mallien ja teorioiden käyttö lisääntyy entistä vaativimmissa sovelluksissa ja siitä seuraa uusia ongelmia ja kokonaisia sovellusalueita. Uusiin sovellusalueisiin ei ole valmiita ratkaisuja ongelmiin, josta johtuen palataan takaisin alkuun eli etsitään toimivaa tapaa ratkaista ongelma. (Haikala - Märijärvi 2001, 15-16.)

Koska ohjelmistotuotanto on kuitenkin nuori (lyhyt historia), sitä voidaan pitää syynä alan kehittymättömyyteen. Ominaisuuksina ohjelmistotyössä voidaan jakaa seuraavasti: monimutkaisuus, näkymättömyys ja muunneltavuus. *Monimutkaisuudella* tarkoitetaan ohjelmistojä, joilla pyritään ratkaisemaan monimutkaisia ongelmia. Hyvä ohjelmistosuunnittelukaan ei pystytä poistamaan monimutkaisuutta vaikka tavoite on minimoida monimutkaisuutta ja saada yksinkertaista ja selkeää, jotta ohjelman hallinta on mahdollisimman helppoa. Järjestelmien kasvaessa suureksi tulee monimutkaisuuden välttäminen hankalaksi. *Näkymättömyydellä* tarkoitetaan ohjelmistotyön etenemistä projektinhallinnan näkökulmasta, koska työn etenemistä on vaikea arvioida tarkkaan. *Muunneltavuudella* tarkoitetaan ohjelmiston helppoa muokkausta. Ohjelmistotyö etenee usein juuri niin, että ohjelmistojen vaatimukset muuttuvat tai tarkentuvat työn edetessä ja aiheuttaa muutospaineita ohjelmistoihin. Tämä siitäkkin syystä, että ohjel-

mistojen muuttaminen tässä vaiheessa kustannusnäkökulmasta katsottuna on halvempaa.

Valmiin ohjelmiston toimiessa toimintaympäristössään havaitaan ohjelmistossa virheitä, jotka saattavat johtua toimintaympäristössä tapahtuvista muutoksista ja vaatimuksista eli ohjelmiston muutospainet jatkuvat sen koko elinkaaren ajan. Hyvällä suunnittelulla voidaan vähentää muutospainetta, vaikkakin muutospainet kuuluvat ohjelmiston elinkaareen. Tässä kohtaa juuri ohjelmiston ylläpidettävyyden tärkeys ja ylläpidon ominaisuudet voidaan jakaa seuraaviin osa-alueisiin: ainutkertaisuus, skaalautumattomuus ja epäjatkuvuus. *Ainutkertaisuus* tarkoittaa ohjelmistojen olevan ainutkertaisia, koska aivan samanlaista ohjelmistoa ei tule uudestaan. Alan kehityskaari on niin kovassa nousussa, koska jatkuvasti suunnitellaan uusia sovelluksia ja uutta teknologiaa ohjelmistotekniikan piiriin. *Skaalautumattomuus* eli aikaisemman kokemuksen perusteella ohjelmisto ei toimikaan järjestelmän koon kasvaessa. Pienemmän ohjelman ollessa kyseessä hallitsevat sen useat projektiin osallistuvat ja omaavat riittävän tietotaidon sen käyttöön. Ongelman ilmaantuessa saattaa yksittäinen henkilö ratkaista ongelman, mutta isomman projektin ollessa kyseessä ei yksittäinen henkilö onnistukaan ratkaisemaan ongelmaan monimutkaisuuden tai järjestelmän koon takia. *Epäjatkuvuus* eli ohjelmistojärjestelmien käyttäytyminen virhetilanteissa ei ole useinkaan epäjatkovaa, mutta ohjelmistoille on tyypillistä, että merkityksettömältä vaikuttava viika voi lamauttaa koko järjestelmän. Epäjatkuvuus on syynä siihen, että ohjelmistojen testauksella ei voida varmistaa niiden toimintaa varmasti, koska testaukseen ei ole mahdollista saada aikaan kaikkia erikoistapauksia tai niiden yhdistelmiä. (Haikala - Märijärvi 2001, 16-19.)

6.2.1. Ohjelmiston elinkaari

Lyhyesti sanottuna se on ohjelmiston kehityksen alkamisesta sen poistamiseen käytöstä ja tähän elinkaareen sisältyy niin sanottuja tukitoimintoja. Tukitoiminnot ovat esitutkimus, määrittely, suunnittelu, toteutus, integrointi ja testaus, käyttöönotto ja ylläpito. Määrittely vaihe voidaan jakaa vielä kahteen osa-alueeseen toiminnalliseen ja tekniseen. Toiminnallinen määrittely on dokumentointi eli siinä kuvataan haluttua toimintoja ja toteutukselle asetettavat ei-toiminnalliset vaatimukset ja rajoitukset. Toimintojen yhteydessä määritellään toteutettavat ominaisuudet, käyttöliittymä ja kommuni-

kointi eri järjestelmien kanssa. Tekninen määrittely tarkoittaa dokumenttia, jossa järjestelmä on jaettu itsenäisiin, toisistaan riippumattomiin moduuleihin. Moduulien sisäinen rakenne suunnitellaan koko järjestelmän eri osissa. (Haikala – Märijärvi 2001, 42.) Voidaan todeta, että esimerkiksi automaatioalalla ollaan siirrytty palveluliiketoiminnan suuntaan eli laitteiden ja järjestelmien myymisestä kestävämpään elinkaariajattelumalliin pelkän kaupankäynnin sijaan.

6.2.2. Elinkaarikustannukset

Kustannukset kattavat tietenkin ohjelmiston kehittämisestä poistamiseen ja kehittämisestä sen ylläpitoon. Elinkaaren kustannuksiin vaikuttavat eri toiminnot jakautuvat niin, että järjestelmän kehitysvaiheessa kokonaisvaltaisesti katsottuna on suhteellisen pieni noin 30% kokonaiskustannuksista kun käytönaikaisella osuudella se on noin 70%. Eli ylläpidolle jakautuu suurin kustannuserä, koska ylläpitohan koostuu ohjelmiston ajan tasalla pitämisestä toiminnan ja ympäristön muuttumisen vaatimista muutoksista johtuen. Uusia ominaisuuksia saatetaan lisätä ja kustannussäästöjä ajatellen ylläpidon laatuun pitää satsata, koska apuna oleva huolellinen suunnittelu ja aina niin merkittävä dokumentointi pitää olla selkeää ja riittävän kattava, jotta ylläpidon on helppoa ymmärtää toiminta. (Haikala - Märijärvi 2001, 42-43)

6.2.3. Tuotteenhallinta

Ylläpitovaiheessa ohjelmistotuotteiden hallinta korostuu. Erilaisia muutoksia (konfiguraatioita) syntyy, joko normaalin versioitumisen takia tai laite- ja ohjelmistoympäristössä tapahtuvien muutosten takia (asiakaskohtaiset muutokset). Asiakkaan suunnalta tulevien virheraporttien ja muutospyyntöjen takia, joten tuote muokkautuu elinkaaren aikana. (Haikala – Märijärvi 2001, 245-246.)

Yleensä testausvaiheessa ilmenee jo virheitä ohjelmistossa ja testauksessa käytetään toiminnallista ja teknistä määrittelyä. Erimielisyyksiä voi ilmetä dokumenttien tulkinassa asiakkaan ja toimittajan välillä esimerkiksi asiakkaan mielestä jokin ohjelmiston toiminto on selvä virhe kun toimittaja pitää sitä niin sanottuna ominaisuutena. Virheen laatu voi olla joko järjestelmän käyttöä hankaloittava virhe tai asiakasta ärsyttävä yksi-

tyiskohta ja kaikkea siltä väliltä. (Haikala - Märijärvi 2001, 269.)

6.2.4. Ohjelmointiin liittyvät erityisongelmat

Piirteitä reaali-ohjelmoinnissa voidaan pitää esimerkiksi poissulkemista, synkronointia, ajastusta ja lukkiutumista. *Poissulkeminen* on tilanne, jossa usean eri prosessin samanaikainen tietorakenteen käsittely on suojattava, ettei tapahdu päällekkäisyyksiä ja sekaannuksia. *Synkronointi* on tilanne, missä prosessi odottaa jotain tapahtumaa. *Ajastus* on tilanne, jota käytetään varmistuskopiointiprosessin käynnistämiseen eli asetetaan jokin tapahtuman odotteluun jokin raja. *Lukkiutuminen* on tilanne, jossa ohjelma odottaa jotakin tapahtumaa, joka ei tapahdu. Muistitilan odottaminen ja sen loppuessa on yleinen lukkiutumistilanne ja ohjelmointivirheet poissulkemisessä tai synkronoinnissa aiheuttavat lukkiutumisen yleensä. (Haikala – Märijärvi 2001, 319-320.)

7. YLLÄPIDON ONGELMAT

Ensimmäisenä ja yleisimpänä ongelmana voidaan pitää *kiirettä*. Järjestelmän rakenteen laajuuteen ja tuntemukseen nähden suunniteluun annettava aika liian lyhyt ja saattaa aiheuttaa turhaa työtä (voimavarojen menetystä muusta työstä) ja kustannusmenetyksiä.

Toimeksiantojen epämääräisyys eli korjauksien tai muutoksien määritys puutteellista, mistä voi syyllinen löytää niin työn vastaanottajan kuin toimeksiantajan suunnasta. *Ylläpidon hallinta* on ongelma niin käyttäjälle kuin ylläpidolle, koska eri osapuolet eivät aina ole tietoisia, mitä eri käyttäjät ovat pyytäneet ylläpitäjää tekemään ja mitä on jo tehty (dokumentoinnin tärkeys). Yleensä pyynnöt ja toteutetut tehtävät on yksittäin tehtyjä ja niitä ei ole välttämättä kerätty tiettyyn paikkaan (dokumentoitu). *Ristiriitaisuus* kun muutostarpeiden vaativat samanaikaisesti muutoksia järjestelmään ja ne voivat olla ristiriidassa toisiinsa nähden. Tarkoittaa sitä, että aiemmin tehty muutos muuttuu tai ei toimi samalla tavalla uutta muutosta tehtäessä. *Priorisointi* eli muutoksien järjestystärkeys, jotta selvittää ilman sekaannuksia ja epätietoisuutta. *Omistajuus/vastuullisuus* siitä kenen vastuulla on tietyt tehtävät, koska kun on vastuussa jostakin, sen tekee huolellisemmin. *Puutteellinen organisointi käyttäjäpuolella* eli tehtävien vastaanotossa pitäisi olla asiantunteva henkilö, joka vastaanottaa työt. Henkilö,

jolla on tarvittava tuntemus järjestelmään ja tieto mitä ylläpitotehtäviä on tarve tehdä. *Kommunikointi käyttäjän ja tietotekniikan ammattilaisen välillä* tarkoittaa niin sanottua kielimuuria. Tietotekniikan ammattilainen ei ymmärrä riittävästi käyttäjän haluan toiminnan tavoitteita, sisältöä tai käyttäjä ei tunne alan termistöä eikä järjestelmän teknisiä ratkaisuja selvittääkseen ammattilaiselle haluamaansa muutostarvetta. Tällaisessa tapauksessa asenteet ovat tärkeitä, jolloin saavutetaan hyvä lopputulos. *Puutteellisten dokumenttien aiheuttamat lisätyöt* on ja tulee varmaan olemaan aina iäisyyskysymys, joka on ongelma järjestelmien ylläpidossa. Dokumentit ovat joko puutteellisia ja päivittämättä tai kiire on niin sanottu este dokumenttien asianmukaisen päivittämisen tekemiselle. Vaikka yrityksille on laadittu dokumentointistandardit niin jostakin syystä niitä ei noudateta riittävästi ja dokumentointi koetaan niin sanotusti ylimääräiseksi työksi tehdyn muutoksen päälle eli se vaatii ylimääräistä aikaa, jota ei useinkaan lasketa aikatauluihin. *Huonosti rakenteelliset järjestelmät ja ohjelmat* tarkoittaa että järjestelmät ovat syntyneet olosuhteiden pakottamana ja ovat joskus laajoja. Ne on suunniteltu niin, että niissä on korkea integraatioaste eli jokin muutos muuttaa muutkin kohdat ohjelmassa automaattisesti.

Nyt tästä on tullut ongelmallinen, koska järjestelmien hallinta on vaikeutunut ja dokumentoinnin puute vaikeuttaa ennestään hallintaa. Järjestelmiin on lisättyjä uusia toimintoja kiireellä, jolloin järjestelmän alkuperäinen selkeä rakenne häviää vähän kerrassaan. Uutta tekniikkaa suositellaan otettavaksi käyttöön vanhoihin järjestelmiin, mutta pitää muistaa, että se monimutkaistaa ja vaikeuttaa ylläpitoa. Järjestelmän kasvaessa ja sekoituessa lopulta käy niin, että ylläpitäjä ei tiedä, mitä mikäkin muutos vaikuttaa ja mihin muutos olisi paras tehdä, tällöin ollaan jo vaarallisella alueella ylläpidon kannalta. *Useat liittymät muihin järjestelmiin* eli liittymien määrä ei niin ole ongelma vaan standardirajapintojen puute tai niiden noudattamattomuus. *Kaikkien osapuolien huono sitoutuminen* tarkoittaa, että ylläpitoon ei suhtauduta oikealla asenteella ja sitoutuen eli arvostus puuttuu. Siihen suhtaudutaan niin sanotusti vasemmalla kädellä eli vain välttämätön tehdään ja siitä esimerkki johtuu resurssien vähäinen varaus ja ajankäyttö suunnittelussa. *Monimutkaiset tekniset ympäristöt* vaatii ylläpidolta niin vanhojen kuin uusien tekniikoiden tuntemusta ja se on melkoinen haaste vastaanotettavaksi. *Ylläpidon arvostuksen puute* tarkoittaa, että ylläpitoa ei pidetä varsinaisena työnä vaan välttämättömänä pahana. Uuden tekeminen ajaa aina vanhan ylläpidon edelle ja saa kaiken huomion, josta johtuen ylläpitäjät kokevat työnsä olevan pienien

ongelmien ratkaisemista kiireellä ilman sen suurempaa suunnitelmallisuutta. (Koistinen 2002, 49-57.)

7.1. Järjestelmän siirto ylläpitohenkilöstölle

Järjestelmän siirto toimittajalta ylläpitohenkilöstölle vaatii tiettyjä toimenpiteitä ja tähän kuuluu erilaisia tehtäviä kuten laitteiston, tiedostojen ja ennen kaikkea tietotaidon siirtoa. Näistä tietotaidon siirto vaatii aika lailla työtä, jotta se saadaan hyvin ja riittävän suunnitelmallisesti hoidettua.

Seuraavilla keinoilla se voidaan asianmukaisesti toteuttaa: *Riittävä koulutus*, jolla tarkoitetaan uuden järjestelmän sisällön ja teknisten ratkaisujen läpikäymistä. Ylläpitäjän täytyy tietää järjestelmästä mitä se tekee, sen tekninen toteutus, mitä varten järjestelmä on tehty ja kuka niitä käyttää. Koulutuksella toteutetaan suunnitelmallisesti asiantuntemuksen siirtoa eteenpäin sen kanssa toimiville. *Järjestelmien dokumentit* tarkoittaa, että toimittaja ja ylläpitäjä käyvät läpi dokumentit järjestelmästä koulutuksen kautta. Ylläpidolla on mahdollisuus niin sanotusti päästä sisään järjestelmän logiikkaan. Nämä dokumentit ovat ylläpitäjälle ohjekirja, joista selviää, miksi on päädytty tiettyyn ratkaisuun ja varsinkin, jos jotain poikkeavaa on jouduttu tekemään. *Toimittaja mukaan ylläpitoon siirtovaiheessa* tarkoittaa sitä, että järjestelmän kehitystyössä ollut henkilö siirtyy ylläpitoon mukaan, kunnes ylläpitohenkilöstö on riittävän asiantunteva ja selviää ylläpidosta omin voimin. Tämä on varmasti hyvä ja toimiva malli siirtää asiantuntemusta eteenpäin hallitusti, mutta aina ei ole mahdollista toteuttaa sitä tässä muodossa ja se tehdään pienemmässä mittakaavassa. Koulutuksen lisäksi pitäisi kehittäjän ja ylläpitäjän viettää edes muutama aika keskenään asian tiimoilta myöhemmin. Näillä tavoilla toteutettaisiin siirto mahdollisimman laadukkaasti ja suunnitelmallisesti ja loppujen lopuksi olisi kaikille parasta saada asiat kerralla kuntoon. (Koistinen 2002, 116-117.)

7.2. Järjestelmien erilaiset toteutukset

Järjestelmien toteutukselle on erilaisia vaihtoehtoja esimerkiksi tehdä järjestelmä itse, hankkia valmisohjelmisto tai räätälöity ohjelmisto, vuokrata sovellus. Tietotekniikan alkuajoilta 1960-luvusta 1990-luvulle itse tehtyjä järjestelmiä käytettiin ja myös yllä-

pidettiin niitä. 1970-luvulta alkaen tulivat valmiit järjestelmät usein pakettiratkaisuna, esimerkiksi palkkalaskennan järjestelmät. Suurin piirtein samaan aikaan tulivat myös räätälöidyt järjestelmäratkaisut ja niillä ratkaistiin yrityksen järjestelmänkehitystarpeet, joihin yrityksen omalla henkilöstöllä ei ollut aikaa eikä resursseja tehdä. Vasta 2000-luvulla tulivat muutoksen tuulet järjestelmäympäristöön, sovellusvuokrausmallit. Siinä sovellus vuokrataan toimittajalta ja he vastaavat ylläpidosta. Toisaalta asiakkaalla ei ole mahdollisuutta vaikuttaa erilaisten omien erityispiirteiden saamiseen järjestelmään mukaan ja se on tärkeää yksityiskohta vuokrattaessa sovellusta. Mahdollista on myös, että erilaiset toteutusmallit sekoittuvat keskenään eli on omia ratkaisuja, valmiita tai räätälöityjä ja vuokrattuja osia. Tämä on varmaan käytännössä vaativa ratkaisu asiakkaalle hallita, koska muutos yhteen sovellukseen saattaa vaikuttaa moneen muuhun sovellukseen. (Koistinen 2002, 197-198.)

7.3. Ylläpidon kehittäminen

Ylläpidolla on haasteellinen tulevaisuus tietotekniikan ja muiden teknisten ratkaisujen edessä. Ylläpitokin on kokenut muutoksien tuulia jakamalla yleisen ylläpidon eri toimintojen alueisiin esimerkiksi laitteistot ja järjestelmät. Koska tietotekniikka valtaa yritysmaailmasta suuremman palan on se myös yritysten liiketoimintayksiköiden huomioitava. Ylläpidon ja liiketoimintayksiköiden vahva panostus mahdollistaa ensinnäkin yrityksen toimintakunnon ja toiseksi järjestelmien hyvän toimivuuden. (Koistinen 2002, 211.)

7.3.1. Sovellukset

Tietotekniikan ammattihenkilöt pitävät heidän ylläpitotyötään sovellushallintana ja sovellushallinta sisältää myös niin sanotun HelpDesk-toiminnon. Tarkemmin, sovellushallinta sisältää sovellus virheiden korjaukset, muutokset ja uusien toimintojen lisäykset. Sovellustuen (HelpDesk) merkitys kasvaa, koska toimintoja ulkoistetaan ja sovellushallinta kattaa resurssien hallinnan, käytön ja laadun valvonnan. Sovellus hallintaa voidaan kuvata seuraavasti, *Korjaava ylläpito* on virheiden korjaus ja ennakoiva tuki. *Kehittävä ylläpito* tarkoittaa, että se tehtävänä on täydentävät, mukauttavat tai kehittävät muutokset uuteen tai olemassa olevaan sovellukseen. *Käyttäjätuki* avulla tuetaan käyttäjiä ja ongelmien hallintaa. *Käytön hallinta* sisältää huoltosuunnitelmat,

valvonnan, raportoinnin, resurssien varauksen ylläpitoon ja laadun valvonnan. Kokonaisuutena sovellus hallinnalla tarkoittaa infrastruktuurin eli laitteiston ja tietoliikenteen ylläpitoa. (Koistinen 2002, 212.)

7.3.2. Vanhat järjestelmät

Tietotekniikan nopea kehittyminen vuosien varrella on asettanut ylläpidon tietyn kysymyksen eteen, mistä tulevaisuudessa löytyy myös vanhoihin järjestelmiin asiantuntevaa ylläpitäjää? Tämä kysymys siksi, että uutta tekniikkaa tulee jatkuvasti mutta vanhat eivät katoa kokonaan. Esimerkiksi teollisuudessa on todella suuria tietojärjestelmiä erilaisissa toiminnoissa ja modernisointi tapahtuu usein osakokonaisuuksina, jolloin vanhaa järjestelmää jää johonkin osaan kokonaisuutta. Lisäksi, vanhojen järjestelmien ylläpito henkilöityy eli joku tietty ylläpitohenkilö hallitsee vanhan järjestelmän ja hänelle lankeaa sen ylläpito, jolloin huomaamatta ylläpito alkaa henkilöityä. Henkilön poistuessa työelämästä eläkkeelle tai vaihtaessa työpaikkaa tietotaito vanhojen järjestelmien ylläpidossa vaikeutuu ja pahimmassa tapauksessa häviää. Siksi pitäisi välttää ylläpitotehtävän tai -alueen jättämistä vain yhden ihmisen varaan. (Koistinen 2002, 217.)

7.4. Automaatiojärjestelmien ja toimistojärjestelmien eroja

Tietoturvan kokemukset automaatiojärjestelmissä perustuvat olemassa olevien toimistojärjestelmien tietoturvaan ja käytäntöihin, mutta automaatiojärjestelmiltä vaaditaan kuitenkin korkeampaa tietoturvasoaa. Automaatiojärjestelmien laitteet tunnetaan paremmin ja niiden käyttö on pidempi aikaista ja ennen kaikkea muutoksia on vähemmän kuin toimistojärjestelmissä, koska automaatiojärjestelmien muutokset asettuvat tavallisesti kunnossapitotöiden aikaan. Automaatiojärjestelmissä ei ole liiketoiminnan kannalta salattavaa tietoa, eikä suoraa internetyhteyttä tarvita. Kuitenkin automaatiojärjestelmiin pääsy on tarkasti määritetty ja henkilöstö koulutettu niiden käyttöön. Teollisuuden automaatiosta puhuttaessa on muista tietotekniikoista eroavia piirteitä. Tuotteiden laadun ja tuotantotoiminnan ylläpitäminen, ihmisten turvallisuuden varmistaminen ja laitteistojen ja tehdaslaitosten suojaaminen voidaan pitää automaation tavoitteina. Toimistojärjestelmät automaatiossa suojelevat palvelimille tallennettuja tietoja ja tuotantojärjestelmät vastaavat toiminnallisten yksikköjen oikeasta toiminnasta. Kum-

matkin järjestelmät ovat hajautettuun suuntaan menossa, koska tekniikka ja verkottuneet liiketoimintamallit kehittyvät siihen suuntaan. Toimistojärjestelmissä ei toivotut seuraukset eivät ole niin vakavia tuotannollisesti ajatellen, mutta tuotantojärjestelmissä ne ovat vahingollisia, jopa tuhoisia. Ohjelmistot tuotantojärjestelmissä ovat monimutkaisempia kuin tietoteknisissä järjestelmissä ja tuotantojärjestelmien ylläpito vaatii paljon niin prosessin kuin laitteiden asiantuntemusta. Automaatiojärjestelmien ja toisaikajärjestelmien resurssit ovat rajoittuneet, eikä niissä voida käyttää tyypillisiä tietoturvateknologioita. Tieto on tärkeää ohjausjärjestelmille ja se täytyy vikatilanteiden takia saada eheytettyä. Ohjausjärjestelmissä protokollat ja tiedonsiirtotavat ovat erilaiset yleisiin tietoteknisiin ympäristöihin verrattuna. Ohjelmistojen ja laitteiden päivittäminen on vaativaa, koska on tunnettava automaatioosovellus ja –järjestelmän toiminta ja käytettävien ohjelmistojen vaikutus toisiinsa. Korjaus- ja ylläpito-ohjelmien päivitys ei hoidu käden käänteessä, vaan automaatiojärjestelmän toimittaja testaa ne ensin, jotta selvittäisiin myöhemmin ilman suurempia ongelmia. (Suomen Automaatioseura ry 2005, 60-62.)

7.4.1. Koventaminen

Koventaminen on tiettyjen toimintojen poisto tai käytön estäminen. Tällaisia toimintoja, jota esimerkiksi automaatiojärjestelmässä ei tarvita voi olla ohjelmistoissa, kokonaisuuden perusominaisuuksissa, palveluissa ja joissakin niiden osuuksissa. Koventamisella tehdään muutoksia järjestelmään vaikeuttamaan tiettyjen ominaisuuksien väärinkäyttöä. PC-työasemissa ja –palvelimissa koventaminen on yleistä, koska niiden käyttöjärjestelmissä on laajoja ominaisuuksia ja niistä poistetaan tai suljetaan päätoiminnalle tarpeettomat osuudet kuten ohjelmistot, käyttöjärjestelmän palvelut tai muut sellaiset. Yleisimmin juuri Microsoft Windows-käyttöjärjestelmissä tämä on ollut ongelma, koska sen oletusarvona on paljon toiminnallisuuksia aktiivisena ja normaalisti ainakaan automaatioissa näitä ei tarvita ja ne saattavat hidastaa toimintaa. Itse automaatioverkko on myös koventamisen kohde ja periaatteena automaatioverkon koventamisessa on rajapinnan liikenteen ja sinne pääsyn rajoittaminen sekä verkkolaitteiden häiriösietoisuuden rakentaminen. Koventamistoimenpiteet pitää tehdä ennen käyttöönottoa, joka onkin normaali toimenpide automaatiojärjestelmän toimitusvaiheessa.

Uusiin järjestelmiin on kyllä ratkaisuja ja palveluja, mutta ongelman aiheuttavat

yleensä vanhemmat järjestelmät, koska valtaosahan käytössä olevista järjestelmistä on vanhempaa uusimpien tullessa markkinoille. Jotta kaikki toteutuu niin kuin pitää, on oltava suunnitelmallinen, tarkkaavainen korjaavia toimenpiteitä tehtäessä ja huolehdittava toimenpiteiden toimeenpanosta kokonaisuuden pysyen hallinnassa. Jotta koventamiselle luodaan onnistumisen mahdollisuus, tulee huomioida seuraavat seikat: automaatiojärjestelmä pitää koventaa ennen verkkoon tai sisäverkkoon kytkemistä, käyttäjällä ja laitteella on vain toiminnallisuudelle välttämättömät oikeudet peruskonfiguroinnissa, mutta koventaminen ei kuitenkaan saa häiritä ajettavien ohjelmistojen toimintaa. Koventaminen käytännön työssä vaatii korkeaa tasoa ohjelmistojen toiminnan asiantuntemuksessa, koska koventajan pitää tietää, mitä tarvitaan tai mitä ei tarvita jotakin tiettyä järjestelmää rakennettaessa, jotta tarpeettomat palvelut saadaan pois päältä. Seuraavassa esimerkki koventamisesta: Kovenna käynnistys (liikkuvat tallennusvälineet CD, USB), palvelut (tiedostojaot, www-palvelut), paikallinen levyjärjestelmä (käyttäjällä pääsy vain tarpeellisiin tietoihin), aseta levytilan käytölle rajat ja valvonta, aseta pakotettu käyttöoikeuksien valvonta, päivitä järjestelmän ohjelmistot ja käyttöjärjestelmän tietoturva. Eihän käyttäjä aina tiedosta tekevänsä väärin joissakin tilanteissa, joten heidän tietämystasoaan on myös pidettävä yllä riskien välttämiseksi. (Suomen Automaatioseura ry 2005, 73-75.)

8. PÄÄTELMÄ

Automaatiojärjestelmän suunnittelussa ja rakentamisessa ja ylläpitämisessä vaaditaan eri osapuolien ennakkoluulotonta suhtautumista toisiinsa. Automaatiojärjestelmän elinkaareen liittyy niin monia vivahteita ja jokaisen osapuolen panostus elinkaaren edetessä on erittäin tärkeää. Kaikki lähtee esisuunnitteluvaiheesta, jossa pitäisi jo olla RIITTÄVÄ JA SELKEÄ kuvaus toiminnasta ja siitä, mitä halutaan. Onnistumisen perusta valetaan alkuvaiheessa, jota ei aina painoteta riittävästi ja kustannusnäkökulmasta. Mitä aikaisemmin asiat ja tehtävät ovat hallussa, sitä vähemmän kustannuksia kokonaisjärjestelmän rakentaminen ja etenkin ylläpito vaatii. MetsoDNA:aan AutoCAD ja Windows järjestelmien elinkaari etenee nopealla syklillä aina uusimman järjestelmän hiljalleen integroituen vanhan rinnalle ja lopulta syrjäyttäen sen. On tapauksia, jossa uuden järjestelmän pitää sopia yhteen vanhemman järjestelmän kanssa. Koska on vanhoja järjestelmiä, joista ei tiedetä tarkkaan, miten ne reagoivat, jos järjestelmä päivitetään ja se kuulostaa oudolta. Jokin osajärjestelmä toimii vanhalla järjestelmällä,

mutta toiminta on sen verran hämärtynyt, että ei haluta koskea. Tästä varoituksen sanan sanoin aiemmin, että silloin ollaan vaarallisilla vesillä ja dokumentointi olisi tällaisessa tapauksissa ollut elintärkeää, koska dokumentoinnista olisi selvinnyt miten toimii, mitä muutoksia tehty. Selkeä dokumentointi suunnittelussa auttaa kaikkia osapuolia, kun joudutaan jonkin ongelman takia tutkimaan, mitä tai miksi ongelma syntyi.

Ohjelmistopuolella maallikolle on vaikeaselkoista, kun ohjelmistoon mennään niin sanotusti sisälle, mutta silti asiantuntijan ja maallikon yhteisymmärrys ohjelman tai sovelluksen toiminnasta eivät aina kohtaa. Maallikko tai asiakas ei ymmärrä, miksi sovellusta ei saada toimimaan juuri niin kuin haluaisi tai miksi se on tehty niin hankalaksi. Asiantuntija saattaa sanoa, että jokin toiminto tai virhe sovelluksessa on ominaisuus, kun taas asiakas näkee sen selvänä virheenä. Tässäkin suunnittelun ja rakentamisen tärkeys korostuu, koska sovellusta ei voida saada toimimaan asiakasta tyydyttävällä tavalla. Syinä tässä saattaa olla joko asiakkaan huonoa toimeksianto tai muutokset matkan varrella eli alkuperäinen tehtävänanto muuttunut tai suunnittelijan huono tietotaito suunniteltavan järjestelmän tai sovelluksen toiminnasta ja toimintaympäristöstä. Lähinnä ongelma on, mitä asioita painotetaan elinkaaren aikana ja onko suurin panostus elinkaaren alkuun vai vasta ylläpitovaiheeseen. Kustannuskehityksen näkökulmasta alkuun kannattaisi panostaa, koska aina vähäisemmäksi työpanoksen määrä jää elinkaaren edetessä.

Automaatiojärjestelmät ovat nykypäivänä tietokonepohjaisia järjestelmiä ja hyvin monien erilaisten sovellusten sekamelska. Niiden keskenään sujuva toiminta tai nopea tietotekniikan kehityksen luoma päivityspaine, jolla korjataan virheitä tai tuodaan uusia ominaisuuksia, asettavat haasteita. Ohjelmistopuolella juuri kehityksen nopea eteneminen on syynä suunnittelun ajankäytön niukkuuteen ja tämä johtaa siihen, että suunnittelutyötä tehtäessä niin sanotusti toisarvoisiin asioihin ei kiinnitetä riittävästi huomiota. Nämä toisarvoiset asiat saattavat myöhemmin aiheuttaa ongelman, mutta se ei ole aina suunnittelijan vika, koska hän tietenkin tekee parhaansa aikataulupaineen alla. Tärkeimmät teesit ovat aika, toimintaympäristön tuntemus, yhteistyö, dokumentointi.

LÄHTEET

Conger, S.A. 1994. The New Software Engineering. Belmont, California: Wadsworth Publishing Company

Cordata, J.W. 1983. Managing DP hardware: Capacity Planning, Cost Justification, Availability and Energy Management. New Jersey: Prentice-Hall, Inc., Englewood Cliffs.

Haikala I., Järvinen H-M. 2003. Käyttöjärjestelmät. Jyväskylä: Talentum Media Oy

Haikala I., Märijärvi J. 2001. Ohjelmistotuotanto 7. painos. Pieksämäki: Talentum Media Oy

Jaakohuhta, H. 2000. Lähiverkot-Ethernet. Jyväskylä: Oy Edita Ab

Kiviniemi, T. 2011. MetsoDNA-Service Sales Guide to the compatibility between activities in C2011 upgrades. Muistio 30.9.2011.)

Koistinen, H. 2002. Tietojärjestelmien ylläpito. Jyväskylä: Talentum Media Oy.

Kosonen, J. 2012. Automaatiojärjestelmän elinkaari ja päivitys. Insinööriyö. Kotka: Kymenlaakson ammattikorkeakoulu.

Kymenlaakson ammattikorkeakoulun laboratoriomateriaali. Metso DNA Manuals. Collection 2011 Fi V.14.1 build 1

Metso Automation Oy:n Intranetsivut. Metso DNA CR Manuals, Collection 2010 Fi V.13.1 build 2.1. Saatavissa: <http://tres500093/eman/dna/ fi/index.htm> (viitattu 27.5.2013)

Metso Oyj. Feel the case. User interface. Saatavissa: http://metso.com/Automation/ip_prod.nsf/WebWID/WTB-070115-2256F-E4E86?OpenDocument&mid=352E8BE4B3EA7311C22575BD0002A80E3 (viitattu 27.5.2013)

Metso Oyj. The hard facts for a good feeling. Saatavissa:

http://metso.com/Automation/ip_prod.nsf/WebWID/WTB-070115-2256F-91897?OpenDocument&mid=6824C2C589A078AAC22575BE002E1253 (viitattu 27.5.2013)

Metso Oyj. Feel the case of one system. Saatavissa:

http://metso.com/Automation/ip_prod.nsf/WebWID/WTB-070111-2256f-8331B?OpenDocument&mid=0DFEA18DC2D983DB22575C100423643 (viitattu 27.5.2013)

Metso Oyj, High availability. Saatavissa:

http://metso.com/Automation/ip_prod.nsf/WebWID/WTB-070118-2256F-E7280?OpenDocument&mid=5A67F920399F3C23C22575C100434EA (viitattu 27.5.2013)

Metso Oyj. True scalability. Saatavissa:

http://metso.com/Automation/ip_prod.nsf/WebWID/WTB-070118-2256F-C62A?OpenDocument&mid=C201F767C9D0E8B3C22575C100454107 (viitattu 27.5.2013)

Pahlama J. 2009. Automaation tietokoneverkot: Ethernet/IEEE 802.3(x) -lähiverkot. Luennot syksy 2009. Kotka: Kymenlaakson ammattikorkeakoulu.

Pahlama J. 2009,1. Automaation tietokoneverkot: Yleisiä tietoverkkoihin liittyviä käsitteitä. Luennot syksy 2009. Kotka: Kymenlaakson ammattikorkeakoulu.

Paperi ja puu. Automaation MetsoDNA: Kattavampaa käyttäjähallintaa. Saatavissa: <http://www.paperijapuu.fi/749/> [viitattu 14.5.2012]

Peltola, J. 2002. UUDET AUTOMAATIOJÄRJESTELMÄT-Komponenttipohjaisen automaatiosovelluksen suoritusympäristö. Teknillinen korkeakoulu, automaation tietotekniikan laboratorio. Saatavissa:

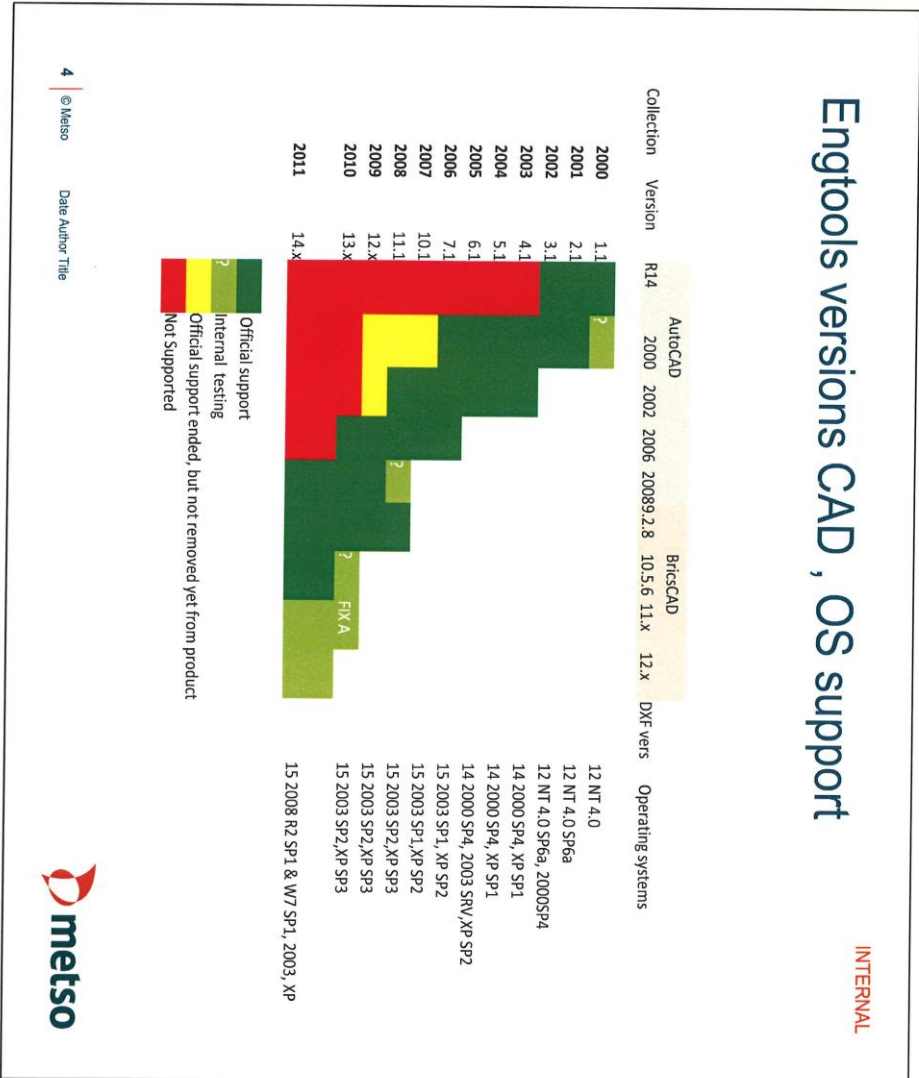
http://www.automationit.hut.fi/julkaisut/documents/peltola_uudet_automatiojärjestelmät_raportti_6.pdf

Suomen Automaatioseura ry: Turvallisuusjaosto 2005. Teollisuusautomaation tietoturva: Verkottumisen riskit ja niiden hallinta. Helsinki: Painomerkki Oy.

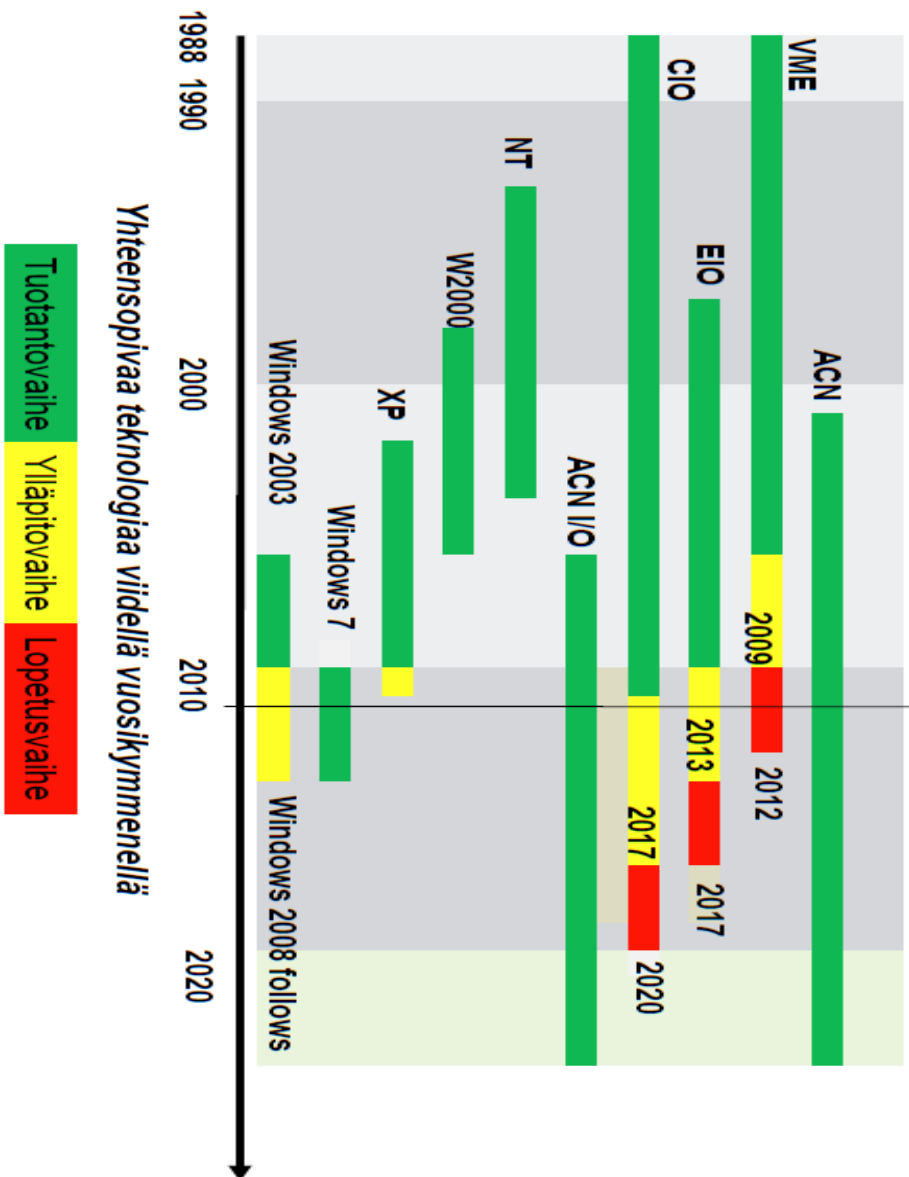
Tuomi P. Oulun seudun ammattikorkeakoulu 2006. Automaatiotekniikka: Automaatiotekniikka 1. Saatavissa: http://www.tekniikka.oamk.fi/~terohi/auto1_s2006u.htm
(Viitattu 23.3.2013)

Ylikunnari, J. 2003. Oppimateriaali kurssiin TL6031 Automaatiojärjestelmät. Oulun seudun ammattikorkeakoulu: tekniikan yksikkö.

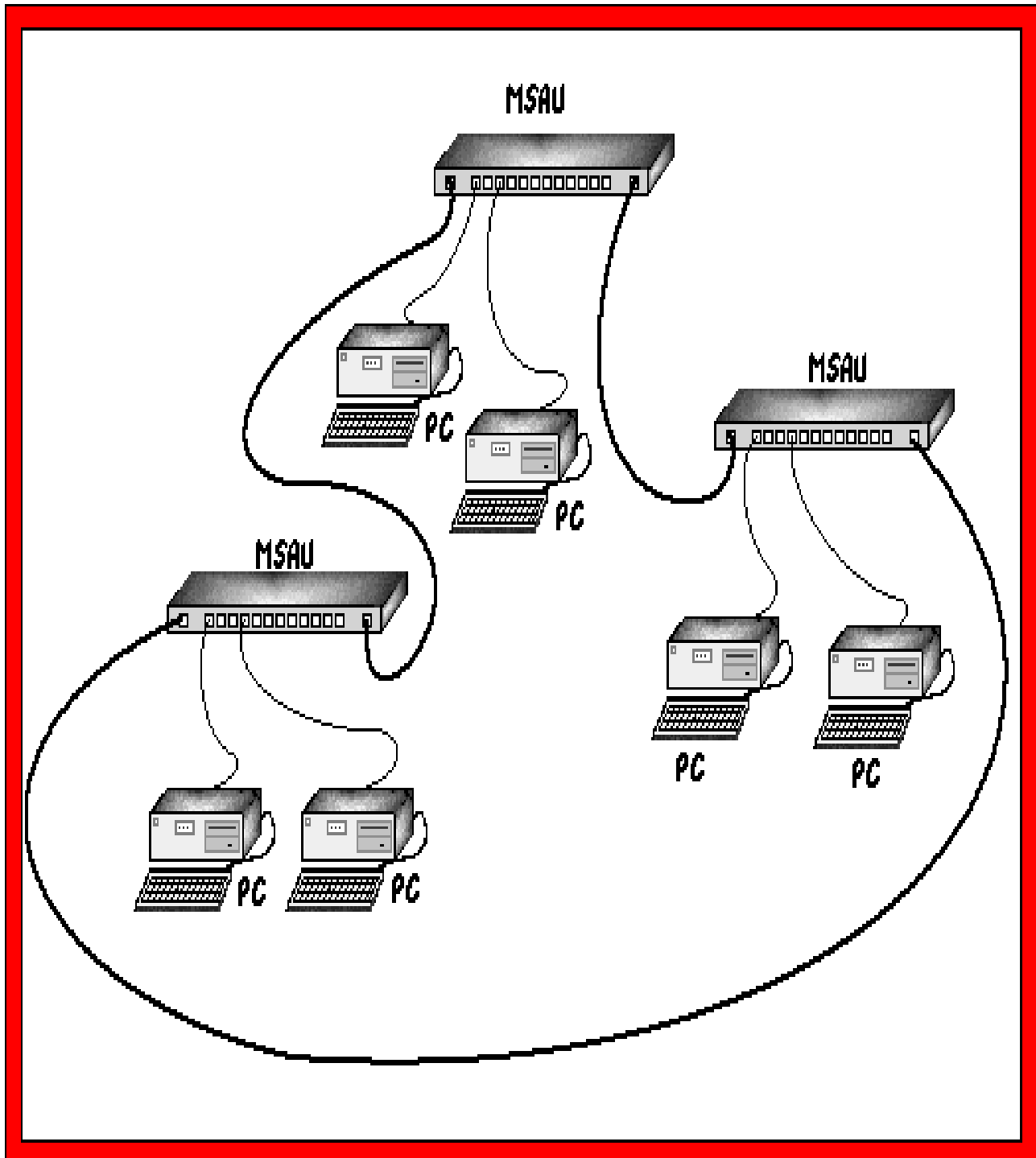
Ylinen, M. Metso DNA CR 2010-2011 news. Asiakasseminaariesitys Tallinna
11.3.2011



Liite 1. Metson AutoCAD- ja käyttöjärjestelmä tuki



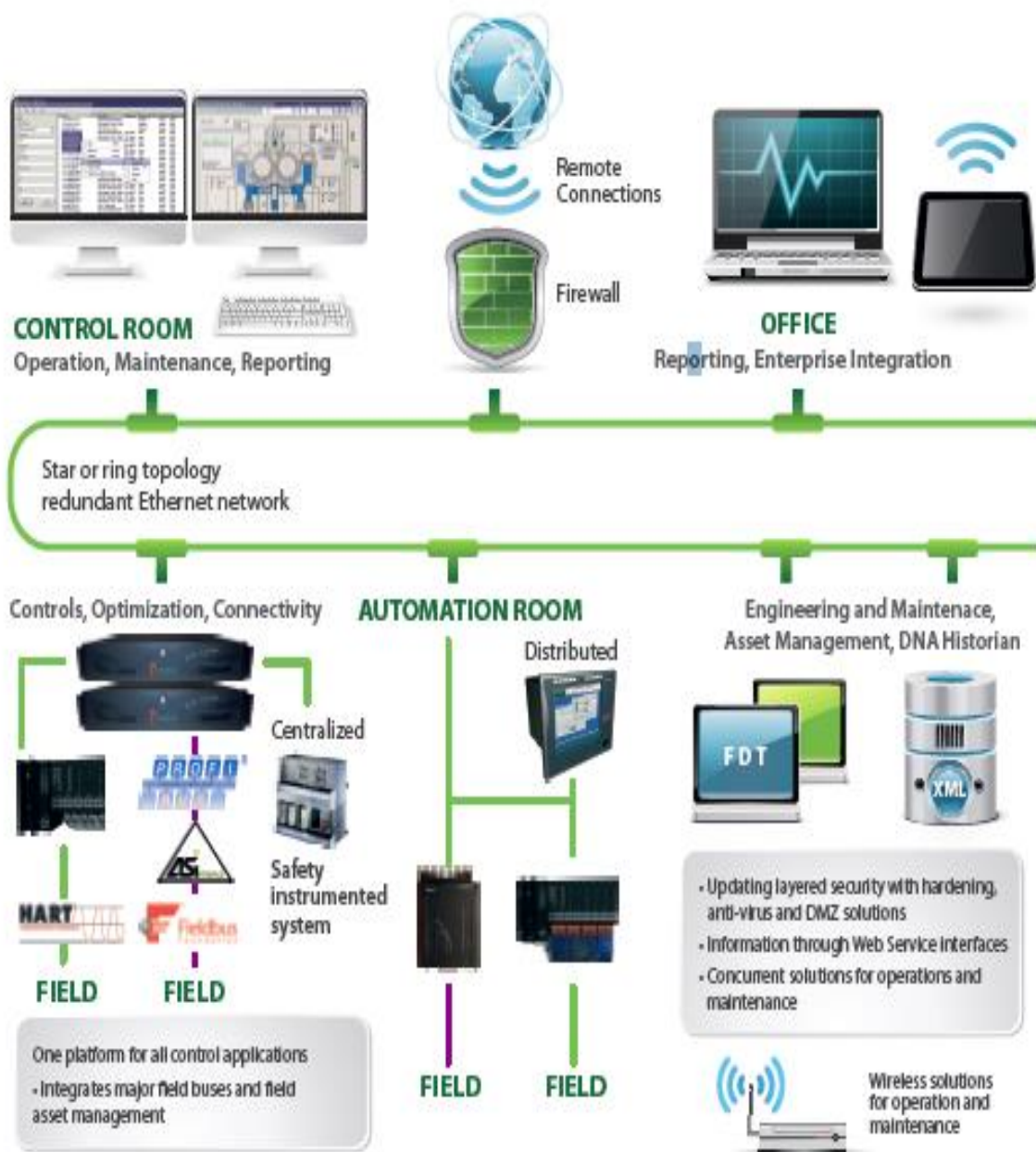
Liite 2. Metson automaatiojärjestelmätekniikan elinkaari (Kosonen 2012. Automaatiojärjestelmän elinkaari ja päivitys)



Liite 3. Token Ring fyysinen topologia (<http://www.maznets.com/tech/token.htm>)



Liite 4. Automaatiojärjestelmän kehitys Metso (Kosonen, Jorma 2012. Automaatiojärjestelmän elinkaari ja päivitys)



Liite 5. Metso DNA arkkitehtuuri (Kosonen 2012. Automaatiojärjestelmän elinkaari ja päivitys)