

Softari

NOKIA DATAN OHJELMISTOLEHTI • 3/1990



Päätoimittaja:
Terhi Ogbeide,
puh. (90) 567 2994
smail: Ogbeide__Terhi__NOK

Toimittaja:
Jyrki J. J. Kasvi,
puh. (90) 567 4064
smail: Kasvi__Jyrki__FRL

SoftariTiimi:
Mikko Helander, Kari Kaipainen,
Outi Kuusela, Rami Raulas,
Tapio Sulkava, Esko Uotila

Ulkoasu:
Paula Jaakkola

Osoitteenmuutokset:
Takakannen palvelukortilla

Toimituksen osoite:
Nokia Data/Softari
PL 780
00101 HELSINKI

Painosmäärä:
13 000

Kirjapaino:
Helsingin Liikekirjapaino Oy

Julkaisija:
Nokia Data

ISSN 0782-8209

Sisällys:

Kansi: Ilkka Kumpunen	
Unix — avoin kysymys	3
Mikko 100 — avoin mutta turvallinen	6
Sarlin uskoo Unixiin	8
Maatalous yhdistyy Unixilla	10
Tandem Integrity S2	12
Tiimikko-talo ja tuote	15
MikroMikko 4 m458	18
Uutisia	19
Hyöty irti ohjelmistoista	22
Nitec kouluttaa Unixiin	23
Pelkkä helppi ei aina riitä	24
Olioita ohjelmiin	28

Harrastuksesta bisnes

”**U**skonto ja hauska harrastus, jolla ei ole rahan ansaitsemisen kanssa juurikaan tekemistä”, luonnehdittiin Unixia vielä muutama vuosi sitten. Nyt luja usko ja harrastelu on alkanut tuottaa. Unix kypsyy.

Olemme koonneet siitä käytännön esimerkkejä tähän lehteen. Uusimpien kotimaisten tutkimusten mukaan Unix on pk-yritysten kokonaisjärjestelmissä hallitseva käyttöjärjestelmä.

Tietojärjestelmää valitsevaa helpottaa se, että käyttäjiä ei enää aseteta joko-tai-asetelman eteen. Unix-järjestelmä voidaan rakentaa monen käyttäjän järjestelmäksi, mutta haluttaessa työasemissa voidaan käyttää lähiverkonkin sovelluksia.

Ja eihän se Unixin suureneva suosio mikään ihme ole. Iki aikojen ihminen on yrittänyt tehdä totta unelmistaan. Tietotekniikan tekijät ovat haaveilleet siirrettävästä käyttöjärjestelmästä ja ohjelmointiympäristöstä. Käyttäjät kaikkein vähiten ovat halunneet sitoutua laitemerkkeihin. He ovat toivoneet ohjelmistojen siirrettävyyttä ja tietokoneiden vaivatonta keskinäistä viestintää.

Tietotekniikkaa tuntemattomalle Unix on vaikea. Lopputyö käyttäjän ei kuitenkaan tarvitse olla Unix-asiantuntija. Jos asialla ovat hyvät ohjelmistontekijät, Unixia ei loppukäyttäjä edes huomaa.



— avoin kysymys

Tuskin mikään tietojenkäsittelyn alue on ollut sellaisten intohimojen kohde kuin Unix-käyttöjärjestelmä. Nyt täysi-ikäisyyden kynnyksellä Unix näyttää löytävän paikkansa. Avoimuuden julistaminen on muodostunut niin tietojärjestelmien toimittajille kuin hyväksikäyttäjillekin lähes uskonnokappaleeksi. Vaikka Unix ja avoimuus eivät maailman ongelmia ratkaisisikaan, ne luovat yhdessä käyttäjille huomattavasti valoisamman tulevaisuudennäkymän kuin aiemmin on kuviteltu.

Unixin asemaa yhtenä käyttöjärjestelmänä muiden joukossa on vaikeuttanut sen akateeminen luonne. Unix ei ole yksi vaan monta. Vaikka AT&T, jonka laboratorioista Unix 1970-luvun alussa lähti maailmalle, vahtiikin mustasukkaisesti tavaramerkkinsä käyttöä, löytyy maailmasta kymmenittäin muita samasta lähteestä peräisin olevia käyttöjärjestelmiä, joilla on jokin muu nimi.

Puristeille on aina kelvannut vain aito tavara. Monia ”vieraita” käyttöjärjestelmistä vierokсутaan ja väheksytään, vaikka ne todellisuudessa ovat suorita johdoksia ja varsin yhteneviä alkuperäisen Unixin kanssa. Itse asiassa näistä muista tuotteista löytyvät yleensä ne piirteet, joilla Unix on saanut hyväksyntänsä käyttäjien keskuudessa: kehittynyt käyttöliittymä, järjestelmän hallintatyökalut ja tehokkaat varus- ja apuohjelmat.

Ainoa tie monille

Ironista kyllä, Unixin hovikelpoisuus tuli todistetuksi vasta,

kun IBM julkisti viimeisten joukossa oman rajoitetun käskykannan suoritinarkkitehtuuriin (Risc) perustuvan Unixin kaltaisen AIX-perheensä ja julisti, että siinä on tulevaisuus. Sitä ennen ovat lukuisat muut järjestelmävalmistajat ennättäneet siirtää kehityspanoksensa Unixin taakse, koska se nähdään pienten ja keskisuurten järjestelmätoimittajien ainoana tienä.

Tietojärjestelmämarkkinoiden kiristytvä kilpailu edellyttää lisää ominaisuuksia, parempaa suorituskykyä ja laatu-hintasuhdetta. Tämän toteuttaminen kilpailukykyisesti yhden valmistajan suljetussa ympäristössä saat- taan käydä valmistajalle taloudellisesti ylivoimaiseksi.

Unixin kehittyvä avoimuus on puolestaan tuottanut muutaman laitearkkitehtuuriin ympärille kohtuullisin ponnistuksin ylläpidettävän ja jatkokehitettävän ”yhteensopivuusmaailman”, joka tarjoaa entisille suljettujen järjestelmien toimittajille pääsyn avoimille markkinoille ilman valtavaa peruskehityspanosta. Tällä ”buy-instead-of-make”-strategialla toimii nykyään val-

taosa Unix-järjestelmätoimittajista. Unixin kehittämisen painopisteen voidaankin sanoa siirtyneen käyttöjärjestelmän kehittämisestä markkinoilla olevan puolenkymmenen merkittävän Unix-toteutuksen tehokkaaseen hyödyntämiseen laitearkkitehtuureja jalostamalla ja lisäarvoja tuottamalla.

Yli leirirajojen

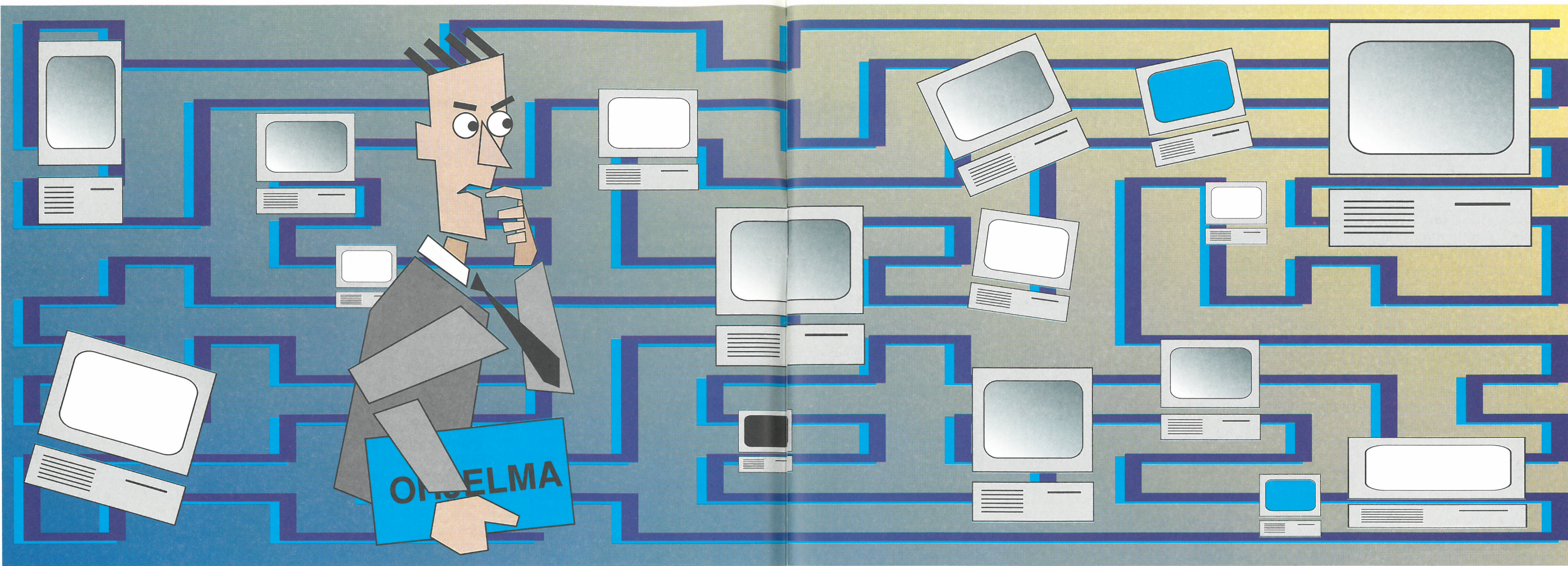
Viime vuosien näkyvin Unix-piirejä jakava tekijä on ollut toimittajien leiriytyminen kahdeksi vastakkaiseksi yhteenliittymäksi. Kilpailevat intressipiirit, Open Software Foundation (OSF) ja Unix International Inc. (UII), ovat molemmat pyrkineet tuottamaan standardituotteen, joka sisältää yhtenäisen käyttöjärjestelmän ja käyttöliittymän. Kummankaan osapuolen tuote ei kuitenkaan ole saanut markkinoilla selvästi hallitsevaa asemaa.

Vaikuttaakin siltä, että näiden leirien vaikutus on jatkossa vähennemässä, sillä useita ns. strategisia yhteenliittymiä ja teknologiasopimuksia on solmittu yksit-

täisten toimittajien ja kehittäjien välille yli leirirajojen. Todennäköinen seuraus on, että vaikka yhtä yhtenäisjärjestelmää ei luotaisikaan, merkittävimmät toimittajat tukevat molemmissa leireissä tuotettujen järjestelmien parhaita piirteitä.

Hyväksikäyttäjien kannalta mielenkiintoisempaa kuin kahden kiistelevän yhteenliittymän kannanotot on se yli rajojen tehtävä määritystyö, jota vetää X/Open Company, Ltd. X/Openiin kuuluvat käytännöllisesti katsoen kaikki Unix-järjestelmätoimittajat sekä joukko ohjelmistotoimittajia. X/Openin tavoitteena on määritellä kriteerit ja testausohjelmat, jotka takaisivat mahdollisimman hyvän sovellusten siirrettävyyden eri järjestelmien välillä. Testit eivät kata vain laitteita, vaan myös sovelluksille asetetaan vaatimuksia.

Silloin kun tuote täyttää X/Openin vaatimukset, se saa ns. X/Open Branding -merkin, jonka tunnuksena tuotteessa on oikeus käyttää X/Open-tavaramerkkiä. Tämäkään merkintä ei tarkoita, että sovellukset voitaisiin sellaisenaan siirtää lait-



Ilkka Kumpunen.

Sovelluskehitysympäristön huolellinen valinta takaa laiteriippumattomuuden paremmin kuin järjestelmien teknisten ominaisuuksien ja kasvatettavuuden vertailu.

teesta toiseen. Se takaa kuitenkin, että sovellus voidaan hankkia useaan erimerkkiseen järjestelmään ja saattaa toimimaan yhtenevästi niissä kaikissa.

Entä Suomessa?

Mikä sitten on Unixin asema Suomessa? Unix levisi kaupallisille sovellusmarkkinoille minikoneiden alueelta, koska lähiverkko ei ole vielä kyennyt tarjoamaan hintakilpailukykyistä ympäristöä yritysmaailman operatiivisille monen käyttäjän sovelluksille.

Unixin ominta aluetta ovat tällä hetkellä pienten ja keski suurten yritysten kokonaisjärjestelmät, joissa se on uusimpien kotimaisten tutkimusten mukaan hallitseva käyttöympäristö. Näitä päätekeittäjiä sovellusalueita täydentää yrityksissä yhä useammin samassa Unix-palvelimessa suoritettava, osin hajautettu tekninen tai suunnittelusovellus, joka hyödyntää lähiverkon ominaisuuksia hajautuksensa tai nopeana siirtotienä.

Sovelluskehitys näyttää kulke-

van vähitellen siihen suuntaan, että älykkäitä työasemia hyödynetään paremmin. Välivaiheessa ns. X-päätteet löytänevät sijansa ainakin joillakin markkinoilla.

X-päätte eroaa tavallisesta näyttöpäätteestä siten, että se liittyy suoraan lähiverkkoon ja siinä suoritetaan sovelluksen koko näytönkäsitely ns. X-protokollin, jotka määrittelevät palvelimen ja päätteiden välisen liikennöinnin. X-päätte voi nopean siirtotien ja melko suuren tietojenkäsittelykapasiteettinsa ansiosta tarjota käyttäjälle kehittyneen graafisen käyttöliittymän ja usean sovelluksen samanaikaisen käytön. Hinnaltaan se on edullisempi kuin MS-Dos-, OS/2- tai Unix-pohjainen työasema, jotka myös kykenevät samaan palveluun.

Risc lyö kiilaa

Unix työaseman käyttöympäristönä herättää edelleen voimakasta ristivetoa. Unix-työasemat ovat olleet viimeaikoihin asti auttamattoman kalliita korvataksien mikrotietokoneet, mutta usean eri laitetoimittajan äsket-

täin julkistamat Risc-työasemat murtavat mikrotietokoneiden aseman ainakin hinnallisesti.

Kun hinta ei enää ole este, katseet kääntyvät työasemien palvelukykyyn. Lähiaikojen mustia hevosia työasemamarkkinoilla saattavatkin yllättäen olla mikroissa käytettyyn Intel-tekniikkaan perustuvat Unix-työasemat, joissa yhdistyvät kyky suorittaa suosittuja MS-Dos-sovelluksia, Unixin moniajo-ominaisuus sekä kehittynyt käyttöliittymä.

Rinnan näiden kanssa puristavat eri Risc-ratkaisut mikroja yhä ahtaammalle. Ei kuitenkaan ole syytä olettaa, että Unix-työasema syrjäyttäisi mikroa lähiverkossa, koska asennettu mikrotietokonekanta vaihtuu hitaasti.

Kumpi parempi?

Avoimuutta korostavassa järjestelmätarjonnassa kiinnittyy päähuomio helposti edelleen tarjolla olevien järjestelmien teknisiin ominaisuuksiin, joita kaikki tarjoajat korostavat. Näillä megatavuilla, megahertzeillä, mipseil-

lä ja flopseilla on käyttäjän kannalta merkitystä vain käytettävän sovelluksen toimintaa arvioitaessa. Koska jokainen sovellus käyttää järjestelmän resursseja omalla tavallaan, järjestelmien vertailu ja paremmuusjärjestyskseen asettaminen ei tahdo onnistua kirjoituspöytätyönä.

Laitevalintaa tärkeämpää olisi sikin selvittää, miten sovellus on siirrettävissä tarjolla olevien järjestelmien kesken. Olennaista on sovelluksen kehitysympäristö eli ohjelmankehitysohjelmat ja tietokannat, joilla sovellus on laadittu.

Unix-järjestelmätöimittajien yhteinen piirre on, että merkittävimmät työkalut ja tietokannat ovat jokseenkin kaikkiin järjestelmiin vapaasti ostettavissa — toisin kuin suljetuissa valmistajakohtaisissa ratkaisuissa, joissa laitevalinta usein sitoo käyttäjän tiettyä työkaluratkaisua. Käyttäjän kannalta tämä piirre on kenties merkittävin avoimuuden tuntomerkki, koska Unixista puhuttaessa avoimuus ei todennäköisesti koskaan voi tarkoittaa sovelluksen täyttä siirrettävyyttä suoritettavana koodina laiteark-

kitehtuurista toiseen. Sovelluskehitysympäristön huolellinen valinta takaa laiteriippumattomuuden paremmin kuin järjestelmien teknisten ominaisuuksien ja kasvatettavuuden vertailu.

Mikä on avoin?

Voidaanko kaikkia Unix-järjestelmiä pitää avoimina? Avoimuuden määrittely riippuu aina määrittelijästä. Järjestelmätöimittajat, jotka ovat luoneet asemansa suljetuilla järjestelmillä, esittelevät kernaasti avoimina sellaisia ratkaisuja, joilla tuetaan liittymistä heidän suljettuihin järjestelmiinsä. Avoimuuden sanomaa on rakennettu myös monipuolisia tietoliikenneominaisuuksia korostamalla. Kumpikin näistä perusteluista on yksipuolinen ja harhaanjohtava.

Hyväksikäyttäjän kannalta avoimena voi pitää sellaista järjestelmää, joka sallii vapaan, sovelluskohtaisen laitevalinnan, mahdollistaa useiden erimerkkisten palvelimien käyttämisen samassa järjestelmässä eri tehtä-

vien suorittamiseen ja pääsyn näihin resursseihin kaikilta työasemilta, sekä tarjoaa riittävät tietoliikennepalvelut ulkoisiin järjestelmiin liittymiseksi hyvin määritellyin liikkennöintitavoin.

Entä tulevaisuudessa? Lähivuosien Unix-hahmotelmat tuovat eteemme monisuoritinjärjestelmät, erityisesti Risc-tekniikalla saavutettavan suorituskyvyn moninkertaistumisen, graafisten käyttöliittymien ja multimedian täysimittaisen hyödyntämisen, järjestelmien hajautumisen sekä sovellusmielessä että maantieteellisesti ja entistä yhtenäisemmät käyttöjärjestelmät. Näiden seikkojen ennustetaan yhdessä ajavan Unixia ennätysvoimalla eteenpäin.

Unix-järjestelmien todelliset kilpailutekijät löytyvät toimittajien kyvystä sulauttaa tarjoamansa järjestelmät osiksi suurempia kokonaisuuksia ja varmistaa asiakkaansa sovellusinvestoinnin jatkuvuus. Laitteiden samankaltaistuksessa entisestään vaaditaan toimittajilta palveluja, jotka auttavat asiakasta käyttämään tietojärjestelmäänsä renkinä eikä isäntänä.



Kimmo Keskinen työskentelee tuotepäällikkönä Nokia Datan Tuotehallinnassa Unix-järjestelmien tuessa.

Mikko 100 — avoin mutta turvallinen

Mikko 100 toimii yli 1400 suomalaisen yrityksen yksittäisenä monen käyttäjän kokonaisjärjestelmänä tai lähiverkon sovellus- ja tietokantapalvelimena. Se tarjoaa järjestelmäratkaisun operatiivisille toimialakohtaisille sovelluksille ja toimitukseen. Mikko 100 -järjestelmät ovat osa Nokia Datan avointa Mikko Unix -tuoteperhettä.

Nokia SCO UNIX on Mikko 100 -järjestelmien uusi käyttöjärjestelmä, jonka uudet piirteet ja parannukset keskittyvät tietosuojaan, kansainvälisten standardien ja yhteensopivuuden noudattamiseen, sovellusvalikoiman laajentamiseen ja järjestelmän hallintaan.

Mikko 100 perustuu Intel 80386/80486 -suorittinteknologiaan. Unix-käyttöjärjestelmä tekee henkilökohtaiselta työasemalta näyttävästä laitteistosta monipuolisen monen käyttäjän järjestelmän, joka tukee mm. myynnin, materiaalihallinnon, tuotannon, liikekirjanpidon, reskontran, terveydenhuollon, palkanlaskennan, kunnossapidon ja tukkukaupan toimintaa pienissä ja keskusuurissa yrityksissä, suuryritysten osastoilla ja julkishallinnossa. Käyttäjät hyödyntävät yhteisiä tietokantoja, ohjelmia, resursseja ja lähiverkko- ja tietoliikennetyhteyksiä muihin järjestelmiin.

Nokia SCO UNIX -käyttöjärjestelmän avoin ympäristö antaa mahdollisuuden valita sadoista valmissovelluksista parhaiten soveltuva ohjelmistokokonaisuus. Asiakas hyötyy PC-teknologian suorituskyvyn ja ominaisuuksien jatkuvasta ja nopeasta teknisestä kehityksestä ja volyymituotannon ja markkinakilpailun tuomista taloudellisista hinnoista/käyttäjän- ja hinta/suorituskyky -eduista. Se on yhdistelmä, johon myös monet johtavat ohjelmistotalot luottavat ja johon ne lisäävät sovellustarjontaansa.

Yrityksen kokoinen järjestelmäratkaisu

Yritykselle on tärkeää voida valita kutakin toimipaikkaa ja tehtävää varten sopivan kokoiset kes-

kenään yhteensopivat laitteistot. Pääsovelluksen tulee toimia erikokoisissa laitteistoissa muutoksista ja eri ympäristöihin on voitava lisätä muita ohjelmistoja. Mikko 100 soveltuu yhden käyttäjän ympäristöstä noin 50 samanaikaisen käyttäjän järjestelmäksi. Nykyinen laiteperhe käsittää neljä eri mallia. Pienin Mikko 100 m36 sopii erinomaisesti pienyrityksen kokonaisratkaisuksi. Perusmalli on valmis kolmelle käyttäjälle ja laajennettavissa seitsemän käyttäjän järjestelmäksi. Muista Mikko 100 -malleista poiketen Mikko 100 m36 toimitetaan ainakin toistaiseksi SCO XENIX -käyttöjärjestelmällä.

Mikko 100 m52 ja Mikko 100 m55 sopivat pienen tai keskisuurisen yrityksen järjestelmäksi, lähiverkon palvelimeksi, pienen ryhmän erikoissovelluksiin ja sovelluskehitykseen. Käyttäjiä Mikko 100 m52:ssa voi olla yli 15 ja Mikko 100 m55:ssä yli 30.

Laiteperheen suurin malli Mikko 100 m57 sopii yrityksen tai osaston työasemaverkon sovellus- ja tietokantapalvelimeksi tai monen käyttäjän kokonaisjärjestelmäksi. Käyttäjiä Mikko 100 m57:ssä voi olla jopa 50.

Keskusyksiköiden lisäksi Mikko 100 koostuu työasemista, sarja- ja lähiverkkoliitännöistä, Unix-käyttöjärjestelmästä, tietoliikenneohjelmistoista, tiedonhallintajärjestelmästä, käyttöympäristöistä ja valmisohjelmistoista. Kaikki osat perustuvat avoi-



Jarmo Heikkinen

miin standardiliitännöihin ja -tuotteisiin, joilla turvataan mahdollisuus laajentaa järjestelmää asteittain.

Runsas sovellustarjonta

Sovellukset ovat ratkaisevassa osassa yrityksen valitessa tietojärjestelmäratkaisua. Oma tai tilattava sovelluskehitystyö varten Mikko 100:aan on saatavilla yleisimmät 4GL-sovellushittimet ja SQL-tiedonhallintajärjestelmät (Progress, Informix, Oracle, Ingres,...) sekä eri ohjelmointikielten kääntäjiä (C, Cobol, Fortran, Pascal, Basic...).

Mikko 100 toimistossa.

Pienenkin oman tai räätälöidyn sovelluskehitystyön kustannukset nousevat kuitenkin helposti suuremmiksi kuin valmiin ohjelmiston ja laitteiden. Mikko 100:lle on saatavilla Suomesta satoja useiden toimialojen yleissovelluksia ja erikoisohjelmistoja. Lisäksi useista henkilökohtaisten työasemien Dos-ohjelmistoista on olemassa monen käyttäjän "look/feel/work-alike" Unix-versiot. MS-Dos-emuloinnin avulla voidaan hyödyntää myös alkuperäisiä 8086 Dos-ohjelmia.

Piirteet yhdistyvät

Nokia SCO UNIX tuli toukokuussa aiemman SCO XENIX -käyttöjärjestelmän rinnalle. Uuden käyttöjärjestelmän nimi on Nokia SCO UNIX System V/386 versio 3.2. Sen ovat kehittäneet AT&T, Microsoft, Santa Cruz Operation ja Nokia Data. SCO XENIX on edelleen maailman eniten käytetty Unix-käyttöjärjestelmä.

Nokia SCO UNIX yhdistää XENIX System V ja UNIX System V -käyttöjärjestelmät. Yhdistäminen tarjoaa etuja sekä loppukäyttäjälle että sovellusten kehittäjille. Loppukäyttäjät voivat hyödyntää tuhansia XENIX System V ja UNIX System V -ympäristöihin kehitettyjä sovelluksia. Sovellusten kehittäjät voivat puolestaan markkinoida ohjelmiaan entistä laajemmalle käyttäjäkunnalle. Nokia SCO UNIXin piirteet luovat tehokkaan sovellusympäristön ja käyttöjärjestelmä käyttää hyödykseen Intel 80386 ja 80486 -suorittimien tarjoaman suoritusky-

Hyväksytty avoimeksi

Avoimuudella ja yhteensopivuudella on suuri merkitys pyrittäessä mahdollisimman laajaan sovellustarjontaan. Mikko 100 noudattaa kaikkia merkittäviä avoimia kansainvälisiä standardeja. Mikko 100 ja Nokia SCO UNIX ovat yhdessä läpäisseet X/Open-hyväksymismenettelyn ja saaneet oikeuden käyttää X/Open-tuotemerkkiä. Nokia SCO UNIX noudattaa AT&T:n System V Interface Definition (SVID) määrittelyn viimeisintä versiota SVID Issue 3:a, POSIX 1003.1:tä, IEEE:n standardia siirrettävälle käyttöjärjestelmälle ja FIPS 151.1:tä (Federal Information Processing Standard).

Sovellukset yhteensopivia

Standardien noudattaminen on perusta Mikko 100:n laajalle yhteensopivuudelle. Nokia SCO

UNIX on binaari- ja lähdekodiyhteensopiva UNIX System V tai XENIX System V pohjaisten sovellusten, AT&T UNIX System V/386 versio 3.0:n ja Microsoftin, SCO:n ja IBM:n XENIX versioiden kanssa. Lähdekielisiä ohjelmia Nokia SCO UNIXiin on mahdollista siirtää esimerkiksi kaikista X/Open- ja POSIX-yhteensopivista ympäristöistä.

Vaivaton käyttöliittymä

Nokia SCO UNIX (versio 3.2B) sisältää ergonomisen valikko- ja ikkunointipohjaisen käyttöliittymän. Tavallisten merkkipohjaisten VDU-päätteidenkin käyttäjät voivat ajaa useita eri ohjelmia samanaikaisesti, siirtyä ikkunas- ta toiseen ja siirtää tietoa niiden välillä. Lisäksi käyttöliittymä sisältää Unix Managerin, helppo- käyttöisen valikon, jolla käyttäjät voivat vaivatta hyödyntää Unixin eri toimintoja ilman kommentojen syntaksin tarkkaa tuntemusta.

Yhteyksiä ja tietoturva

Mikko 100 tukee tiedonvälitystä useilla eri tietoliikennetyhteyksien toteutusvaihtoehdoilla: TCP/IP, NFS, X.25, X.25/PAD, SNA 3270 SDLC/QLLC, SNA RJE SDLC/QLLC ja BSC RJE. Omat tiedot ovat suojassa, sillä Nokia SCO UNIX tietoturvaominaisuudet vastaavat tietokoneille määritellyn C2-tason vaatimuksia (U.S. DoD NCSC TC-SEC).

Nopea käyttöönotto

Nokia SCO UNIX toimitetaan valmiiksi Mikko 100 -järjestelmään asennettuna. Esiasennuksen yhteydessä siihen tehdään määrittelyt päätteille, kirjoittimille, levykeasemille, kiintolevyille, kasettinauha-asemalle ja muille oheislaitteille. Myös muut toimitettavat lisäpiirteet ja ohjelmistot esiasennetaan valmiiksi, joten asiakas voi ottaa järjestelmänsä heti käyttöön. Nokia Da-

Nokian ja SCO:n yhteistyö laajeni

Nokia Datan ja maailman johtavan Unix-käyttöjärjestelmätoimittajan The Santa Cruz Operation Inc:n yhteistyö laajeni äskettäin, kun Nokia Data ja SCO solmivat sopimuksen, jonka mukaan Nokia Data laajentaa tuotevalimoimaansa SCO:n Open Desktop -tuotteilla. Open Desktop on Unix-käyttöjärjestelmän, graafisen käyttöliittymän, verkkoliitännän ja SQL-tiedonhallinnan ja Dos-ympäristön muodostama ohjelmistokokonaisuus, joka tukee useiden erityyppisten sovellusten samanaikaista suorittamista työasemassa ja monen käyttäjän palvelimissa. Open Desktopin avulla graafisia sovelluksia voidaan käyttää tehokkaasti ja hajautetusti myös verkoissa.

Lisätietoja tuotepäällikkö Glen Koskelalta, Nokia Data, p. (90) 1241.

Mikko 100-sarja

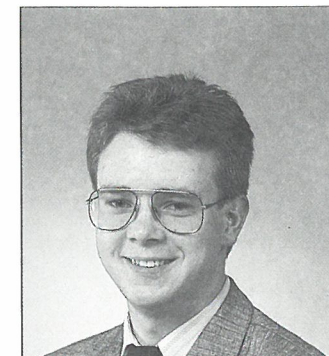
Mikko 100	m36	m52	m55	m57
Suoritin	80386	80386	80386	80486
Kellotaajuus	20 MHz	20 MHz	25MHz	25MHz
Keskusmuisti (Mt)	4 - 16	4 - 16	4 - 24	8 - 32
Pikamuisti (Kt)	-	-	64	8 (sisäinen)
Levymuisti (Mt)	100	150 - 1320	150 - 1320	300 - 1320
Streamer 150 Mt	optio	vakio	vakio	vakio
SCSI-liitäntä	vakio	vakio	vakio	vakio
Ethernet-liitäntä	optio	optio	optio	optio
Sarjaliitäntä	2 - 6	10 - 34	10 - 34	2 - 34
Tyyp. käyttäjien lkm	1 - 7	3 - 20	6 - 30	12 - 50

Mikko 100 -mallien tekniset ominaisuudet.

tan asiakaspalvelulta saa Mikko 100:lle sekä koulutusta että laitteiston ja ohjelmiston ylläpitopalveluja. Loppukäyttäjä voi soittaa muun muassa puhelinpalveluun, jos järjestelmän käytössä on ongelmia.

Sovelluksia yhteistyökumppaneilta

Kymmenet suomalaiset ohjelmistotalot ovat valinneet Mikko 100:n laitteistoratkaisuksi, johon yhdistävät oman toimialaosaimisensa. Nämä Nokia Datan yhteistyökumppanit toimivat lähellä asiakasta ja voivat keskittää resurssinsa toimialaosaimiseen ja sovelluskehitykseen. ■



Glen Koskela työskentelee Nokia Datan Tuotetuki ja markkinointi -osastolla Mikko Unix -järjestelmien tuotepäällikkönä.

Sarlin uskoo Unixiin

Oy E. Sarlin Ab uusii kaikkien toimialojensa tietojärjestelmät vaihe vaiheelta kokonaisvaltaiseksi ja avoimeksi kokonaisuudeksi.

Sarlinin toiminta keskittyy neljälle liiketoiminta-alueelle, joiden tietotekniset tarpeet eroavat huomattavasti toisistaan. Sarlin Pumpput kehittää, valmistaa ja markkinoi erilaisia pumppuja ja pumppaamoja. Sarlin Automaatio ja Sarlin Rakennuskoneet tuovat maahan nimensä mukaisia tuotteita. Sarlin Lämpökäsittelytekniikka suunnittelee, valmistaa ja markkinoi metalliteollisuuden lämpökäsittelylaitteita ja -järjestelmiä.

Rasitteena vanhat ongelmat

Työn alla oleva järjestelmä on jo Sarlinin toinen operatiivinen tietojärjestelmä. Vuonna 1981 suunniteltiin pantu ja 1983 valmiiksi saatu kaupan ja teollisuuden Terttuun perustunut järjestelmä on vastannut Sarlinin talous- ja materiaalihallinnosta aina näihin päiviin saakka. Vuonna 1987 käyttöön otettuja työasemaverkkoja lukuun ottamatta järjestelmää on kehitetty lähinnä ohjelmistojen kautta.

Vuosien mittaan järjestelmä on kuitenkin vanhentunut, ylläpitokulut nousseet ja luotettavuusvaatimukset tiukentuneet. Vanhan järjestelmän räätälöintiin osallistuneet ihmiset ovat jo siirtyneet muualle, joten ohjelmiston toimintaan liittyvän tiedon saanti aiheuttaa jonkin verran ylimääräistä työtä. Lisäksi Sarlin Pumpput -yksikön tuotannonohjauksessa alettiin vuonna 1986 käyttää Tukiset-ohjelmistoa ensin yksittäisessä mikrossa ja sitten verkossa. Tietojärjestelmän muutamia projekteja vetänyt **Sakari Puhakka** kertoo, että Tukiset ei tahtonut istua muuhun järjestelmään: "Tertun ja Tukiset-järjestelmän yhteiselo ei oikein toiminut. Tiedon siirtäminen oli hankalaa ja hidasta, eikä Pumpput-yksikössä koskaan

päästy reaaliaikaiseen ohjaukseen. Lisäksi samoja tietoja jouduttiin syöttämään molempiin järjestelmiin erikseen, mikä on nykyaikana turhauttavaa työtä."

Pelkän tuotannonohjauksen asemesta haluttiin saada tuotannonohjausta, myyntiä ja materiaalihallintoa ohjaava kokonaisjärjestelmä. Tällaiselta operatiiviselta perustalta on sitten helppo ruveta kehittämään johdon ja markkinoinnin järjestelmiä.

Kolmen vuoden suunnittelu

"Vuoden 1987 syksyllä eri toimittajat esittivät käsityksensä siitä miten he näkivät tulevaisuuden muovautuvan ja mitä heillä on tarjota Sarlinin tyyppiselle yritykselle", Sakari Puhakka kertoo. Lisäksi tietoa kerättiin ulkopuolisen konsultin avulla. Jo vuonna 1987 hän suositti Unix-ympäristöä, mutta arvioi että se ei ollut tuolloin vielä kypsä tuotantokäyttöön.

Vuonna 1988 tehtiin käyttäjäkysely, jossa haastateltiin koko yrityksen käyttäjäkuntaa. Sen pohjalta tehtiin yhteenvedo käyttäjien tarpeista. Seuraavaksi ryhdyttiin kehittämään yrityksen sisäistä organisaatiota eli valittiin eri yksiköihin atk-yhdyshenkilöt ja muodostettiin valvontaryhmä, johon kuuluvat eri yksiköiden johtajat, atk-yhdyshenkilöt, talousjohtaja ja toimitusjohtaja. Ryhmä teki tietojärjestelmävalinnan viime syksynä.

Eri paketit eri tarpeisiin

Sarlinin tietojärjestelmäprojekti jakautuu kolmeen pakettiin: taloushallintoon, kaupallisten yksiköiden (Automaatio ja Rakennuskoneet) toiminnanohjauk-

seen sekä Pumpput-yksikön toiminnanohjaukseen. Kukin paketti sisältää omat ohjelmistonsa, joilla asiat hoidetaan.

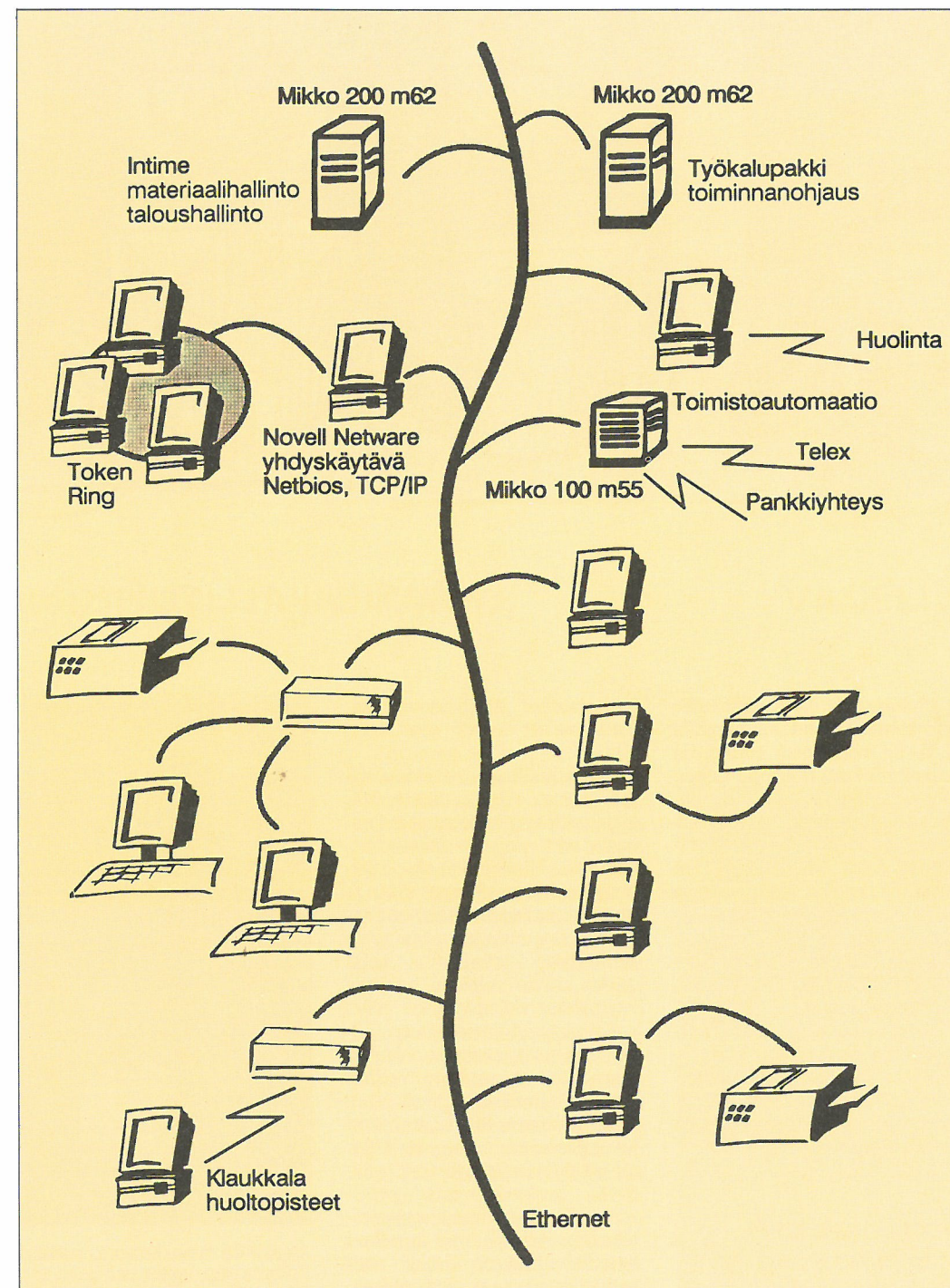
Taloushallinto otettiin käyttöön viime vuoden vaihteessa. Kaupallisten yksiköiden toiminnanohjaus ja Pumpput-yksikön järjestelmä tulevat käyttöön vuodenvaihteessa.

Selkeän vaihejaon syistä tärkeimpänä Sakari Puhakka pitää koko projektin hallittavuutta: "Eri projekteissa on kiinni samoja henkilöitä, ja jos tehdään kerralla paljon suuria muutoksia niin on olemassa vaara, että koko organisaatio menee sekaisin. Lisäksi etenkin kaupallisissa yksiköissä järjestelmään joudutaan syöttämään runsaasti lähtötietoa, jotta tilastot saadaan näyttämään jotain. Jo pelkkiä tuotetietoja on noin 12000, asiakastietoja 9000 ja toimittajatie-toja 4000. Vaikka tietoja syötetäänkin järjestelmään ulkopuolisin voimin, kaikki kooditukset tulee suunnitella ja tulokset tarkastaa itse."

Vastuu yhdellä toimittajalla

Kysymykseen siitä, miksi Sarlin päätyi Nokia Dataan, Sakari Puhakka vastaa: "Me haimme kokonaisvaltaista toimittajaa, yhtä kumppania, joka ottaa vastuun koko projektin onnistumisesta. Emme halunneet niinkään Mikrokoja tai Intime Openia tai Työkalupakia, vaan toimivan kokonaisjärjestelmän, joka koostuu näistä osista. Vaikka tällaisessa laajassa projektissa joudutaan turvautumaan alihankkijoihin, Nokia Data otti mielestämme parhaiten kokonaisvaltaisen toimittajan roolin harteilleen."

Laitepuolella erityisesti järjestelmän avoimuus ja modulaarisuus kiinnitti huomiota. Vaikka



laitteiden suorituskyvyn mitoituksessa onkin otettu huomioon myös tulevat tarpeet, järjestelmän tietojenkäsittelykapasiteettia on helppo lisätä sijoittamalla osa sovelluksista omiin keskus-koneisiinsa.

Ohjelmistopuolella eräs ratkaiseva tekijä on ollut se, että ohjelmistot ovat Progress-pohjaisia, ja Sarlinilla on Progress

Query-lisenssi. Tämä mahdollistaa tarpeen vaatiessa myös sellaisten tietojen yhdistelemisen ja etsimisen, joita valmissovellukset eivät osaa kaivaa esiin. Näin pystytään myös ohjelmistopuolella pitkäaikaisiin mahdollisimman pitkälle pakettiratkaisuissa vastakohtana aiemman järjestelmän pitkälle vietyyn räätälöintiin.

Sarlinin kokonaisjärjestelmä, kun kaikki on valmiina.



Sakari Puhakka Sarlinilta kertoo järjestelmän ohjelmistovalinnoista: "Unicin Intime oli luonnollinen valinta, sillä taloushallinnossa se on johtava vaihtoehto Suomessa. Vastaavasti Intime toiminnanohjaus on peruslähtökohdiltaan ja käyttöliittymältään nykyaikainen paketti. Systian Työkalupakki on puolestaan tuotannollisten pk- ja suurempienkin yritysten toiminnanohjauksen kärkinimiä."

Maatalous yhdistyy Unixilla



MAATILAHALLITUS

MAATALOUSPIIRI

KUNTA

VILJELIJÄ

Maatilahallitus on valtionhallinnon tietotekniikan edelläkävijöitä.

Kehitystyö lähti liikkeelle kahteen suuntaan. Samalla kun henkilöstö totutettiin tietotekniikkaan mikropohjaisen toimistoautomaation avulla, maatilahallituksen, maatalouspiirien, kuntien maatalouslautakuntien sekä eri sidosryhmien tietotarpeista laadittiin kattava atk-kokonaisselvitys. Tämä selvitys saatiin valmiiksi vuoden 1988 alkupuolella, ja sitä on käytetty kokonaisvaltaisen atk-kehitystyön perustana.

Parhaillaan kehitetään keskuskonepohjaisia järjestelmiä. Atk-päällikkö **Jaakko Kettunen** mukaan ne ovat paras keino yhteistiedostojen ja viestinvälityksen järjestämiseksi.

Tietojärjestelmiä vaiheittain

Kokonaisselvityksessä maatilahallituksen tietotarpeiden havaittiin jakautuvan viiteen erilliseen osaan.

Henkilöstöhallinnon henkilöstötiedoista ja palkanlaskennasta huolehtii henkilöstöhallinnon tietojärjestelmä, joka on tällä hetkellä pääosiltaan valmis.

Kaikkein olennaisimman osan muodostaa kuitenkin maataloushallinnon operatiiviset rutiinit kattava tietojärjestelmä, jo-

hon kuuluu 25 osasovellusta. Ensimmäiset niistä ovat työn alla.

Taloushallinnon tietojärjestelmä vastaa maksuliikenteestä, kirjanpidosta, kustannuslaskennasta sekä yleisestä lainajärjestelmästä. Yleisen lainajärjestelmän kehityksessä ovat Vtkk ja Valtionkonttori vahvasti mukana. Sen laajuutta kuvastaa hyvin kehitystyön aikataulu, neljä vuotta. Koko systeemi saadaan käyttöön arviolta vuonna 1993.

Toimistotietojärjestelmän piiriin kuuluvat puolestaan kaikki tavanomaiset toimiston rutiinit kuten tekstinkäsittely, tiedonhallinta, sanomavälitys, diariointi ja julkaisutoiminta. Järjestelmässä ollaan siirtymässä mikroista keskuskoneesta pyöri-vään Finkis-toimistotietojärjestelmään. Aluksi tämä tapahtuu maatilahallituksen lisäksi vain kahdessa maatalouspiirissä, mutta jatkossa kaikkien piirien on tarkoitus liittyä järjestelmään.

Kaiken tämän kokoa yhteen johdon tietojärjestelmä, joka tiivistää käyttäjilleen kaiken olennaisen tiedon muista järjestelmistä muun muassa graafisessa muodossa. Johdon tietojärjestelmää lähdetään kehittämään kuitenkin vasta sitten kun kaikki muut osajärjestelmät ovat valmiina.



Atk-päällikkö Jaakko Kettunen maatilahallituksesta uskoo syntyvässä olevan tietojärjestelmän soveltuvan maataloushallinnon tarpeisiin myös tulevaisuudessa: "Tapahtuipa melkein mitä muutoksia tahansa, tietojärjestelmäjäkoon meidän ei tarvitse puuttua."

Tietoliikenne ratkaisee

Tietojärjestelmäkehityksen tavoitteena on parantaa palvelukykyä, tuottavuutta ja taloudelli-

suutta. Tavoitteisiin pyritään tietoliikennepalvelujen hyväksikäytöllä. Eritasosten tietoverkkojen avulla voidaan nopeuttaa tiedon kulkua, vähentää päällekkäistä työtä ja saada sama tieto samaan aikaan eri hallintotasoille ministeriöstä kuntiin asti.

Koska viljelijät ovat viime vuosina ostaneet paljon mikroja ja neuvontajärjestöt ovat tuottaneet runsaasti maatalohjelmistoja, myös maatilat voivat joskus välittömästi hyötyä valtakunnallisesta tietojärjestelmästä.

Lisäksi yhteydet muihin ministeriöihin ovat kehittyneet. Kaiken kehitystyön pohjana on valtionvarainministeriön antama suositus T90, jonka kantavana periaatteena on avoimiin, käytännössä Unix-pohjaisiin tietojärjestelmiin siirtyminen.

Keskuskoneita, mikroja ja päätteitä

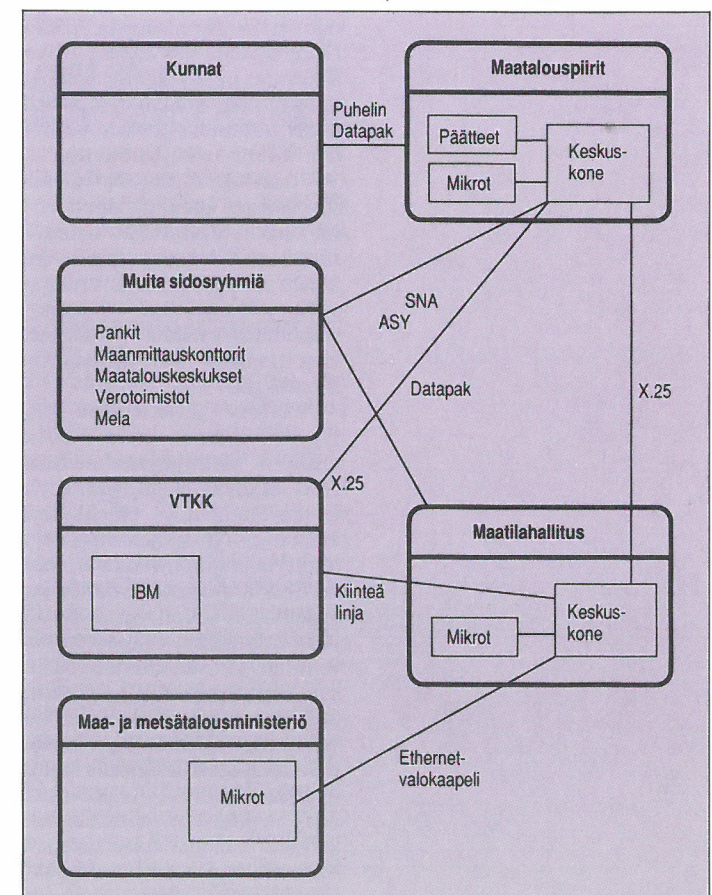
Järjestelmäarkkitehtuuri perustuu hajautettuun tietojenkäsittelyratkaisuun. Maatilahallitukseen ja kaikkiin 16 maatalouspiiriin tulee keskuskone sekä Ethernet-verkko. Maatilahallituksessa työasemina käytetään mikrotietokoneita ja piireissä mikrotietokoneita ja päätteitä. Tietoliikenne maatilahallituksen ja piirien

välillä tapahtuu x.25-välitteisenä.

Ensimmäiset hankitut keskuskoneet kuuluvat Nokia Datan Mikko 200-sarjaan. Maatilahallitukseen tulee Mikko 200 m64 ja Kuopion ja Turun piiriin Mikko 200 m61. Valinnan perusteista Jaakko Kettunen kertoo: "Yhtenä tekijänä olivat toki valtiovarainministeriön suositukset. Hinta-suorituskyky -suhde oli näissä Unix-laitteistoissa erittäin hyvä, sillä Mikko 200:han on Risc-pohjainen tietokone. Lisäksi Unix on epäilyksettä se järjestelmä, joka tulee jatkossa kehittymään. Yhdysvalloissa tehtyjen tutkimusten mukaan vuoteen 1993 mennessä valtaosa monen käyttäjän järjestelmistä rakentuu Unixin varaan."

Syntyvässä olevan järjestelmän tulevaisuus näyttää hyvältä, sillä tietojärjestelmäjako pitää Jaakko Kettusen mielestä paikansa myös hyvin pitkälle tulevaisuuteen: "Itse maataloushallinto saa muuttua todella paljon ennen kuin tietojärjestelmäjako tarvitsee ryhtyä remontoimaan. Toisaalta hankittu tekniikka on uusinta uutta eli lähivuosina ei tekniikkapuolellakaan tarvita sen kummempia toimenpiteitä — mitä nyt uusia maatalouspiirejä otetaan järjestelmän piiriin hankkimalla niille tarvittava tietotekniikka."

Maatilahallituksen tai maatalouspiirin sisäinen järjestelmä rakentuu Ethernetin varaan. Tietoliikenne muihin järjestelmiin tapahtuu x.25-välitteisesti.



Tandem Integrity S2 Vikasietoinen Unix-ratkaisu

Avointen järjestelmien yleistymisen on luonut tarpeen myös vikasietoisille ja tiedon eheyden takaaville Unix-järjestelmille. Tandem on siirtynyt Integrity S2 järjestelmän julkistuksen myötä vauhdilla Unix-maailmaan.

Unix-järjestelmien markkinat ovat laajentuneet viime vuosina teknisistä ja teollisista järjestelmistä myös kaupallisiin sovelluksiin. Voimakkainta on ollut keskikokoisten järjestelmien kasvu. Avoimet Unix-järjestelmät turvaavat sovellusohjelmiin tehty sijoitukset, sillä asiakkaat voivat siirtää sovelluksensa tarpeen mukaan useiden eri laitevalmistajien tuotteisiin.

Vaativuudesta ei tingitä

Käyttäjät ovat asettaneet selvät vaatimukset vikasietoisille Unix-laitteille. Ensinnäkin niiden tulee olla täysin yhteensopivia AT&T:n Unix System V:n kanssa. Vikasietoisuus täytyy saavuttaa ilman muutoksia Unixin ytimessä, mikä tarkoittaa uusien versioiden joustavaa käyttöönottoa.

Vikasietoisuuden täytyy olla sovelluksien kannalta läpinäkyvä ominaisuus. Käyttäjät haluavat siirtää sovelluksensa paitsi koneesta toiseen, myös tavanomaisesta järjestelmästä vikasietoiseen ilman sovelluksen uudelleen kirjoittamista tai ylimääräisiä muutoksia.

Avointen järjestelmien täytyy olla avoimia myös muuten kuin käyttöjärjestelmältään. Niin oheislaitteiden, verkkoyhteyskäytännöjen, ohjelmointikielten ja -työkalujen kuin käyttöliittymienkin on oltava standardien mukaisia. Avoimet järjestelmät mahdollistavat käyttäjälle kolmansien osapuolten sovellusten ja lisälaitteiden joustavan käyttöönoton ja helpottavat uusien teknisten ratkaisujen soveltamista. Usein Unix-laitteet palvelevat laajalle alueelle hajautetuissa verkoissa, jolloin osaan henkilöstön ylläpitäminen kaikilla paikoilla tulee kalliiksi. Kustannusten kiristämiseksi onkin tärkeää, että suuri osa jär-



jestelmän vaatimista huolto- ja tukitehtävistä ei vaadi erikoiskoulutusta.

Suorituskykyä Risc-suorittimella

Tandemin Integrity S2 on suorituskykyinen vikasietoinen Unix-tietokone. Se perustuu Risc-tekniologiaan ja tarjoaa jatkuvaa käytännön ympäristön AT&T Unix System V -sovelluksille.

Laitteen käyttöjärjestelmä on Tandemin NonStop-UX, joka on AT&T:n System V Interface Definition -määritelmän (SVID) mukainen. NonStop-UX:n pohjana on AT&T:n Unix System V versio 3. Siihen on lisätty Berkeley Unixin ominaisuuksia kuten nopea

tiedostojärjestelmä ja TCP/IP-tietoliikennetyökalut sekä Tandemin omia laajennuksia, jotka parantavat Unixin luotettavuutta ja varmistavat tiedon eheyden.

Koneen hyvä suorituskyky pohjautuu 32-bittiseen R2000 Risc-suorittimeen, jonka teho on 11–12 MIPSiä. Suoritin on MIPS Computer Systemsin valmistama, ja sen kellotaajuus on 16,67 megahertsiä.

Vikasietoinen laitearkkitehtuuri

Vikasietoisuutta voidaan parantaa sekä laitteisto- että ohjelmistotasolla. Järjestelmät, joissa vi-

kasietoisuutta on yritetty parantaa lisäämällä koodia Unixin ytimeen, ovat kohdanneet kuitenkin vaikeuksia. Ongelmaksi on muodostunut yhteensopivuuden saavuttaminen sekä uusien Unixin versioiden käyttöönoton vaikeutuminen. Tandem Integrity S2:ssa vikasietoisuus on toteutettu laitteistotasolla.

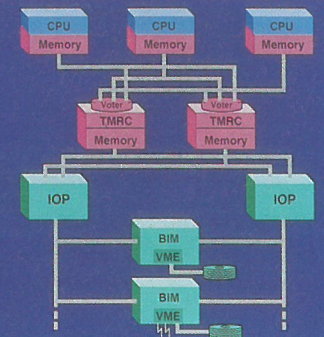
Kone on suunniteltu siten, että mikään yksittäinen laitevika ei kaada sitä. Järjestelmän arkkitehtuuri perustuu kolminkertaiseen suoritusosaan sekä kahdennettuihin muisti-, I/O-väylä-, levy- ja virransyöttöjärjestelmiin.

Tandem S2:n teknisesti ehkä mielenkiintoisin piirre on sen suorittimien TMR-arkkitehtuuri (triple modular redundancy). Ta-

vallisesti modulaarista redundanssia on käytetty sotilas- ja avaruussovelluksissa, joissa monta eri suoritusyksikköä muodostaa loogisesti yhden suorittimen. Jokainen yksikkö suorittaa samoja käskyjä, ja tuloksista äänestetään. Mikäli jonkin yksikön tulos eroaa muista, se todetaan vialliseksi ja poistetaan käytöstä.

Jotta modulaarinen redundanssi toimisi, täytyy verrattavia yksiköitä olla vähintään kolme. Jos käytössä olisi kaksi yksikköä, jotka antaisivat eri tuloksen, tietäisimme toisen niistä olevan viallisen, mutta kumpi niistä olisi oikeassa? Avaruussovelluksissa on käytetty jopa seitsemää eri suorittinta, mutta kyseisellä sovellusalueella laitteiston hin-

Integrity S2 Architecture



Integrity S2 -arkkitehtuuri.

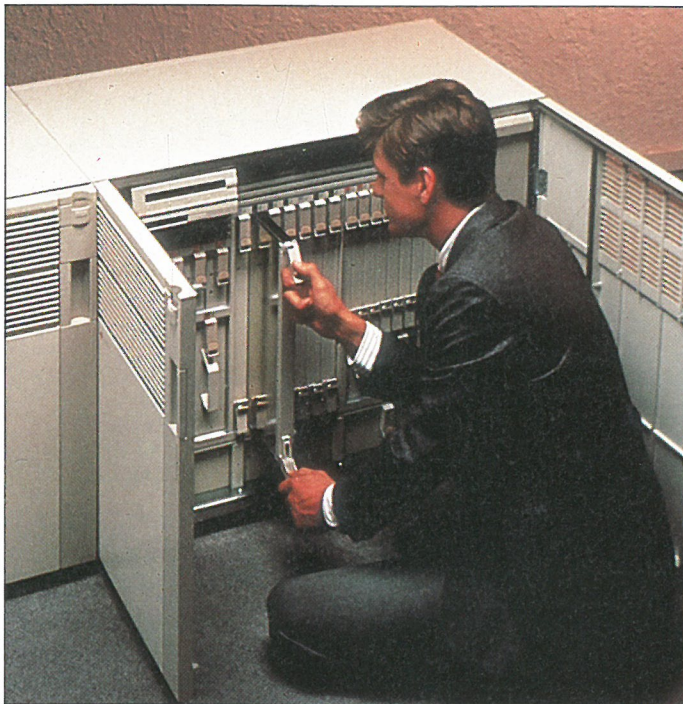
nalla ei ole juurikaan merkitystä. Tandem valitsi arkkitehtuurinsa perustaksi kolminkertaisen redundanssin, sillä silloin toteutus on vielä järkevän hintainen ja edelleen vikasietoinen.

Vikasietoisten arkkitehtuurien suunnittelun yksi suurimpia haasteita on vikasietoisuuden saavuttaminen ilman, että laitteiston suorituskyky kärsii siitä olennaisesti. Vaikka käytössä olisi tehokkaat 12 MIPSin suorittimet, mutta jokaisen käskyn jälkeen jouduttaisiin äänestämään, koneen suorituskyky putoaisi 3–4 MIPSin luokkaan. Tandemin ratkaisuna on ollut kytkeä suorittimet löyhästi yhteen. Jokaisella suorittimella on oma paikallinen muistialueensa ja kellonsa. Äänestykseen turvautaan vain silloin, kun viitataan globaaliin muistialueeseen, käsitellään keskeytyksiä tai kun 2048 käskyjaksoa on kulunut ilman äänestystä.

Kahdennus varmentaa

Täydellinen vikasietoisuus on saavutettu kahdentamalla kaikki kriittiset osat. Suorittimien yhteiset globaalit muistit sisältävät molemmat saman tiedon, joten muistin tai jonkin suorittimen vikaantuminen ei vaikuta Tandem S2:n suorituskykyyn.

Järjestelmän I/O-väylänä on NonStop V+ -väylä, joka on Tandemin kehittämä vikasietoinen VME-väylän versio. Luonnollisesti koneessa on kaksi väylää, joita kumpakin hallitsee oma I/O-suorittimensa. Ohjaimina käytetään tavallisia VME-ohjaimia, jotka on erillisten liitäntäyksiköiden avulla kytketty molempiin V+ -väyliin. Toisin kuin suorittimet ja muisti, I/O-väylät toimivat epäsymmetris-



Valtioneuvoston kanslian sähköpostijärjestelmän takana on Tandem Integrity S2 -laitteisto.

sähkökatkon aikana kone tallettaa tilansa levyille, ja sähköjen palaututtua sovellukset voivat jatkaa siitä tilasta, jossa ne olivat. Koneen toiminta sen uudeleen käynnistyessä voidaan määrittää sovelluskohtaisesti.

Standardien maailmassa

Toiminnallisen joustavuuden varmistamiseksi Tandem S2 noudattaa merkittävimpiä tietoliikenne-, ohjelmointikieli- ja tietokantastandardeja. Integrity S2 voidaan liittää SNA-, x.25- ja tahdistamattomiin verkkoihin. Paikallisverkkoratkaisuna on Ethernet TCP/IP, ja verkkoon haajautettua tietojenkäsittelyä tukevat NFS (Network File System) ja X-ikkunointijärjestelmä.

Käyttöjärjestelmän mukana tulee C-kieli. Erillisinä tuotteina on saatavissa Fortran-, Pascal- ja Cobol-kääntäjät. Vaikka Integrity S2 on suunniteltu yleiskäyttöiseksi koneeksi, useat sovellukset ovat tapahtumakäsittely-ympäristössä, jossa apuna on relaatio-tietokantaohjelmistot kuten Informix, Ingres ja Oracle.

Kolmansien osapuolien sovel-lusohjelmistotarjonta on runsas, sillä MIPS:lle tehdyt tavanomaiset ohjelmat ovat suoraan siirrettävissä Integrity S2:een.

Kone soittaa itse apua

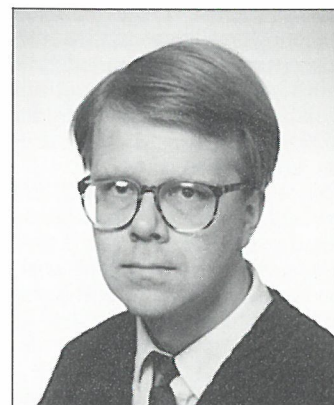
Integrity S2:n suunnittelun yhtenä lähtökohtana on ollut ylläpidon helppous, millä on erityistä merkitystä sen elinikäisistä kustannuksista arvioitaessa. Kone on suunniteltu siten, että jokainen sen pääkomponentti kuten suoritin, muisti, levyt, tuulettimet ja tehonlähde voidaan vaihtaa koneen ollessa päällä. Vaihdamisen voi suorittaa kuka tahansa, sillä osien irrottamiseen ei tarvita työkaluja.

Varusohjelmiston mukana tulee asiantuntijajärjestelmään pohjautuva ylläpito- ja diagnostiikkaohjelmisto, joka valvoo järjestelmän toimintaa. Jos esimerkiksi johonkin suorittimeen on tullut vika, järjestelmä poistaa suorittimen käytöstä. Sen jälkeen suorittimella ajetaan testejä, ja jos sen havaitaan olevan kunnossa, se liitetään takaisin järjestelmään.

Koneessa on myös erillinen huoltosuoritin, jonka avulla järjestelmä voi ottaa modeemiyhteyden huoltopisteeseen. Jos esimerkiksi suorittimen havaitaan olevan viallinen, kone voi itse ottaa yhteyden huoltoon ja ilmoittaa asiasta.

Sovelluksia Suomessa

Suomeen on toimitettu tämän vuoden aikana ensimmäiset Tandem S2 -järjestelmät. Valtioneuvoston kanslian sähköpostijärjestelmä perustuu Nokia Datan toimittamaan sähköpostiohjelmistoon ja Tandem S2 -laitteisiin. Helsingin Puhelinyhdistykselle on puolestaan asennettu kone Elisa-käyttöön. Integrity S2:lla toteutettu Nokia Datan huollonohjausjärjestelmä SMS on koekäyttövaiheessa.



Tapio Kaukonen työskentelee järjestelmä-asiantuntijana Nokia Datassa vastuualueenaan Tandem/Unix-tuotteet.

Tiimikko -talo ja tuote

Tietovalli Oy:n toimitusjohtaja Tarmo Lindén kutsuu Mikro-Ässä-ohjelmistoa "monen käyttäjän moniyritys-ohjelmistoksi". Perustelu löytyy toteutuksen modulaarisuudesta ja tietokannan käytettävyydestä lähiverkossa.

Mikro-Ässä-ohjelmisto koostuu useista sovelluksista: Kirjanpito ja kustannuslaskenta, Palkanlaskenta ja henkilöstöhallinta, Tilauksen käsittely ja laskutus, Myyntireskontra, Ostoreskontra, Pankkiliikenne, Tarjouslaskenta ja työmaaseuranta, Varastovalvonta, Ostotilaukset ja -tilastot sekä Osoiterekisteri. Jokaista sovellusta voidaan käyttää erikseen ja niistä voidaan koota kokonaisuus yrityksen tarpeen mukaan.

Massamuistittomia työasemia

Ohjelmiston tietokannat sijaitsevat yhteisellä palvelimella ja ohjelmistojen suoritus tapahtuu työasemissa. Työasemilta voidaan suorittaa samanaikaisesti sekä tapahtumankäsittely- (esimerkiksi myyntitilausten syöttö) että eräajotyyppejä (muun muassa raportointi) toimintoja. "Eräajotoiminnot ovat itse asiassa hyvin kriittisiä työasemaverkoissa. Niitä pitäisi pystyä ajamaan muun työskentelyn ohessa ilman että eräajot häiritsevät tai hidastavat muuta työskentelyä", Tarmo Lindén toteaa.

Massiivisia eräajoja varten Mikro-Ässä-ohjelmistossa voidaan verkkoon määritellä eräajotyöasema, jolle mitkä tahansa verkon työasemat voivat lä-

Monen yrityksen monen käyttäjän Mikro-Ässä



Ulf Ekholm

Tietovallin ohjelmiston Mikro-Ässä-käyttöliittymästä löytyvät monentasoiset komennot sekä aloittelijalle että kokeneille käyttäjille.

hettää eräajojonoja käsiteltäväksi. Näin saadaan käytössä olevat työasemat pidettyä varsinaisessa tapahtumankäsittelyssä. Toisaalta eräajotyöasemaa voidaan hyödyntää normaalina työasemana eräajokäytön ulkopuolella.

Tietovallin kokonaisjärjestelmätöimitukset sisältävät ohjelmistojen lisäksi Lan Manager tai 3+Share verkkokäyttöjärjestelmän, Ethernet-kaapeloinnin ja MikroMikko palvelimet ja työasemat.

Työasemina on tyypillisesti käytetty 1 megatavun työmuistilla varustettuja 80286-pohjaisia MikroMikko 3 m215 -mikrotietokoneita. Tarmo Lindénin mukaan tullaan massamuistitonta Intelin 80386SX-suorittimella varustettua MikroMikko 4 m330-verkkotyöasemaa myymään jatkossa Tietovallin toimittamien verkkojen työasemiksi. Verkkotyöasemat ovat tervetulleita niiden edullisuuden ja parantuneen tietoturvan takia.

Perustellut välinevalinnat

Tietovallin ensimmäiset ohjelmistot koodattiin Cobol-kielillä. Vuonna 1986 ryhdyttiin pohtimaan siirtymistä nykyaikaisempaan ja helpompaan ohjelmankehitystympäristöön.

Tietokannalle asetettiin vaatimuksiksi, että se olisi maailmalla laajasti käytössä oleva monen käyttäjän tietokanta mikroverkkoon. Myös tietokannan eheyttä ja toipumista poikkeustilanteista pidettiin tärkeänä. Tietokantoja ja välineitä arvioitiin yhdessä muiden ohjelmistotalojen kanssa. Arvioinnin tuloksena päätettiin siirtyä ohjelmatuotannossa dBASE-tietokannan ja Clipper-kehittimen käyttöön.

Clipperillä tuotettu koodi voidaan kääntää konekieliseksi. Osa ohjelmistossa käytettävistä funktioista ja aliohjelmista on toteutettu Microsoft C:llä ja assemblerilla. C-kieltä käytetään oheislaitteajureiden (esim. luottokortinlukija) toteutuksessa. Assemblerilla puolestaan on toteutettu pieniä konversiorutiineja. Kääntäjistä (Clipper, C, assembler) syntyneet objektimoduulit

voidaan linkata yhdeksi konekieliseksi ohjelmaksi.

Mikro-Ässä voidaan yhdistää myös valmissovelluksia. Excel- ja Lotus-taulukkolaskentaohjelmistoilla voidaan tehdä suoraan hakuja dBASE-tietokantaan. Taulukkolaskentaohjelmistoa käytetään esimerkiksi silloin kun halutaan simuloida tarjouslaskentaa omien räätälöityjen laskentasaattajien pohjalta.

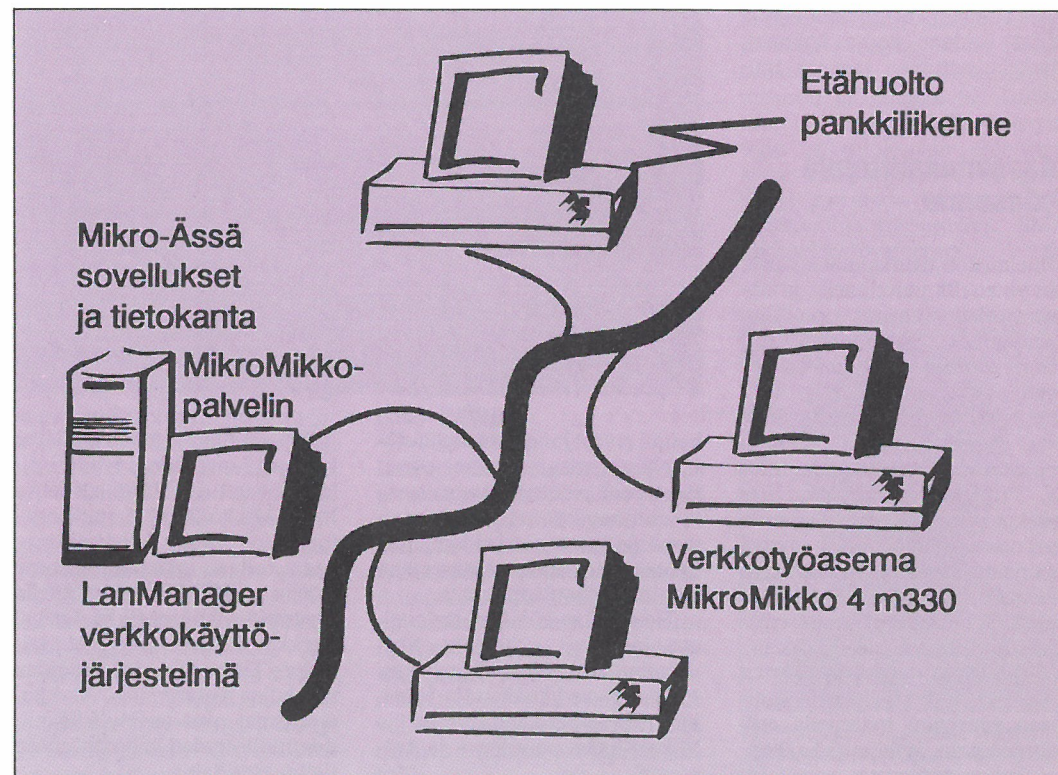
Ohjelmistonkehityksessä ja versionhallinnassa käytetään kehitysyörymän omaa Lan Manager-pohjaista verkkoa. Jokaisen ohjelmankehittäjän käytössä on Norton Guides -ohjelmistolla toteutettu vuorovaikutteinen opastusjärjestelmä. Se sisältää Clipperin komentojen kuvaukset sekä Tietovallissa kehitettyjen omien kirjastojen kuvaukset. Jokaisesta funktiosta ja aliohjelmasta on kerrottu komennon syntaksi, yleinen kuvaus toiminnasta, parametrit sekä versio- ja tekijätiedot. Kirjastot on jäsennetty esim. ikkunointi-, kirjoitin-tulostus-, tarkistus-, verkkotuki- ja käyttöliittymätoimintoihin. Kun opastusjärjestelmä sijaitsee yhteisellä palvelimella, siihen tehdyt muutokset ovat aina kaikkien käytettävissä.

Monentasoinen käyttöliittymä

Mikro-Ässän ensimmäisten ohjelmistoversioiden aikaan ei juurikaan keskusteltu ohjelmistojen yhteneväisestä käytöstavasta ja -liittymästä.

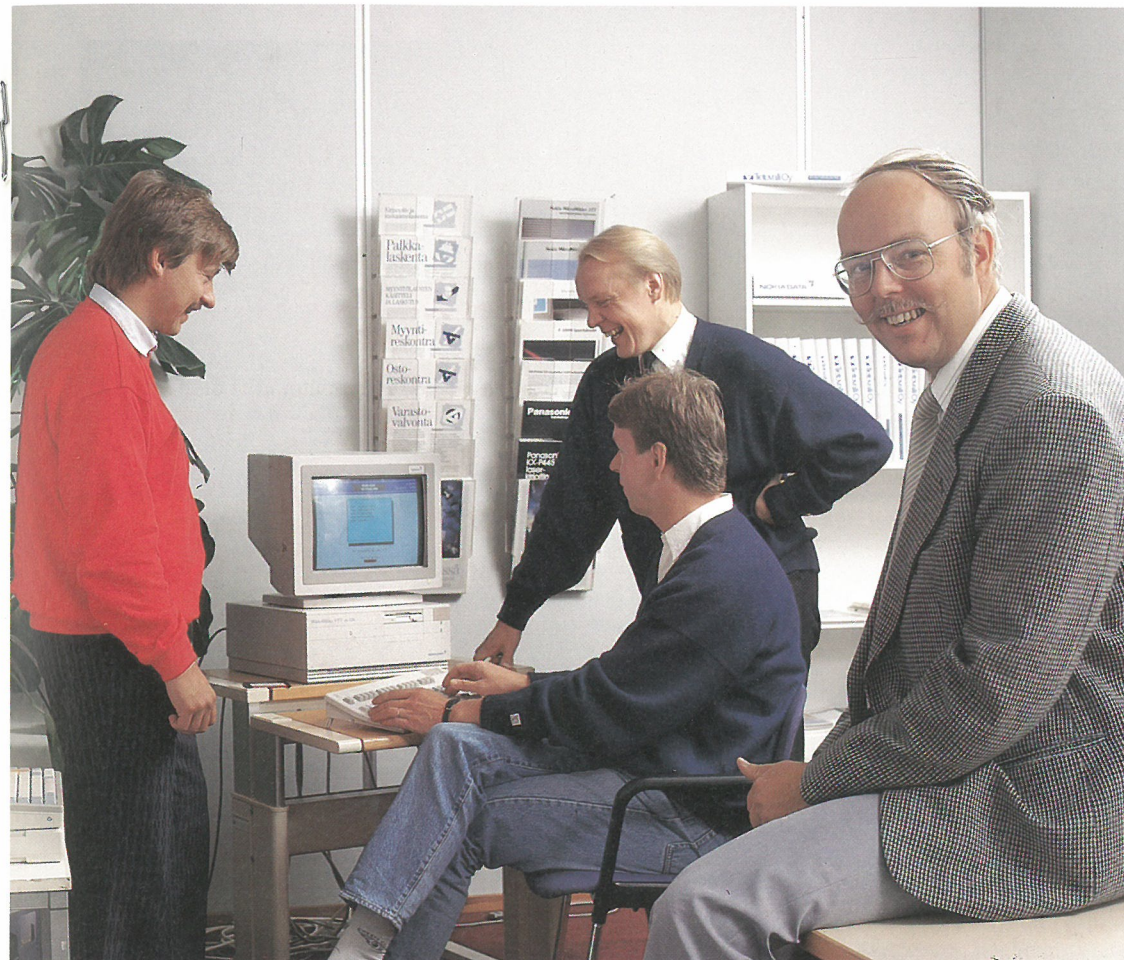
Mikro-Ässän käyttöliittymät ovat kehittyneet yleisen käyttöliittymäkehityksen myötä. Ohjelmisto noudattaa näppäimistön käsittelyssä vakiintunutta standardia. Esimerkiksi F1 on avustustoiminto, ESC on paluu edelliseen kohtaan jne. Nuolia ja hiirtä voidaan käyttää selailuissa ja valinnoissa. Valintoja voidaan tehdä myös numeronäppäimillä.

Ikkunointia käytetään hyödyksi tilanteissa, joissa halutaan esimerkiksi tietojen syöttövaiheessa selata vastaavaa tietokantaa. Tilausten käsittelyssä pelkkä Enter-näppäimen painaminen vaikkapa asiakaskentän kohdalla aiheuttaa asiakastietokannan avaamisen uuteen ikkunaan. Tämän jälkeen voidaan tarvittava asiakastieto hakea syöttökenttään.



Tietovallin toimittamien verkkojen työasemiksi myydään jatkossa massamuititonta Intelin 80386SX-suorittimella varustettua MikroMikko 4 m330 -verkkotyöasemaa. Verkkotyöasemiin siirytään muun muassa niiden edullisuuden ja parantuneen tietoturvan takia.

"Uudet ohjelmistonkehitystekniikat kiinnostavat, mutta päätöksiä niihin siirtymisestä ei voi tehdä hätäillen", toteaa Tietovalli Oy:n toimitusjohtaja Tarmo Lindén (oikealla) ohjelmistokehittäjineen, joista kuvassa ovat Kauko Tähkävuori (vasemmalla), Timo Hokkanen ja Harri Hellsten.



Ulf Ekholm

Ohjelmistohuoltoa etäpalveluna

Mikro-Ässä-ohjelmiston käytön neuvonnasta ja koulutuksesta huolehtii koulutusyhtiö Tukivalin henkilökunta. Vaihtoehto puhelinneuvonnalle ja konsultoinnille paikan päällä on Tukivallin tarjoama etähuoltopalvelu.

Asiakas voi ottaa modeemin välityksellä yhteyden etähuoltotietokoneeseen. Kun yhteys kytkeytyy, on asiakkaan kone neuvojan käytössä. Asiakas näkee koko ajan ruudultaan toimenpiteet, joita asiakasneuvoja tekee ja voi halutessaan osallistua työskentelyyn. Neuvojan ja asiakkaan välillä voidaan käydä keskustelua ruudulla. Etähuoltoa käytetään myös ohjelmistoversioiden päivityksiin.

Välineitä punnitaan

Entä miten Tietovallissa suhtaudutaan tämän päivän uutuuksiin ohjelmistonkehityksessä: SQL-tietokantaan ja palvelin-työasemarakkitekhtuuriin? "Suuret asennukset asettavat haasteita tietokantakäsittelylle, joten välineitä on arvioitava jatkuvasti. Kehitämme valmiutta siirtä SQL Server -ympäristöön, jotta se olisi mahdollista silloin kun asiakkaat sitä haluavat. Päätöksiä entisistä tekniikoista luopumisesta ja uusiin tekniikoihin siirtymisestä ei kuitenkaan voi tehdä hätäillen", toimitusjohtaja Lindén toteaa.

Tukkukaupoista paikallisradioihin

Tietovalli Oy on työasemaverkkopohjaisiin talous- ja materiaalihallinnon ohjelmistoihin keskittynyt Nokia Datan jälleenmyyjä eli Tiimikko-talo.

Se on perustettu vuonna 1985 ja työntekijöidensä omistama. Yrityksen liikevaihto oli vuonna 1989 15 miljoonaa markkaa ja työntekijämäärä 27.

Tietovallin kehittämän ohjelmistoperheen Mikro-Ässän sovelluksia on toimitettu 2800 kappaletta 900 asiakkaalle. Keskimäärin Mikro-Ässä-järjestelmässä on 3-5 työasemaa. Suurimassa järjestelmässä on 50 työasemaa.

Tietovallin päätoimialoja ovat LVIS-urakointi ja tilitoimistot. Muille toimialoille tukkukaupasta ja järjestöistä paikallisradioihin ja videovuokraamoihin on olemassa valmiita sovituksia, joita toimitetaan yhteistyökumppanien kautta. Uusimmalla toimialalla, matkatoimistoalalla, Tietovalli tekee yhteistyötä Nokia Datan ja Amadeus Finland Oy:n kanssa. Tietovalli kehittää matkatoimistoille tarkoitettua taloushallinto-ohjelmistoa ja sovitaa sen yhteen Amadeus Pro -matkavarausohjelmiston kanssa.

Ensimmäinen EISA-MikroMikko

MikroMikko 4 m458



MikroMikko 4 -tuoteperhe laajenee EISA-väyläisellä MikroMikolla. MikroMikko 4 m458 on myös ensimmäinen MikroMikko-malli, jossa on käytetty Intelin 33 megahertsin kello- taajuudella toimivaa 80486-suoritinta.

Tehokkaampien verkkopalvelimien markkinakasvu on tänä vuonna ollut noin 30 prosenttia. Nokia Datalla on palvelinmarkkinoista runsaan 20 prosentin markkinaosuus.

MikroMikko 4 m458 on suunniteltu tehokkaaksi verkon palvelimeksi. Siinä on nopea suoritin, suuret, laajennettavat massamuistit, kolmitasoinen salasanasuojaus ja hyvät liitettävyysominaisuudet.

Tehotekniikkaa

MikroMikko 4 m458:n 80486-suoritin toimii 33 megahertsin kelloaajuudella ja siinä on sisäinen liukulukyksikkö sekä 8 kilotavua suorittimen välimuistia. Lisäksi ulkoista välimuistia voi saada 128 kilotavua. Ulkoiselle välimuistipiirille on valmis kanta. Työmuistia on kahdeksasta 64 megatavuun ja koko muisti mahtuu systeemikortille eikä vie laajennuspaikkoja. Laitteessa voidaan käyttää sekä EISA- että

ISA-yhteensopivia laajennuskortteja.

Vakiokokoonpanoon kuuluu SCSI-massamuistiohjain. Vaihdoetoisesti voidaan käyttää EISA-SCSI-kiintolevyohjainta, jossa on kaksi megatavua välimuistia. Kiintolevyjen kapasiteetit ovat 200 tai 660 megatavua tai 1 gigatavu. Kiintolevyjä mahtuu laitteen sisälle 2. Valinnaisena on saatavana 5,25 tuuman levykeasema sekä nauhavarmistin.

MikroMikko 4 m458:ssa on neljä sarjaliitintä, rinnakkaisliitintä ja PS/2-hiiriliitintä. Vaapaana on viisi EISA-korttipaikkaa lisälaajennuksille. Korttipaikoista kolme on BusMaster- eli väylänhaltijakortteille.

Käyttöturvaa

MikroMikko 4 m458 on usein yrityksen tietojärjestelmän keskeinen laite. Käyttäjää yhdellä MikroMikko 4 m458:lla voi olla jopa useita kymmeniä. Näin

MikroMikko 4 m458:ssa on noin 30 prosenttia nopeampi suoritin kuin tähän asti nopeimmassa MikroMikko-mallissa 448. Käytettäessä EISA-SCSI-kiintolevyohjainta voidaan suorituskykyä parantaa vielä huomattavasti.

myös käyttöturvapalvelujen ja laitteen turvapiirteiden merkitys korostuu. Verkon mahdollisimman häiriöttömän toiminnan turvaamiseksi Nokia Data tarjoaa erilaisia käyttöturvapalveluja. Käytön turvallisuutta lisää myös varavirtalähteen eli UPS:n käyttäminen. Etenkin käytettäessä EISA/SCSI-ohjainta on varavirtalähteen käyttö suositeltavaa, koska osa tiedoista saattaa olla sähkökatkoksen tultua vasta ohjaimen välimuistissa. Verkko-käyttäjärjestelmä Nokia MS LAN Manageriin saa varavirtalähteen käyttöä tukevan ohjelmiston.

Lisätietoja: Anneli Martonen, p. (90) 1241.

Charisma 2.0 soveltuu vaativankin esitysgrafiikan ja kalvojen tekemiseen.

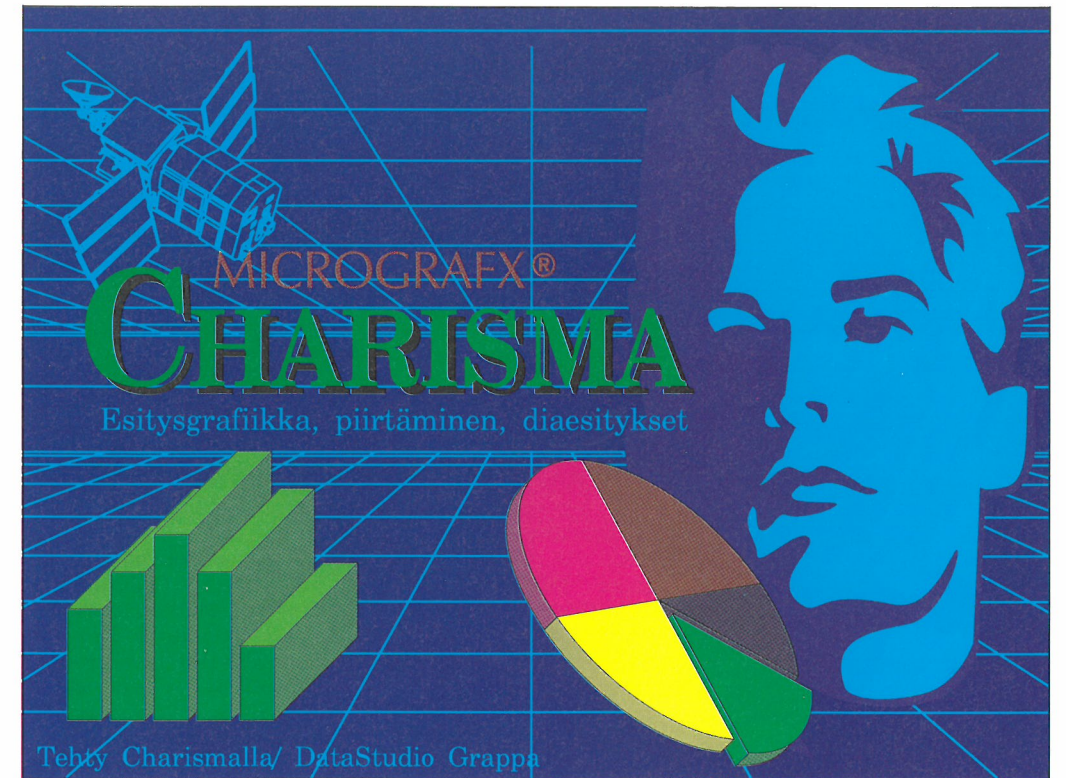
Charisma 2.0 jatkoa Graph Plussalle

Johtava Windows 3 -yhteensopivien grafiikkaohjelmistojen tekijä Micrografx on tuonut markkinoille uuden grafiikkaohjelman Charisma 2.0:n. Järjestysnumero 2 huolimatta kyseessä on ohjelmiston ensimmäinen versio. Numero 2 selittyy sillä, että Charisma rakentuu toisen ohjelman, Graph Plussan päälle ja korvaa sen jatkossa.

Charisma 2.0 muistuttaa hyvin paljon saman valmistajan Designeria. Näistä Windows 3 -yhteensopivista ohjelmistoista Charisma sopii vaativan esitysgrafiikan ja kalvojen tekemiseen, Designer vaatimaan tekniseen ja mainospiirtämiseen.

Charisma 2.0:ssa ovat muun muassa samat piirtoalkiot ja väripaletit kuin Designerissa. Lisäksi siinä on mukana yli 2200 valmista kuvaa. Tekstikalvojen tekoon löytyy kalvoille valmiita tyyli pohjia tai niitä voi tehdä itse. Slide show -toiminnolla kalvot voi esittää mikron näytöllä. Esitysgrafiikan lisäksi Charismassa on monipuolinen yritysgrafiikka, jolla taulukkolaskennan numerodatasta piiryy selkeitä esityksiä. Ohjelma lukee suoraan Excel ja Harvard Graphics -taulukkoita.

Charisma 2.0:ssa on hyvät WYSIWYG-ominaisuudet. Ohjelman 40 kirjasinlajia näkyvät koneen näytöllä samana kuin ne tulostuvat. Kirjasinlajit pystytään tulostamaan millä tahansa grafiikkaa tulostavalla kirjoittimella. Ohjelmiston saa 3500 markalla.



WordPerfect 5.1 suomenkielisenä

Tekstinkäsittelyohjelma WordPerfectin uusin versio 5.1 on julkaistu suomenkielisenä. Ohjelmiston uusi piiri on mm. mahdollisuus käyttää hiirtä. Toimintonaappainten rinnalle ovat tulleet pudotusvalikot.

Nyt WordPerfectillä on myös mahdollisuus liittää taulukoita tekstinkäsittelyyn. Taulukkolaskennan tiedot saadaan päivitettyä automaattisesti asiakirjoihin. Oman taulukkolaskentaoh-

jelman Plan Perfectin lisäksi WP-asiakirjaan voidaan tuoda ja päivittää taulukoita Lotus 1-2-3:sta tai Microsoft Excelistä.

Kaavamuokkain helpottaa matemaattisten ja teknisten kaavojen tuottamista. Sarkainasetelut ovat joustavimmat ja tavutuksen toimintaa on kehitetty vähemmän kyseleväksi. Asiakirjojen nimeämisessä voi käyttää 40 merkkiä pitkiä nimiä. Grafiikka tulostuu väreissä.

Windows kiinnosti

Softari 1/90:n lukijakysely kertoi Windows-artikkeleiden olleen kaikkein mielenkiintoisimpia. Listan kärjessä oli Windowsin muistinhallintaa ja käyttöä käsitelleen artikkelisarjan jälkimmäinen osa: "Moni Windows pyörii vajaateholla — turhaan". Lähes yhtä paljon suosiota sai myös Windows-sovellusten välisestä DDE-tiedonsiirtoa käsitellyt "Enemmän irti Windows-sovelluksista". Muut pidetyt artikkelit käsitelivät värillisiä nestekidenäyttöjä ja ohjelmistoergonomiia. Palkinnon saivat Leila Honkanen Lahdesta, Rauno Satopää Turusta, Simo Sauni Ylöjärveltä, Valto Rantanen Hämeenlinnasta ja Risto Ruuskanen Mikkelistä. Onnea!

Käyttötuen puhelinnumerot uudistuivat

Nokia Data on uudistanut puhelinpalvelunsa työasemakäyttäjille. Käyttötuen palvelunumero 567 2200, jonne on voinut soittaa mikrojen ohjelmistoja koskeissa pulmissa, poistui käytöstä 1.10.1990.

Käyttötukisopimusasiakkaiden yhteydenotot on otettu samasta päivästä lähtien vastaan asiakaspalvelukeskukseen numerossa (90) 567 3700, josta puhelut ohjataan edelleen käyttötuen järjestelmäasiantuntijoille. Sopimusasiakkailta tiedustellaan asiantuntijamääränsä heidän soittaessaan puhelinpalveluun.

Asiakkaat, jotka eivät halua solmia puhelinpalvelusopimusta, voivat soittaa numeroon 9700-7260. Heiltä veloitetaan palvelusta 14 markkaa minuutilta.

"Uudistuksella pyritään lisäämään puhelujen vastaanottokykyä ja nopeuttamaan ongelmien ratkaisua", toteaa palvelusta vastaava Juha-Pekka Salo, jolta saa myös lisätietoja p. (90) 567 2934.

Nokia MS LAN Manager 2.0:ssa uusia turvapiirteitä

Lokakuussa on tullut markkinoille uusi Nokia MS LAN Manager 2.0 -verkonhallintajärjestelmä. Sen avulla pyritään parantamaan muun muassa verkon tehokkuutta, turvallisuutta, hallintaa ja käyttövarmuutta. Lisäksi uuteen Nokia MS LAN Manageriin on saatavissa ominaisuus, joka käynnistää levykkeettömät työasemat LAN Manager -palvelimesta.

LAN Manager 2.0:n HPFS-tiedostojärjestelmä nopeuttaa kiintolevyn luku- ja kirjoitustoimintoja. Jos palvelimessa on Intel 80386- tai 80486-suoritin, asennetaan HPFS-tiedostojärjestelmän rinnalle automaattisesti verkkokäyttöön sovitettu 386-optimized network -syöttö- ja tulostusjärjestelmä.

Verkon hallintaa helpottamaan on muodostettu Single System Image -menetelmä, jossa käyttäjätietojen määrittely tehdään keskitetysti ryhmän (Domain) yhteen palvelimeen (Primary Server). Käyttäjien tietosuojaa on parannettu lisäämällä tietojen turvallisuusluokitusten määrää. Käyttäjätietojen hallittavuutta on helpotettu määrittelemällä uudentyyppisiä käyttäjäluokkia muun muassa tulostimien, tietoliikenteen, palvelimien resurssien ja käyttäjien hallintaan.

Uninterruptible Power Supply (UPS) -laitteen ja LAN Manager -palvelimen välille on rakennettu yhteys, jossa LAN Manager -ohjelmisto saa tietoa verkkovirran häiriöistä. Näin varmistetaan tie-

dostojärjestelmän eheys virtakatkosten jälkeen.

Lisäksi monistus- (File Replication) ja peilauspalveluiden (Disk Mirroring, Disk Duplexing) avulla voidaan palvelimessa olevat tiedot varmuuskopioida automaattisesti toiselle levyille tai palvelimelle.

Nokia MS LAN Manager 2.0:n Remote Boot -ohjelmistolla voivat levykkeettömät verkkotyöasemat käynnistyä palvelimien, tietoliikenteen, palvelimien resurssien ja käyttäjien hallintaan RIPL Manager -ohjelmiston.

Lisätietoja: Petri Karjalainen, p. (90) 1241.

Uusi virheenkorjaava korttimodeemi

Nokia Datasiirto on tuonut markkinoille virheenkorjaavan 1/2-kokoisen korttimodeemin, joka sopii muun muassa useisiin kannettaviin mikroihiin. Nokia ECM Quad PC/2 on keskinopea valintaverkon modeemi, joka hallitsee yleisimmät virheenkorjaavat yhteyskäytännöt (MNP 2-5).

Modeemin mukana saa Nokia Trans-Send Plus -tietoliikenneohjelman, johon sisältyvät kaikki yleisimmät tietoliikenne-emulaattorit. Se tukee muun muassa VT 220- ja Videotex-Prestel-emulaattoreita. Trans-Send Plussan avulla voi tehdä tiedonsiirron salauksen. Modeemi on yhteensopiva CCITT:n suositusten kanssa V.22 bis 2400 bit/s, V.22 1200 bit/s, 75/1200 bit/s, V.21 300 bit/s.

Modeemin ohjehinta on 3400 markkaa ja sitä myyvät HPY Data, Tele ja alueelliset teleyrietykset.

Lisätietoja: Lippon Rantanen, Nokia Datasiirto, p. (90) 567 3522 ja Kari Tuominen, p. (90) 567 3621

...ja uusi 19 200 bit/s modeemi

Lisäksi Nokia Datasiirrolta on ilmestynyt markkinoille kiinteiden yhteyksien modeemi N1920.

N1920 soveltuu X.25-verkkojen runko- ja tilaajayhteyksille, limitin(multiplekseri)verkojen runkoyhteyksille ja ylipäänsä kaksipisteyhteyksille, joissa tarvitaan 19 200 bit/s nopeutta ja joihin digitaalinen verkko ei yllä.

N1920 toimii kiinteillä 4-johdinyhteyksillä. Nopeuksilla 2400—14400 bit/s modeemi on CCITT:n suositusten V.33, V.29 ja V.27 bis mukainen. Nopeuksilla 19 200 ja 16 800 bit/s ei ole olemassa CCITT:n suositusta. Siksi N1920 käyttää niillä Nokia Datasiirron soveltamaa trelliskoodausta.

Häiriönsietokyvyn parantamiseksi N1920 on rakennettu metallikoteloon. Laitteen pääteliitäntä on suositusten V.24 ja V.28 mukainen.

N1920 sisältää suositusten V.54/V.52 mukaiset testitoiminnot. Modeemin asetukset voidaan tehdä kauko-ohjatusti etäispään modeemista etulevyn näppäimistön ja näytön avulla.

Lisätietoja: Kari Tuominen, p. (90) 567 3621 ja Heikki Mäki-järvi, p. 567 3075.

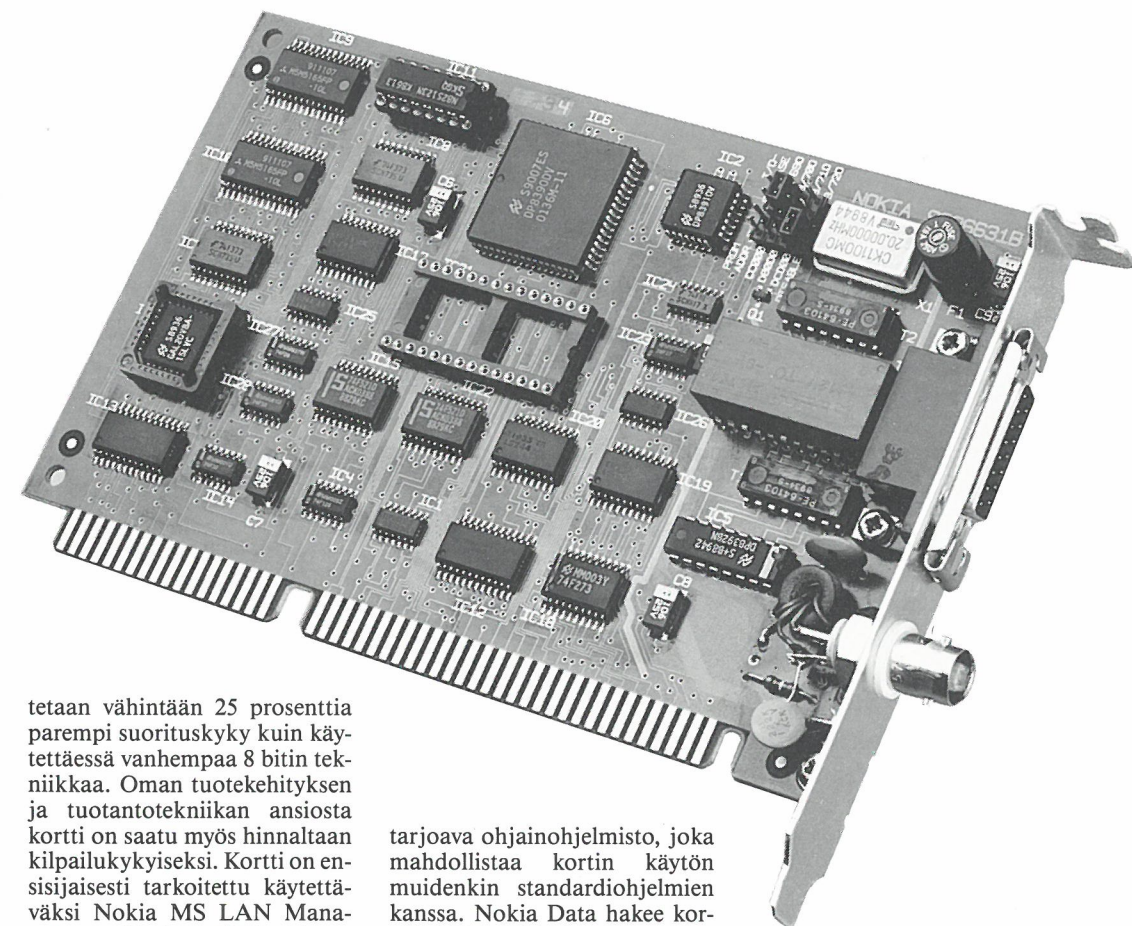
Nokia Datalta oma ethernet-kortti

Nokia Data tuo markkinoille oman EtherNet 2e -verkkoliityntäkortin. Kortti on kehitetty Suomessa Nokia Datan omassa tuotekehityksessä ja korttien valmistus on aloitettu Nokian Kilon tehtaalla.

Korttia markkinoidaan Suomessa sekä suoramyynninä että Tiimikko-myyjien kautta. Myös muut Nokia Datan eurooppalaiset myyntiyhtiöt ottavat kortin myyntiinsä.

Nokia Data on ollut Suomen suurin ethernet-korttien toimittaja. Viime vuonna se toimitti yhteensä 4900 3Comin valmistamaa ethernet-korttia. Omalla EtherNet 2e -kortilla Nokia uskoo valtaavansa entistä suuremman markkinaosuuden Suomessa. Kortteja toimitetaan tämän vuoden aikana muutamia satoja, mutta jo ensi vuonna toimitusmäärien arvioidaan olevan tuhansia.

EtherNet 2e on tehokas liittynäkortti standardeihin ethernet-verkkoihin. Sen tehokkuus perustuu 16-bittisen AT-väylän hyödyntämiseen, jolloin saavu-



tetaan vähintään 25 prosenttia parempi suorituskyky kuin käytettäessä vanhempaa 8 bitin tekniikkaa. Oman tuotekehityksen ja tuotantotekniikan ansiosta kortti on saatu myös hinnaltaan kilpailukykyiseksi. Kortti on ensisijaisesti tarkoitettu käytettäväksi Nokia MS LAN Manager -verkkokäyttöjärjestelmän kanssa. Kortin mukana toimitetaan myös NDIS-rajapinnan

tarjoava ohjainohjelmisto, joka mahdollistaa kortin käytön muidenkin standardiohjelmien kanssa. Nokia Data hakee kortille Microsoftin hyväksynnän.

Lisätietoja: Petri Karjalainen, p. (90) 1241

Oman tuotekehityksen ja tuotantotekniikan ansiosta Nokia Datan ethernet-kortti on myös hinnaltaan kilpailukykyinen.

MS SQL Server 1.1 OS/2-ympäristöön

Nokia Data julkistaa uusia työasema/palvelin-arkkitehtuuriin perustuvia tuotteita: MS SQL Server 1.1 for OS/2 on Microsoftin tietokantapalvelimen uusin versio OS/2-ympäristöön. SQL Windows - SQL Server Connector (SSC) on liittymäohjelmisto SQL Windowsin ja MS SQL Server 1.1:n välille.

SQL Server -tietokantapalvelin on kehitetty vaativiin hajautettuihin tapahtumakäsittelytehtäviin. Se toimii sekä OS/2- että Unix-palvelimissa. Monisäikeisen (multi-treaded) arkkitehtuurinsa ja varastorutiininsa (stored procedures) ansiosta se on erittäin suorituskykyinen.

SQL Serverin DB-Library on kirjasto, jolla työaseman sovelus käyttää tietokannan palveluja. Tämä avoin rajapinta on saatavana Dos-, OS/2-, Unix- ja nyt myös Windows 3 -ympäristöön.

Uudessa versiossa sen muistin-

tarve on pienentynyt ja suorituskyky kasvanut merkittävästi.

SQL Windows - SQL Server Connector mahdollistaa SQL Server -tietokantojen hyväksikäytön SQL Windows -sovelluksesta. SQL Windowsin 4GL -ohjelmointikielestä voidaan kutsua suoraan SQL Serverin palveluja.

Julkistetut tuotteet mahdollistavat SQL Windows -sovellushityksen nyt sekä SQL Base että SQL Server -ympäristöihin. SQL Serveriä suositellaan käytettäväksi suurta tehoa vaativissa tapahtumakäsittelytehtävissä tai haluttaessa laajennettavuutta Unix-ympäristöön. SQL Base on erityisen käyttökelpoinen silloin, kun tarvitaan IBM-yhteensopivuutta, tai kun on kyse pienistä sovelluksista, jolloin laitteistovaatimukset ovat pienet.

Lisätietoja: Esa Aho, p. (90) 1241

Uniplex-toimisto-ohjelmisto Nokian valikoimiin

Solidware Oy ja Nokia Data ovat allekirjoittaneet yhteistyösopimuksen Uniplex-toimisto-ohjelmiston markkinoinnista Nokia Datan Mikko-Unix-järjestelmiin. Sopimuksen mukaan Nokia Data alkaa markkinoida ja toimittaa Uniplex-ohjelmistoa yhdessä Tiimikko-jälleenmyyjänsä kanssa.

Uniplex on 65 prosentin markkinaosuudellaan Euroopan johtava integroitu toimisto-ohjelmisto Unix-järjestelmissä. Suomessa Uniplex on asennettu noin 500 järjestelmään.

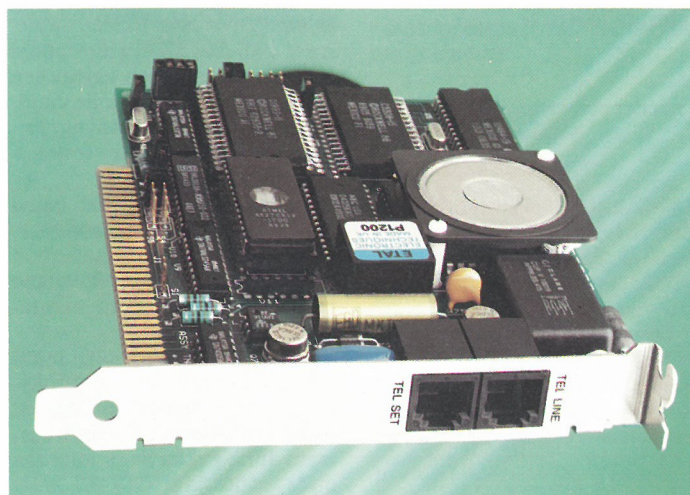
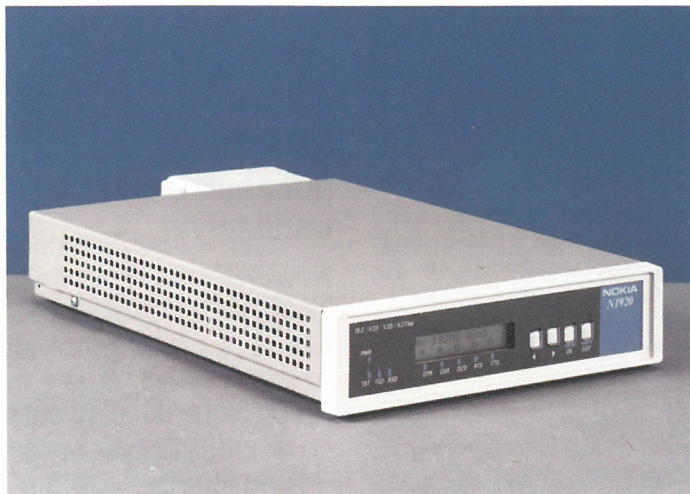
Nokia Data asentaa myymänsä Uniplex-ohjelmistot asiakkaidensa Mikko-järjestelmiin ja Solidware puolestaan tarjoaa asiakkaille Uniplex-koulutusta, ylläpitoa ja tukea. Nokia Data ja Solidware yhdessä soveltavat ohjelmistoon Nokian haluamia lisäpiirteitä, kuten X.400- ja telefax-yhteyden Uniplexin sähköpostiin sekä tuen Nokia Datan päätteille ja kirjoittimille.

Uniplexin perustoiminnot ovat tekstinkäsittely, taulukkolaskenta, tietokanta, SQL-kyselykieli ja piirto-ohjelma. Sen varsinainen toimisto-osa koostuu sähköpostista, ajanvarauksesta, kortistosta ja raportoinnista. Lisäksi Uniplexillä voidaan käsitellä kuvia ja esitysgraafikkia.

Lisätietoja: toimitusjohtaja Jorma Sauren, Solidware Oy, p. (90) 5281 ja tuotepäällikkö Glen Koskela, Nokia Data, p. (90)1241.

Kiinteiden yhteyksien modeemi N1920.

Virheenkorjaava korttimodeemi ECM Quad PC/2.



Hyöty irti ohjelmistoista

Nokia Datan Ohjelmistojen käyttö- ja palvelusopimuksen (OKP) rinnalle on tullut uusia ohjelmistopalveluja. Niiden kaikkien tavoitteena on, että asiakas saa mahdollisimman suuren hyödyn varusohjelmistoistaan.

Kuin tapaturmavakuutus

Ohjelmistojen käyttö- ja palvelusopimukseen kuuluu varusohjelmiston käyttöluva ja avustaminen ohjelmiston ensiasennuksessa, päivityksissä, käytössä ja ylläpidossa. Ongelmatilanteissa apua saa soittamalla Asiakaspalvelukeskukseen tai -pisteeseen. Helsingin yliopisto solmi muutamia vuosia sitten OKP-sopimuksen. Kaikkiaan sillä on yli kymmenen vuoden kokemus Nokia Datan asiakaspalveluista.



"Olemme tehneet ohjelmistojen käyttö- ja palvelusopimuksen Nokian kanssa ennen kaikkea siksi, että pysymme kehityksessä mukana", kertoo pääsuunnittelija Petri Kutvonen Helsingin yliopistosta.

"Yliopisto on hieman hankala asiakas, koska sillä on monia ehkä erikoisiakin toiveita. Yliopistolla saatetaan monesti olla tietoa enemmän kuin toimittajalla, mikä on välillä varsin turhauttavaa. Nokian kanssa tällaisia ongelmia ei ole ollut, mutta joidenkin toimittajien asiakaskoulutus on sitä, että asiakas kouluttaa toimittajan ylläpitopalveluista vastaavaa henkilökuntaa", virnuilee yliopiston pääsuunnittelija Petri Kutvonen. Hän toteaa, että palvelut on usein suunniteltu yrityksille ja monet korkeakoulut ja yliopis-

yliopistot ostavatkin vain kerta- palveluja. "Muutamaa OKP-sopimuksen ehtoa onkin muutettu, jotta sopimus vastaisi paremmin tarpeitamme", hän kertoo.

Kehityksen kannoilla

Sopimukseen kuuluu avustaminen muutosten ja korjausten yhteydessä syntyvissä ongelmissa. Neuvoja annetaan ongelmien määrittelyssä, tilapäisten korjausten tekemisessä ja ongelman ohittamisessa.

"Ensiasennuksen jälkeen yliopisto melko harvoin pyytää apua häiriötilanteissa", Kutvonen toteaa. "Se ei kuitenkaan tarkoita, etteikö näitäkin palveluja joskus tarvittaisi. Sopimus on ikään kuin tapaturmavakuutus. Ennen kaikkea se on solmitu siksi, että haluamme pysyä kehityksessä mukana. Kerran kuussa saamme sopimukseen piiriin kuuluvien laitteistojen valmistajan julkaisun, jossa käydään läpi teknisiä ongelmia ja esitetään ratkaisut niihin. Uusien ohjelmistoversioiden toimitus on myös tärkeää", hän toteaa.

Uusia palveluja Mikoille

Asiakailta saadun palautteen perusteella on OKP-sopimuksen pohjalta kehitetty uudet ohjelmistopalvelusopimukset (OP). OP-sopimukset on tarkoitettu Unix-pohjaisille Mikko-tietokoneille ja niitä on neljän tasoista.

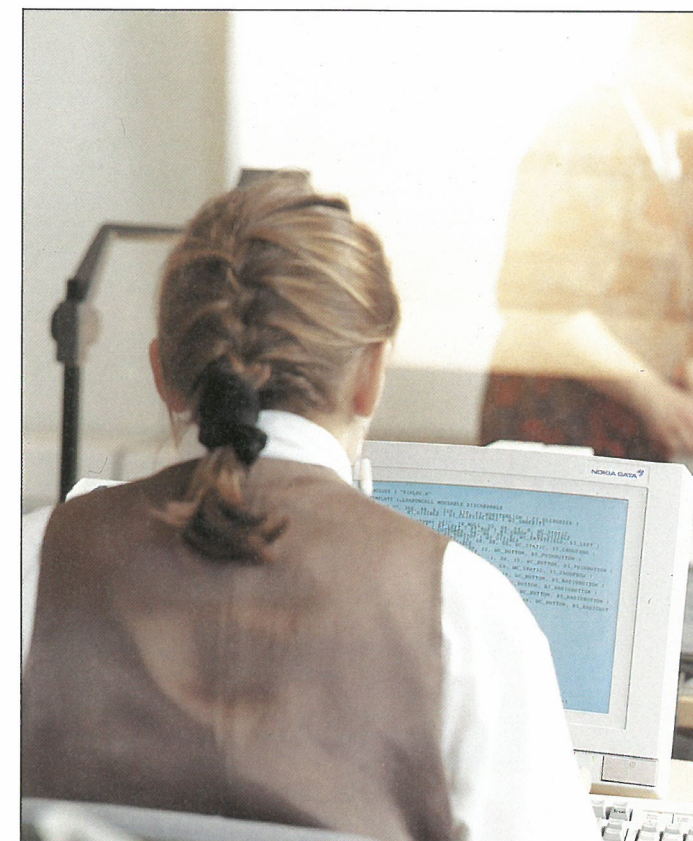
Asiakaille, joilla on vankka Unix-asiantuntemus, sopii puhelinpalvelu OP 1, johon kuuluu avustaminen ongelmissa puhelimitse, ohjelmistojen korjausten toimitus postitse ja uusista ohjelmistoversioista ja tuotteista tiedottaminen.

Peruspalvelu OP 2 tarjoaa OP 1:n palveluiden lisäksi etäyhteysohjelmiston ja ohjelmistojen vikojen selvittämisen asiakkaan luona. Etäyhteysohjelmistossa Nokia Datalla on sopimuksen piiriin kuuluvien laitteistoihin tietoliikenneyhteys, jonka avulla saadaan tarvittavat tiedot laitteiston tilasta.

OP 3:n ja OP 4:n palvelut ovat uusia. Päivitysohjelmistoversio OP 3:ssa uusi, testattu ohjelmistoversio toimitetaan postitse ja uusista versioista tiedotetaan. Laajennettuun palveluun OP 4:ään sisältyy OP 2 ja OP 3 sekä ohjelmistopäivitysten asennus, konfigurointi ja testaus. Lisätietoja: Rauno Paananen, Nokia Data, p. (90) 567 2375

Nitec kouluttaa Unixiin

Nokia Datan koulutuskeskus Nitec antaa Mikko-tuoteperheen Unix-laitteistokoulutusta. Kaksipäiväiset kurssit on tarkoitettu loppukäyttäjille. Aikaisempi Unix-tietous ei ole välttämätöntä, koska kurssit alkavat Unix-peruskoulutuksesta.



Unix-peruskurssin tärkeimpänä tavoitteena on saada tuntumaa Mikko-laitteistoon, hahmottaa käyttöjärjestelmä ja käyttöliittymä sekä ymmärtää Unixin hierarkkinen hakemis-orakenne ja siinä liikkuminen.

Pääkäyttäjän peruskurssi on Mikko-tuoteperheen tärkein kurssi. Asiakas saa valmiudet selviytyä laitteiston käytöstä arkipäivän tilanteissa. Kurssilla opitaan esimerkiksi varmistusten otto, järjestelmän pysäytys ja käynnistys, tiedostojen rakennus, kirjoittimen ohjaus jne.

Miten selviytyä ongelmista?

Pääkäyttäjän jatkokurssi 1:lla

perehdytään toimintaan muuttuvassa ympäristössä ja harjoitellaan ongelmatilanteista selviytymistä. Esimerkkejä kurssin sisällöstä ovat mm. käytön seuranta, varmistusten palautus ongelmatilanteissa, käyttäjien perustaminen, kirjoittimen asennus, verkko-ympäristössä toimiminen sekä päätepalvelimen parametointi.

Pääkäyttäjän jatkokurssi 2 on tarkoitettu käyttöympäristöihin, joissa on useita koneita ja asiakas itse asentaa ja ylläpitää käyttöjärjestelmiä. Kurssin sisällöstä mainittakoon levyjen alustus, käyttöjärjestelmän asennus, päätepalvelimen asennus, verkko-ympäristössä toimiminen jne.

Unix-jatkokurssilla perehdytään laiteriippumattomasti Unixin ominaisuuksiin. Kurssilla tutkitaan ja harjoitellaan komentotulkkien, prosessien, vi-

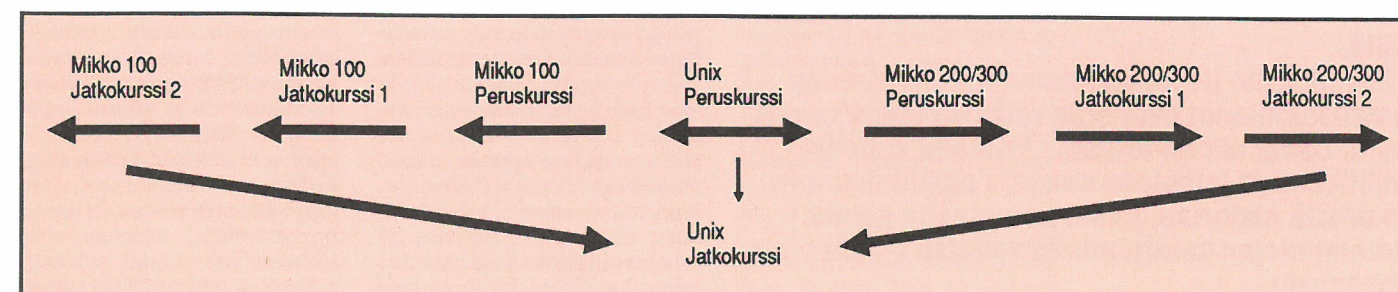
editorin, Unix-mailin jne. hienouksia. Tällä kurssilla huomioidaan kurssilaisten omat erityistarpeet ja sisältö painottuu sen mukaisesti.

Loppukäyttäjän näkökulmasta

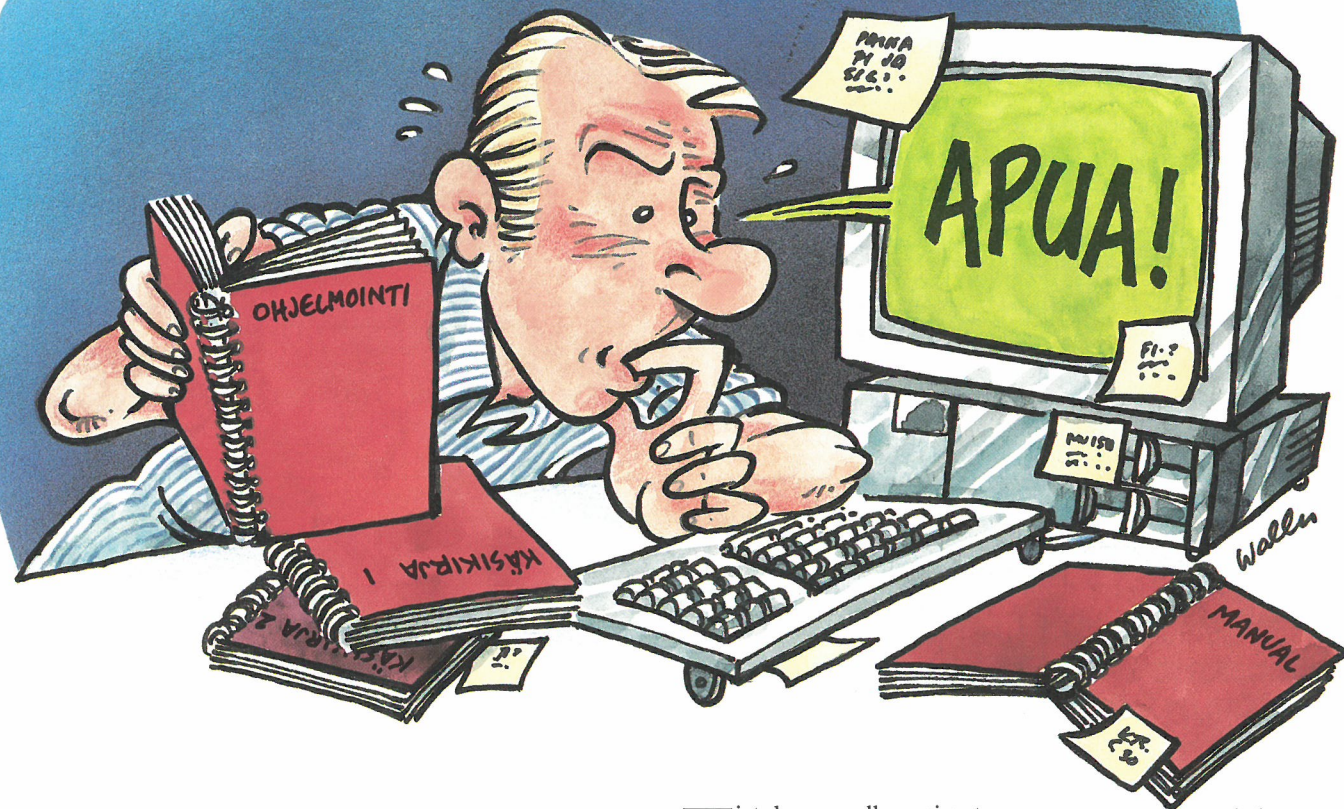
Nitec järjestää myös seminaareja eri aiheista. Keväällä oli Network File System (NFS) -seminaari, jonka päävetäjänä oli Gary Taylor Sun Microsystems UK:sta. Yksipäiväisessä tilaisuudessa perehdyttiin NFS:n ja PC-NFS:n asennointiin ja toimintaan.

Seuraavassa seminaarissa perehdytään Unix-tietoliikenteen problematiikkaan. Siinä painotetaan loppukäyttäjän näkökulmaa aiheeseen eli kurssille ovat tervetulleita kaikki käyttäjät. ■

Unix-koulutuksessa suositellaan etenevistä vaihe vaiheelta. Lisätietoja: Hannu Örn, p. (90) 567 2803.



Pelkkä helppi ei aina riitä



Hyvä ohjelmisto neuvoo käyttäjiänsä

Tämän päivän loppukäyttäjä eroaa edeltäjistään paitsi siinä, että hän ei enää olekaan tietotekniikan saloihin vihkiytynyt atk-ammattilainen, myös siinä että hänellä ei välttämättä ole aikaa eikä halua osallistua käyttökoulutukseen. Hän haluaa kyetä käyttämään uutta sovellustaan ensiyrittämällä.

Sovelluksiin itseensä rakennetut avustus- ja opastustoiminnot ovat eräs ratkaisu tähän ongelmaan. Usein niiden toteutus jätetään kuitenkin puolitiehen ja jatketaan samalla epäonnistumisten pitkää historiaa, jonka perusteella näiden tukitoimintojen kehittäminen voidaan laiminlyödä jatkossakin.

Tietokonesovellus voi antaa käyttäjälle itsestään tietoa monella eri tavalla. Perinteisin menetelmä on oheisdokumentit, jotka edelleen puolustavat paikkaansa käsikirjoina, joista etsitään muistin pettäessä apua yksityiskohtiin. 70-luvulla yleistyneet ohjelmiin sisäänrakennetut avustustoiminnot eli helpit ovat myös yleisessä käytössä, vaikka huonosti tai keskinkertaisesti toteutettu avustustoiminto häviää aina hyvin toteutetuille oheisdokumenteille. Niiden lisäksi sovellusohjelmistoihin kuten tekstoreihin on viime aikoina alkanut ilmestyä opastustoimintoja, jotka toimivat perinteisten opetusohjelmistojen keinoin.

Otetaan esimerkkitapaukseksi loppukäyttäjä, joka on hankkinut ensimmäisen teksturinsa. Avustustoiminto antaa hänelle välitöntä tietoa kulloinkin käytössä olevasta toiminnosta, opastusohjelma opettaa käyttämään koko teksturia ja oheisdokumentit antavat yksityiskohtaista tietoa sen toiminnasta ja toiminnoista. Itse asiassa kaikki eri vaihtoehdot sisältävät käy-

tännössä samat tiedot, mutta tiedon esitystapa ja saatavuus vaihtelee.

Tässä artikkelissa keskitytään lähinnä avustus- ja opastustoimintoihin, joita käsitellään myös yhteisnimekkeellä tukitoiminnot, eikä oheisdokumentteihin viitata kuin mahdollisesti vertailukohtana.

Avustustoiminto eri näkökulmista

Avustustoiminto eli "helppi" vaikuttaa ensi kuulemalla varsin yksinkertaiselta toiminnolta sekä suunnitella että toteuttaa. Sitä se ei kuitenkaan ole, sillä huonosti ja väärin tarpeisiin suunniteltu avustustoiminto hidastaa sovelluksen oppimista huomattavasti ja aiheuttaa turhia virhetilanteita. Avustustoimintoa kehitettäessä on tärkeitä tiedostaa ne tietotarpeet, joita loppukäyttäjä sen avulla pyrkii tyydyttämään, sekä ne vaihtoehdot, joita niiden tyydyttämiseksi on käytettävissä

Sovelluksen käyttäjän lähtötiedot sovellusalueesta, sovellusohjelmasta ja tietotekniikasta voivat vaihdella suuresti, ja avustustoiminnon tulee kyetä reagoimaan näihin vaihteluihin (kuva 1). Tyypillinen esimerkki on vanha tilinpitäjä, joka tuntee sovellusalueen erinomaisesti, mutta ei ymmärrä uudesta mikrosta tai siinä pyörivästä tilinpidosovelluksesta hölläsen polähtävää.

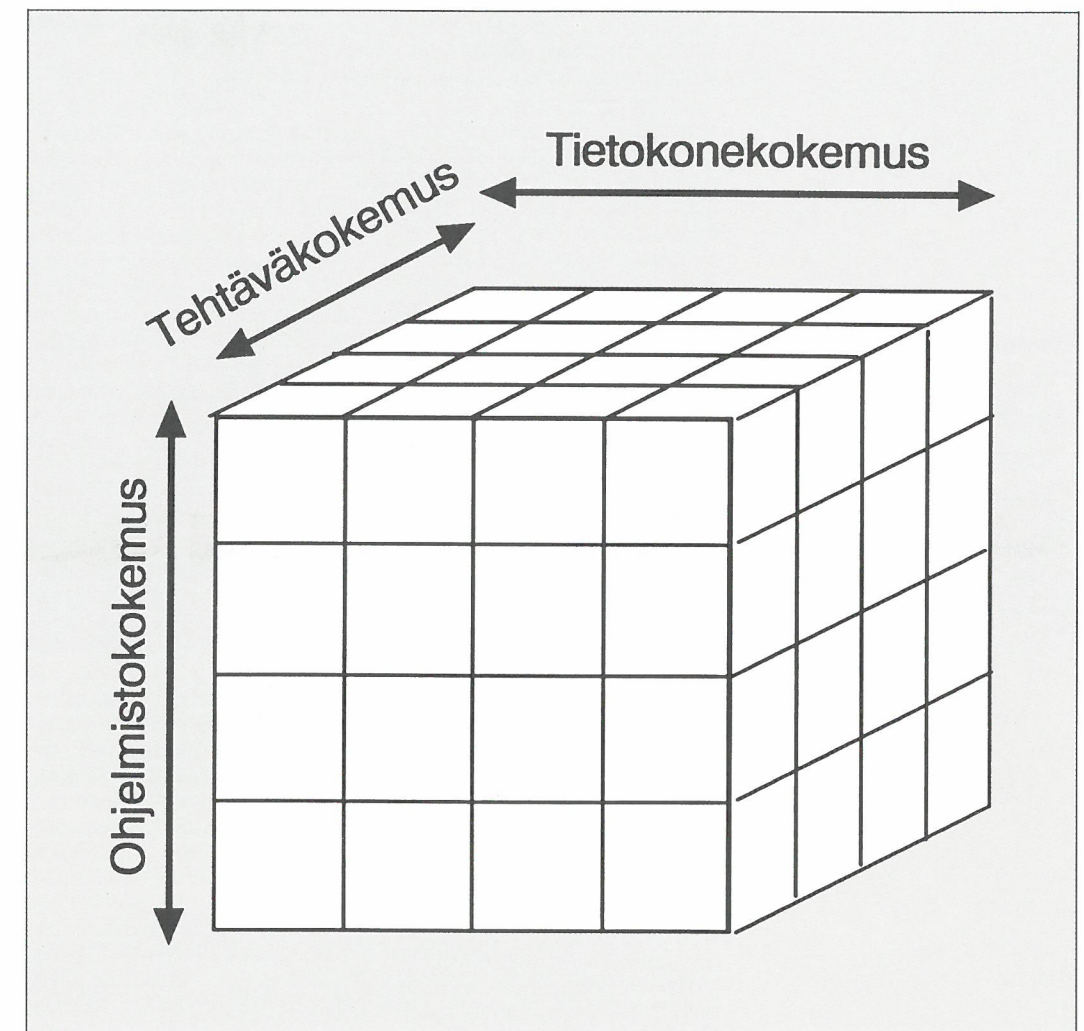
Avustustoiminnon voi käynnistää joko käyttäjä tai sovellus itse. Esimerkiksi virhetilanteiden yhteydessä näytölle tulostuvat selviytymisohjeet ovat hyvä esimerkki sovelluksen käynnistämästä avustustoiminnosta. Käyttäjät eivät kuitenkaan välttämättä pidä järjestelmän heille itseksentantamista viesteistä vaan kokevat niiden häiritsevän työskentelyään. Niinpä jos sovellukseen aiotaan sisällyttää järjestelmän kutsuttavissa olevia avustustuviestejä, käyttäjälle on varattava mahdollisuus niiden esiintymisen kontrolloimiseen (esimerkiksi vain varoitukset tulostuvat näkyviin).

Myös käyttäjän käynnistämien avustustustietojen tasoa on hyvä kyetä säätämään WordStarin tapaan. Tämä sopeuttaa avustustustietoja eri tasoille käyttäjille, jolloin aloittelija saa nähdäkseen kaiken tiedon ja kokenempi käyttäjä tietotasonsa mukaan vaihtelevan määrän informaatiota.

Kolme eri tapaa hallita avustustoimintoja

Avustustoiminto voi valita annettavan tiedonjyvän kolmella eri tavalla. Staattinen avustustoiminto on periaatteessa sovelluksen käyttöohjeen tietokoneistettu versio. Sen etuna painettuun sanaan verrattuna on sen välitön saatavuus ja huomion säilyminen näyttöruudulla koko lukemisen ajan. Se ei kuitenkaan missään määrin hyödynnä tietokoneen kykyä vuorovaikutteiseen tiedon välittämiseen.

Tilannekohtainen (kontekstisensitiivinen) avustustoiminto sen sijaan pyrkii päättämään, mistä aiheesta käyttäjä tarvitsee tai haluaa tietoa. Yksinkertaisimmillaan tämä tarkoittaa tiedon antamista edellisestä toiminnosta tai virhetilanteesta, mutta edellyttää käyttäjältä läh-



tötietoja, joiden perusteella kyseinen edellinen käsky tai virhe on voitu tehdä!

Esimerkiksi Lotus 1-2-3 sisältää tilannekohtaisen avustustoiminnon, jonka käyttäjälle antamien viestien sisältö riippuu ohjelman kulloisestakin tilasta.

Tilannekohtaista avustustoimintoa voidaan pyrkiä tehostamaan asiantuntijajärjestelmällä, joka tuntee paitsi sovelluksen myös käyttäjän. Se yrittää päätellä, mitä käyttäjä on tekemässä ja missä hänellä on ongelmia, ja antaa apua juuri näihin ongelmiin.

Keskusteleva (dialogipohjainen) avustustoiminto sallii sekä ongelman täsmällisen määrittelyn että vaihe vaiheelta etenevän neuvonnan. Keskustelun muotoon rakennettu avustustoiminto voi antaa yksityiskohtaisen kuvauksen kustakin työvaiheesta, odottaa kunnes käyttäjä on saanut kyseisen työvaiheen suoritettua ja jatkaa seuraavasta työvaiheesta.

Luonnollista kieltä ymmärtävän avustustajärjestelmän toteut-

taminen on kuitenkin melkoinen ohjelmointiurakka ja voi osoittautua työläemmäksi kuin itse sovellus.

Aloittelijan housuihin asettuminen

Erityisesti täysin kokemattomille käyttäjille tarkoitetut avustustoiminnot ovat vaikeita toteuttaa, sillä ohjelmien suunnittelijoiden ja toteuttajien on ilmeisen vaikea asettua aloittelijan rooliin.

Kearsley (1988) lainaa useita tutkimuksia, joista osassa avustustoiminnot todetaan kokemattomille käyttäjille täysin sopimattomiksi ja osassa saavutetaan puolestaan erittäin hyviä tuloksia. Ratkaisevaksi tekijäksi näissä tutkimuksissa muodostui Kearsleyn mielestä avustustoiminnon käyttöliittymän ja sisällön laatu.

Esimerkiksi taulukkoon yksi on koottu Magersin (1983) saamat tulokset, kun 30 aloitteli-

Kuva 1: Sovelluksen käyttäjän lähtötiedot sijoittuvat kolmiulotteiseen avaruuteen, ja avustustoiminnot on osattava sopeuttaa niihin.

Avustustoiminnon on kyettävä vastaamaan sovelluksen käytössä kehittyvän käyttäjän jatkuvasti muuttuviin tarpeisiin.

jaa opetteli käyttämään VAX/VMS-käyttöjärjestelmää, jonka avustustoimintoa oli tutkijoiden toimesta paranneltu. Järjestelmään oli lisätty erillinen help-näppäin, ja avustustoiminto oli staattisen avainsanahaun asemesta tilannekohtainen. Avustusviesteihin oli lisätty opastavaa aineistoa, niistä oli karsittu atkslangin sanasto ja ne lähestyivät aihetta suoritettavan tehtävän näkökulmasta. Lisäksi abstraktit esitykset oli käskyjen syntaksin kuvauksissa korvattu esimerkeillä, käskyjen synonyymit oli koottu sanakirjan muotoon ja viestit oli sivutettu vierityksen asemesta.

Kokeneille käyttäjille suunnatut avustustoiminnot eivät ole niin kriittisiä, sillä he osaavat jo suhteuttaa keskinkertaisenkin avustustoiminnon irralliset tiedonjyvät kokonaisuuden viitekehukseen, soveltaa abstraktissa muodossa esitettyä tietoa käytäntöön ja lukea atk-jargonin nimellä tunnettua salakieltä.

Käytännönläheisiä yksityiskohtia

Eräs tapa lisätä avustustoiminnon monipuolisuutta on sallia käyttäjän käsitellä avustustekstejä. Näin hän voi lisätä viesteihin omia muistisääntöjään, jotka hän voi antaa edelleen muille saman sovelluksen käyttäjille. Tämä helpottaa myös avustustoiminnon päivytystä varsinaisen sovelluksen päivityksen tai räätälöinnin yhteydessä.

Käyttäjät voivat auttaa toisiaan myös tietoverkkojen välityksellä. Esimerkiksi sähköpostitaulut voivat toimia keskustelufoorumeina, joilla käsitellään avustustoimintojen ulkopuolelle jääneitä ongelmatilanteita. Lisäksi verkoista löytyvät asiantuntijat voivat antaa neuvoja joko sähköpostitse tai verkon yli keskustellen.

Muita käytännön kysymyksiä ovat käytettävä tiedostorakenne (yksi tiedosto erotusmerkein vai kukin viesti omana tiedostonaan), kirjoitintulostusmahdollisuuden välttämättömyys sekä avustusviestien merkkäminen myöhempää hakua silmälläpitäen.

Taulukko 1: Magersin (1983) parantelmallaan VAX/VMS:n avustujärjestelmällä saamat tulokset suhteessa alkupe-
räiseen avustujärjestelmään.

Opetuskeskustelut eivät toimi

Tietokonesovelluksen käyttöön opastavaa opetusohjelmaa kirjoitettaessa on syytä muistaa, että Kearsleyn lainaamien tutkimusten mukaan tavanomainen opetuskeskusteluohjelma (tutorial) ei tunnu tietotekniikkaa opetettaessa toimivan. Ihmisillä on tapana lähestyä uusia sovelluksia yritys-erehdys-menetelmällä, johon opetuskeskustelun jäykkä rakenne ei yksinkertaisesti istu. Sen sijaan tarvitaan neuvovaa opastusta, joka seuraa oppilaan kehitystä ja antaa neuvoja tarpeen vaatiessa.

Ihanneratkaisuna voitaisiinkin pitää asiantuntijajärjestelmää, joka on paitsi opetettavan sovelluksen myös tietokoneavusteisen työnopetuksen (TATO) asiantuntija (Zachary & al 1988). Asiantuntijajärjestelmä luo opetuksen edetessä mallin oppilaan opetettavasta aiheesta luomasta sisäisestä mallista ja pyrkii muokkaamaan sitä TATO-asiantuntija-osan ohjauksella kohti sovelluksen asiantuntija -osan sisältämää mallia...

Mikäli TATOa käytetään tietotekniikan sovellusten käytön opastamiseen, etuna muihin opastusvaihtoehtoihin nähden on opetusympäristön identtisyys varsinaisen sovelluksen käyttöympäristön kanssa. Sen vaikutusta kannattaa myös tehostaa edelleen käyttämällä mahdollisimman realistisia harjoitustehtäviä opastuksessa annettavien harjoitustehtävien pohjana.

Avustustoiminto oppimisen tukena

Gloria Gery (1987) jakaa sovellusten oppimiskaaren viiteen vaiheeseen, joista kaikkien tukena voidaan käyttää vuorovaikutteisia avustustoimintoja. Gery katsoo käyttäjien tietojen

Alkuperäinen	Paranneltu	Kokonaisaika (min.)
75,6	52,0	Käskyjä kaikkiaan
70,8	98,5	Käskyjä minuutissa
0,99	1,95	Virheellisiä käskyjä
37,9	17,9	Avustustoiminnon käyttökertoja
22,4	60,8	Oheisdokumenttien käyttökertoja
11,1	1,8	Kysyttyjä kysymyksiä

ja taitojen jäävän perinteisten opastusmenetelmien varassa oppimiskaaren toiseen vaiheeseen, sillä ne painottavat oppimistapahtumaa oppimisprosessin asemesta.

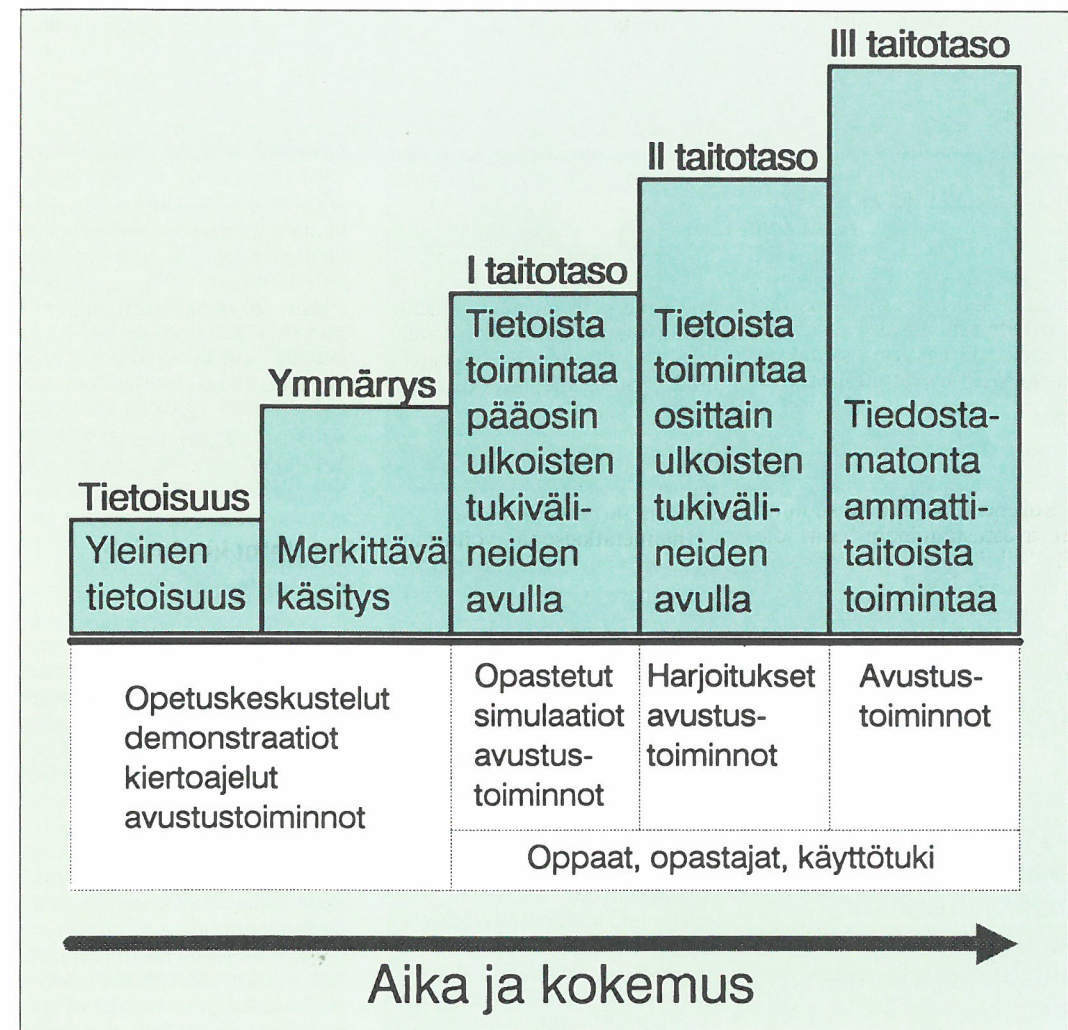
Kuten kuvasta 2 käy ilmi, Geryn mielestä perinteisillä tietokoneavusteisen työnopetuksen (TATO) keinoilla ei päästä kuin oppimisen toiselle portaalle asti. Älykkäät simulaatiotkaan eivät hänen näkemyksensä mukaan riitä kuin kolmannelle portaalle. Siitä eteenpäin ainoa apukeino on hyvin toteutettu vuorovaikutteinen avustustoiminto.

Opastettu kiertoajelu antaa perustiedot

Eräs mielenkiintoinen tapa toteuttaa sovelluksen perustietojen opettaminen on luoda niin kutsuttu opastettu kiertoajelu (Gresham & al 1987). Kiertoajelu laaditaan varsinaisen sovelluksen päälle siten, että kiertoajeluohjelmisto ohjaa sovelluksen etenemistä käyden lävitse sovelluksen perustoiminnot ja mahdollisuudet. Kiertoajeluun voidaan ympätä myös orientoivaa lisätietoa esimerkiksi ikkunoinnin tai erillisen ääninauhan tai digitoidun puheen muodossa.

Lisätiedot tosin hidastavat kiertoajelun kehittämistä, sillä yksinkertaisen kiertoajelun luomiseen riittää ohjelma, joka sieppaa kiertoajelun laatijan sovellukselle antamat käskyt tiedostoon. Nämä käskyt välitetään edelleen sovellukselle siten, että se kuvittelee niiden tulevan suoraan käyttäjältä, vaikka käyttäjänä onkin tällä kertaa ennalta laadittua kiertoajelua seuraava oppilas.

Mikäli kyseessä on kehittynyt kiertoajelu, oppilaan on kyettävä keskeyttämään sen eteneminen koska tahansa ja päästävä kokeilemaan sovelluksen käyttöä itsenäisesti. Näin hänen ei tarvitse istua passivisena tark-



Kuva 2: Glora Geryn (1987) viisivaiheinen tietokonesovelluksen oppimiskaari sekä keinot, joilla nousta portaalta toiselle.

kailijana kiertoajelun alusta loppuun, vaan hän voi kokeilla oppimansa oikeellisuutta välittömästi ja kerrata osia kiertoajelusta tarpeen niin vaatiessa.

Tarjolla luulon lisäksi myös tietoa

Tietokonesovellusten käytön opastaminen sovellusten itsensä kautta on siis mahdollista, mutta vaatii sovelluksen suunnittelijoilta huomattavaa panosta sovelluksen avustustoiminnon kehittämiseen. Pelkkä kunkin toiminnon päälle jälkikäteen liimattu parin rivin pituinen viesti ei riitä, vaan avustustoiminnon on kyettävä vastaamaan sovelluksen käytössä kehittyvän käyttäjän jatkuvasti muuttuviin tarpeisiin. Ei hän ensimmäisellä käyttökerralla halua tietää kaikkea vaan vain kaikkein oleellisimman, kuukauden kuluttua hänen kysymyksiinsä on kuitenkin vastat-

tava aivan eri tavalla.

Avustus- ja opastustoimintojen toteuttamista ei saa myöskään irroittaa itse sovelluksen kehityskaaresta, sillä hyvä tukitoiminto vaatii onnistuakseen kaikki ne suunnittelu- ja työvaiheet mitä itse sovelluskin. Tukitoimintoja ei saa unohtaa myöskään sovellusta päivitettäessä tai edelleen kehitettäessä, vaan niiden on kehitettävä itse sovelluksen mukana.

Tärkeätä on toki myös vetää raja, jonka jälkeen sovelluksen avustustoimintoon ei enää kannata uhrata aikaa tai vaivaa. Muuten avustustoiminto voi paisua käenpojaksi, joka potkii itse sovelluksen pois pesästä.

Kaikkeksi onneksi myös tutkijat ovat oivaltaneet tukitoimintojen merkityksen. Niinpä ohjelmoijien ei tarvitse enää turvautua pelkkään näppituntumaan avustus- ja opastustoimintoja suunnitellessaan, kunhan muistavat Kearsleyn kokoaemien esimerkkien opetuksen: mitään tutkimustuloksia ei saa niellä pureskelematta. ■

Kirjallisuutta:

- Elkerton J. (1988) Online Aiding for Human-Computer Interfaces. Teoksessa: Handbook of Human-Computer Interaction, Elkerton J. (ed.), Amsterdam: North Holland. 345-364.
- Gery G. (1987) Making CBT Happen. Boston: Weingarten Publications.
- Gresham S. F., Ferguson, L. R. & Allen B. S. (1987) Journaling Systems: Integrating Computer-Based Learning Activities with Applications Software. ADCIS 29th Conference Proceedings, 7-13.
- Kearsley G. (1988) Online Help Systems - Design And Implementation. Norwood, New Jersey: Ablex Publishing Corporation.
- Magers C. S. (1983) Experimental Evaluation of Online HELP for Non-Programmers. ACM CHI'83 Proceedings, Boston. 277-281.
- Zachary W., Glenn F. & Hicinbothorn J. (1988) Architectures for Intelligent Computer-Assisted Embedded training. ADCIS 30th Conference Proceedings, 297-304.

Olioita ohjelmiin

Olio-orientoituneesta ohjelmoinnista on muodostunut viime vuosien kuluessa todellinen muoti-ilmiö. Mutta mitä olio-ohjelmointi oikein tarkoittaa, ja miten eräs suosituimmista olio-orientoituneista ohjelmointikielistä eli C++ eroaa muista kielistä? Ainakin se käy hyvin kau-paksi, sillä esimerkiksi Microsoftin suosittamaa Glockenspieliä on jo viime kesänä myyty noin 50 000 kopiota kuukaudessa.

Nyt alkavassa artikkelisarjassa selvitetään joitakin C++-kielen tärkeimpiä piirteitä konkreettisin esimerkein. Artikkelisarja on kohdistettu lähinnä jo C-kieltä taitavalle lukijalle, mutta varsinkin tämä ensimmäinen osa antaa olio-tietoutta myös laajemmalle yleisölle.

Mitä muutoksia C++ sitten tuo C-ohjelmoijan työhön? Noin 95 % C++:n syntaksista on yhteneväistä C-kielen kanssa eli kieliopin puolesta siirtyminen on helppoa. Itse asiassa useimmat C-ohjelmat ovat sinällään täysin laillisia C++-ohjelmia. C-kielen perusosa on siis sama, mutta siihen on lisätty oliomaailman luokkia käyttävät tuntematta muita pienehköjä parannuksia.

Olioajattelu ei ole mitään uutta. Ohjelmoijathan ovat kautta aikojen käyttäneet number-olioita. Number-tyyppihän tarkoittaa tiettyä arvoaluetta ja matemaattista laskentaa. Tätä tyyppiä ohjelmoijat käyttävät tuntematta välttämättä tarkemmin sen rakennetta tai laskennan toteuttamistapaa. Näin he voivat keskittyä tärkeämpiin asioihin, mikä onkin olioajattelun keskeinen tavoite.

Vanhoillakin työvälineillä ohjelmointityötä voi tehdä olioajattelun suuntaisesti järjestämällä ohjelmansa osat sen mukaisesti ja käyttämällä siihen sopivia nimeämistapoja. Uudet työvälineet vain helpottavat uusien abstraktien tyyppien luomista ja hallintaa.

Tässä artikkelisarjan osassa keskitytään C++-kielen uusista ominaisuuksista tärkeimpään, luokkiin ja olioihin.

C++:n historia ja nykypäivä

C++-kielen kehitti 1980-luvun puolivälissä Bjarne Stroustrup tutkittuaan perinteisten kielten rakennetta AT&T:n Bellin laboratorioissa noin seitsemän vuoden ajan.

AT&T otti C++-kielen C-kielen viralliseksi jatkajaksi. Niinpä AT&T kehitti C++-kielelle oman de facto -standardinsa, jonka pohjalta ANSI laati oman OOP C -standardinsa. Tällä hetkellä johtavat C++-kääntäjien valmistajat ilmoittavat kilvan tuotteidensa mukaillevan AT&T:n C++-kääntäjän 2.0-versiota.

Kaksi johtavinta C++-tuotetta ovat tällä hetkellä Glockenspiel C++ sekä Zortech C++. Näistä ensin mainittu on esikääntäjätyyppinen ja tuottaa siis

C-kielistä lähdekoodia, joka on vielä käännettävä C-kääntäjällä. Glockenspielin etuna on se, että se on integroitu käytettäväksi myös Microsoftin C 6.0 -ohjelmointiympäristössä. Zortech C++ on puolestaan suoraan objektikoodia tuottava kääntäjä. Näiden tuotteiden lisäksi markkinoilta löytyy useita muitakin C++-nimeä kantavia tuotteita, kuten esimerkiksi Guidelines C++, Oregon C++ ja Borlandin Turbo C++.

Oliokielen periaatteet

Kun sovellus aiotaan toteuttaa olio-ohjelmointikielellä, mitenkysymysten sijaan täytyy miettiä vastauksia mitä-kysymyksiin: Mitä olioita sovellus käsittelee? Mitä palveluja olio tarjoaa? Miten oliot kommunikoivat keskenään?

Ohjelmointikielen onkin toteutettava vähintään seuraavat neljä asiaa ollakseen olio-suuntautunut.

Ensinnäkin kielen on toteutettava tiedon abstrahointi (data abstraction), joka tarkoittaa sitä, että tiedon käyttäjän ei tarvitse tietää olion toteutuksesta mitään voidakseen käyttää oliota. Esimerkkinä voisi olla aikaisemmin mainittu number-tyyppi. Käytännön elämän esimerkkinä olkoon auto, jota kyllä useimmiten osataan ajaa, vaikkei välttämättä tiedetä auton tekniikasta mitään.

Tiedon abstrahointiin liittyy myös se, että paitsi olion tietorakenne, myös kaikki siihen kohdistettavissa olevat toiminnot (palvelut) kootaan yhteen. Tällaista olion tyyppimäärittelyä kutsutaan C++-kielessä luokaksi (class). Luokka siis toimii abstraktin tietotyypin mallineena (template). Seuraavassa käytetään nimitystä olio tarkoittamaan olion tyyppimäärittelyä eli luokkaa.

Oliolle määriteltyjä toimintoja kutsutaan nimellä metodi. Esimerkiksi auto-olion metodeja voisivat olla Käynnistys, Liikkuminen, Kääntäminen jne. Olion omia metodeja kutsutaan myös jäsenfunktioiksi (member functions).

Toisinaan oliot erotetaan siten, että puhutaan erikseen oliosta tai luokasta ja toisaalta

olion/luokan ilmentymästä (instance) eli esimerkissämme toisaalta autosta yleisenä käsitteenä ja toisaalta sen fyysisistä toteutuksista, joita näemme päivittäin kaduillamme.

Oliot kommunikoivat keskenään lähettämällä sanomia. Sanomien avulla voidaan esimerkiksi pyytää toista oliota tulostamaan itsensä tai suorittamaan jonkin laskutoimituksen. Sanoman lähettäneen olion ei tarvitse välttämättä tietää lainkaan, miten sen pyytämä palvelu toteutetaan.

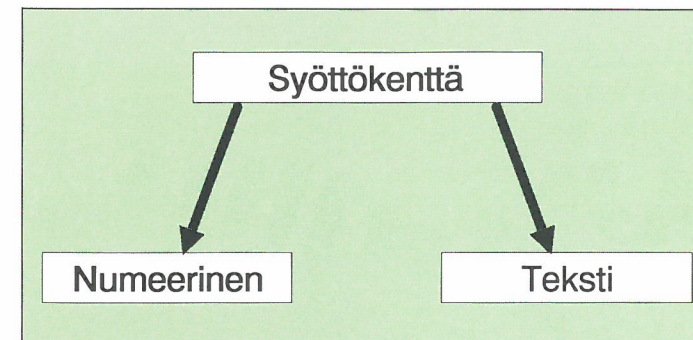
Toiseksi kielessä on oltava mahdollisuus tiedon kätkenään (encapsulation, information hiding). Tiedon kätkemisen tarkoituksena on suojata olion käyttäjältä. Toisin sanoen olion tietorakenteisiin päästään vain oliossa määriteltyjen metodien avulla. Osa metodeista voidaan määritellä myös olion yksityiseksi ominaisuudeksi, jolloin niitä ei voida käyttää olion ulkopuolelta.

Kolmanneksi olio-ohjelmointikielen tulee mahdollistaa olioiden luokkahierarkia ja periytyminen. Tämä tarkoittaa sitä, että uuden olion voi rakentaa toisen pohjalta. Tällöin uutta oliota kutsutaan nimellä aliluokka (derived type, descendant, child instance, subclass) ja se sisältää omien, erityisten piirteidensä lisäksi myös kantaluokalta (base class, parent class) perityt ominaisuudet.

Periytymisestä voisi olla esimerkkinä vaikkapa näytön ikkunassa oleva syöttökenttä (kuva 1), joka toimii kantaluokkana. Sen aliluokkia ovat numeerinen ja tekstisyöttökenttä, jotka toimisivat samoin kuin kantaluokka (piirtyvät näytölle tietyn näköisinä, ottavat vastaan käyttäjän syötettä, kaiuttavat sen näytölle jne.), mutta nämä aliluokat hyväksyisivät vain tietyn muotoista syötettä tai olisivat näytöllä hieman eri näköisiä.

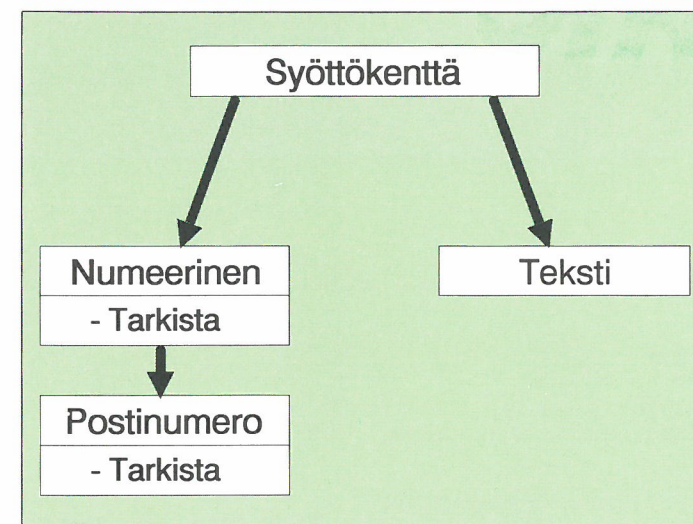
Neljäntenä ominaisuutena olio-ohjelmointikieliltä vaaditaan polymorfismia. Se tarkoittaa käytännössä sitä, että luokkahierarkian eri tasoilla voidaan käyttää esimerkiksi samannimisiä funktioita, jotka toimivat eri tavoin.

Kuvassa (2) on esimerkki polymorfismista. Siinä sekä Numeerinen syöttökenttä -oliassa



C ! = C++

Kuva 1: Esimerkki luokkahierarkiasta.



Kuva 2: Esimerkki polymorfiasta.

Olio-ohjelmointikieliltä vaaditaan polymorfismia: luokkahierarkia eri tasoilla voidaan käyttää esimerkiksi samannimisiä funktioita, jotka toimivat eri tavoin.

että Postinumero syöttökenttä -oliassa on käytetty samannimistä Tarkista-metodia. Ensimmäinen vain tarkistaa, että syöte on numeerista, kun taas jälkimmäinen tarkistaa tämän lisäksi myös sen, että syöte koostuu korkeintaan viidestä numerosta.

Muitakin piirteitä olio-ohjelmoinnista löytyy, mutta tarkastellaamme nyt, miten C++ toteuttaa nämä tärkeimmät neljä piirrettä.

C++:n luokkakäsite

Luokkakäsite voidaan C++-kielessä hoitaa parhaiten class-

määrittelyllä, josta on esimerkki kuvassa (3). Siinä määritellään luokka opiskelija, johon kuuluvia olioita voidaan perustaa (luo) ja tulostaa näytölle (nayta). Kuvassa (3) esimerkissä nähdään luokan metodien määrittelyt, jotka kirjoitetaan muuten kuten C-kielen funktioiden määrittelyt, mutta ensimmäisellä rivillä liitetään metodi omaan luokkaansa kirjoittamalla:

void opiskelija::nayta()

eli luokan nimellä, jota seuraa kaksi kaksoispistettä.

Luokkamäärittelyssä näkyy tiedon abstrahointi siten, että perinteisestä struct-määrittelystä poiketen class-määrittelyssä ku-

Ohjelmien laatu paranee ja luotettavuus kasvaa siksi, ettei luokkien sisäistä rakennetta pääse muuttamaan muualta kuin kyseisen luokan toimintojen kautta.

```
#include <stream.h>
#include <string.h>
// Määritellään kantaluokka

class opiskelija {
    char *nimi; // opiskelijan nimi
    int vuosi; // monesko opiskeluvuosi
public:
    void nayta(); // näyttää nimen ja vuoden
    void luo (char *, int); // anna oliolle alkuarvot
};

void student::nayta()
{
    cout << form ("nimi: %s vuosi: %d\n", nimi, vuosi);
    // tulosta näytölle nimi ja vuosi
}

void student::luo (char *n, int v)
{
    nimi = new char[strlen(n)+1];
    // varaa muistia sopiva pala
    strcpy (nimi, n);
    //kopioi parametrina saatu nimi luokkaan
    vuosi = v; //ja parametrina annettu vuosi luokkaan
}
```

Kuva 3: Luokkamäärittely C++-kielessä.

Kuva 4: Oliopohjaisen ratkaisun periaate.

vataan paitsi tietorakenteet, myös luokan olioihin kohdistettavissa olevat toimenpiteet eli funktiot (metodit).

Kuvan (3) määrittelyssä näkyy selvästi luokan jako kahteen osaan: alkuosa on luokan sisäistä tietoa (private) ja public-otsakkeen jälkeinen osa näkyy luokan ulkopuolellekin. Myös metodeja voitaisiin piilottaa muualta ohjelmalta. Public-osa toimii näin eräänlaisena käyttöliittymänä muun sovelluksen ja olion välillä.

Luokkahierarkia C++-kielessä

Kuvitellaan, että edellä määritellyn opiskelija-luokan lisätiedoiksi tarvittaisiin vielä osoitetiedot. Ne voitaisiin tietenkin lisätä perinteisesti samaan luokkamäärittelyyn nimi- ja opiskeluvuosi-lisäksi. Entä miten käsiteltäisiin niiden opiskelijoiden tiedot (talon nro, huoneen nro), jotka asuvat sisäoppilaitoksessa.

Perinteisessä ratkaisussa jouduttaisiin ehkä määrittelemään kaksi eri tietorakennetta: toinen kotonaan asuville ja toinen sisä-

oppilaitosoppilaille. Tällöin molempiin tarvittaisiin oppilaiden tunnistetiedot (nimi, vuosi). Lisäksi jouduttaisiin korjaamaan kaikki ne funktiot, joissa kyseisiä rakenteita käytetään.

Oliopohjaisen ratkaisun periaate on nähtävissä kuvassa (4). Siinä käytetään hyväksi valmiiksi määriteltyä opiskelija-luokkaa, josta peritään kaikki ominaisuudet uusiin asuntolassa ja kotona-luokkiin.

Kuvassa 5 nähdään osa ratkaisusta C++-kielellä toteutettuna. Siinä määritellään asuntolassa-luokka, joka on kuvassa (3) määritellyn opiskelija-luokan aliluokka. Määrittelyssä tämä ilmaistaan ensimmäisellä rivillä:

```
class asuntolassa : opiskelija
```

Kotona-luokka määriteltäisiin vastaavasti.

Tällaisella toteutuksella järjestelmä saadaan joustavaksi myös tulevien muutoksien tekijöiden kannalta.

Esimerkin vuoksi kuvassa (6) on vielä esitetty, miten edellä määriteltyjä opiskelija- ja asuntolassa-luokkia käytettäisiin C++-ohjelmassa.

Olio-ohjelmoinnin edut

Olio-ohjelmoinnin käyttöönotto on perustava kohennus ohjelmoinnin työtapoihin. Koska C++-kääntäjä on paljon tarkistuksissaan tarkempi kuin C-kääntäjä, ohjelmointi on aluksi hankalampaa. Mutta kun vauhtiin päästään ja saadaan valmiiksi joukko luokkia, ohjelmointi helpottuu huomattavasti perinteisestä. Tämän jälkeen käytetään valmiita luokkia pohjana, ja ohjelmointityö on lähinnä sopivan luokan etsimistä.

Ohjelmien laatu paranee ja luotettavuus kasvaa sen vuoksi, ettei luokkien sisäistä rakennetta pääse muuttamaan muualta kuin kyseisen luokan toimintojen kautta (encapsulation).

Ohjelmien ylläpito helpottuu huomattavasti, kun voidaan keskittyä vain yhden olion toimintaan kerrallaan. Jos esimerkiksi mikään taulukon lokeroista ei toimi oikein, tarkistetaan ylimmän tason lokero-olion luokka, jos vain tekstilokeroissa on vikaa, tarkistetaan vain tekstilokero-olion luokka jne.

Ohjelmien laajennettavuus ja tuottavuus paranee mm. siten, että yrityksen sisällä kerätään valmiiksi määriteltyjä luokkia luokkakirjastoihin, joista ne ovat helposti kaikkien hyödynnettävissä. Tämä ajatus laajentaa sitä käytäntöä, jolla C-kielen funktioita on tähän asti kirjastoitu. Nyt pelkkien funktioiden asemesta kirjastoidaan kokonaisia olioita tietorakenteineen ja funktioneen. Tällaisia valmiita luokkakirjastoja on myös myytävänä, esimerkiksi irlantilaisen Glockenspielin Common-View, jonka avulla voi tehdä Windows- ja Presentation Manager -ohjelmia.

Myös ohjelmien siirrettävyys laiteympäristöstä toiseen paranee C++-kielen käytön myötä.

Jo valmiina olevia C-kielisiä funktioita voidaan hyödyntää suoraan C++-kielen metodeissa. Kaiken kaikkiaan saattaa kuitenkin kulua vuosia, ennen kuin C++:n kaikki piirteet ovat yrityksissä täysimittaisessa käytössä ja samaan aikaan käytetään vielä pitkään perinteisiä ohjelmointikieliä. Kuitenkin C++:n tarjoamien pitkän tähtäimen etujen vuoksi sen käyttöä kannattaa vakavasti harkita. ■

```
// Määrittele opiskelija-luokalle aliluokka

class asuntolassa:opiskelija {
    char *talo;
    char *huone;
public:
    void nayta_as ();
    // näyttää asuntola-opiskelijan tiedot
    void uusi_opiskelija (char *, int, char *, char *);
    // uuden asuntola-opiskelijan olion luonti
};

void asuntolassa::nayta_as()
{
    nayta(); // näytä kantaluokan tiedot
    cout << form ("talo: %s, huone: %s\n", talo, huone);
    // näytä asuntola-tiedot
}

void asuntolassa::uusi_opiskelija ( char *n, int v, char *t, char *h)
{
    luo (n, v); // aseta kantaluokalle tiedot
    talo = new char [strlen(t) + 1];
    // varaa sopiva muistipala aliluokalle
    strcpy (talo, t);
    // kopioi parametrina annettu t luokan talo-tiedoksi
    huone = new char [strlen (h) + 1];
    strcpy (huone, h);
}
```

Kuva 5: Olioluokan määrittely C++-kielessä.

Kuva 6: Hierarkkisten luokkien käyttö C++-kielessä.

```
// Yksinkertainen esimerkki luokkien kytöstä

main()
{
    student x; // kantaluokkaan kuuluva olio x

    x.luo ("Anni Kirjanen", 1);
    // alkuarvot oliolle x
    x.nayta (); // näytä olion x arvot

    asuntolassa y; // aliluokkaan kuuluva olio y

    y.uusi_opiskelija ("Lauri Lehti", 1, "ALATALO 1", "V155");
    // alkuarvot oliolle y
    y.nayta_as (); // näytä olion y arvot
}
```



Tiina Mikonmäki toimii Tietospesialistit Oy:ssä vastaavana asiantuntijana. Hänet tavoittaa puhelinnumerosta (915) 49 388 tai smailitse Mikonmäki Tiina frl.

00003 HELSINKI

Sopimus 00380/166
VASTAUSLAHEYS

PL 780/10
Softari
Nokia Data

Palvelukortti

Softari 3/90

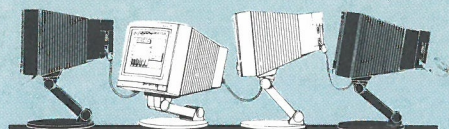
Kiinnostavaa tässä lehdessä oli

Turhaa oli

- ☐ Korjaan yllämainitut osoitetiedot.
☐ Poistan yllämainitut osoitetiedot
☐ Lisään allamainitut osoitetiedot
☐ Netin ☐ Softarin rekisteristä/rekisteriin
☐ Haluan lehden kotiosoitteeseeni

yritys _____
 osasto _____
 lähiosoite _____
 postinro _____ postitoimip. _____
 nimi _____

Nokia
maksaa
postimaksun



T i i m i -tuotteet

Tiimi on Nokia Datan avoin, hajautettu työasemajärjestelmä. Siihen on kehitetty useita eri toimialojen sovelluksia. Nokia Data on rekisteröinyt seuraavat Tiimi-tavaramerkit:



ToimistoTiimi

ToimistoTiimi on toimistojärjestelmä, johon on Nokian kehittämien peruspalvelujen kuten sähköpostin ja arkiston lisäksi valittu toimiston tehtäviin parhaiten sopivat ohjelmistot. Ne kuuluvat graafiseen Windows- tai merkkipohjaiseen Perfect-perheeseen. ToimistoTiimi yhdistää ergonomiset ja älykkäät MikroMikot työasemajärjestelmäksi, jonka yhteiset palvelut ovat kaikkien työasemien käytettävissä.



VerstaTiimi

VerstaTiimi on tehtaille ja tuoteverstoille kehitetty tuotannonohjausjärjestelmä. Sen avulla tuotannon läpäisyajat lyhenevät murto-osaan aikaisemmasta. Yrityksen reagointi- ja toimituskyky paranevat huomattavasti. Edistyskellisiä ohjelmistotyövälineitä käyttämällä järjestelmä sovitetaan asiakkaan muihin tietojärjestelmiin.



LakiTiimi

LakiTiimi on asianajotoimistojen ja yritysten lakiasiaanosastojen työtä tukeva ohjelmisto sekä juristin päivittäinen työkalu. Ohjelmistopakettiin on sisällytetty kaikki lakiasioiden kannalta olennaiset toiminnot.



VarastoTiimi

VarastoTiimi on järjestelmä, jolla ohjataan ja seurataan varastoa ja kuljetuksia ja pidetään niiden kustannukset kurissa. VarastoTiimi kuten muutkin Tiimit voidaan liittää jo käytössä oleviin järjestelmiin.



MyyntiTiimi

MyyntiTiimillä ohjataan ja seurataan myynnin ja markkinoinnin suunnittelua ja toteutusta. MyyntiTiimin avulla tiedot tuotteista, asiakkaista, kilpailijoista ja muista sidosryhmistä ovat nopeasti käytettävissä.



JohdonTiimi

JohdonTiimi on päätöksentekijöiden käyttöön suunniteltu raportointijärjestelmä, joka kokoaa tietoa yrityksen muista tietojärjestelmistä ja esittää sen havainnollisessa muodossa.



TeleTiimi

TeleTiimi on teleyritysten liiketoimintaa palveleva työasemajärjestelmä, joka sisältää erityisesti markkinointia, johtoa ja asiakaspalvelua tukevia sovelluksia.



OppilaitosTiimi

OppilaitosTiimi koostuu mikroista, jotka voidaan yhdistää opetusverkoksi, sekä sähköpostista ja arkistosta. Järjestelmää voidaan kasvattaa tarpeen mukaan ja se voidaan liittää koulun ulkopuolisiin järjestelmiin, esimerkiksi kouluviraston järjestelmään.