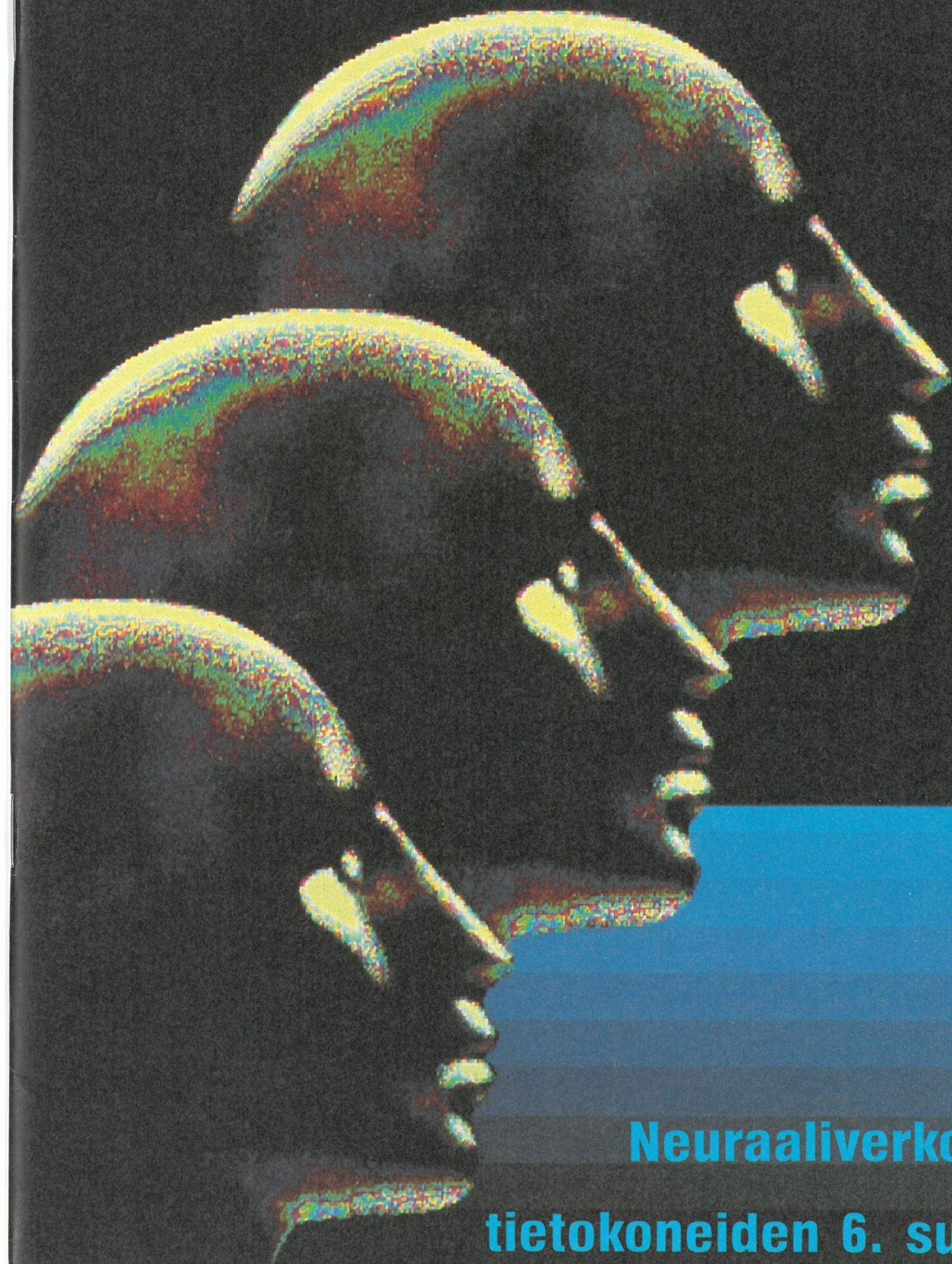


Softari

NOKIA DATAN

OHJELMISTOLEHTI 3/89



Neuraaliverkot

tietokoneiden 6. sukupolvi

Nokia Datan ohjelmistolehti

Päätoimittaja:

Terhi Ogbeide,
puh. (90) 567 2994

Toimitus:

Jyrki J. J. Kasvi,
puh. (90) 567 4064

SoftariTiimi:

Mikko Helander, Kari Kaipainen,
Outi Kuusela, Rami Raulas,
Tapio Sulkava, Anne Typpö,
Esko Uotila

Ulkoasu:

Jyrki J. J. Kasvi

Osoitteenmuutokset:

Takakannen palvelukortilla

Toimituksen osoite:

Nokia Data/Softari
PL 780
00101 HELSINKI

Painosmäärä:

14 200

Kirjapaino:

Helsingin Liikekirjapaino Oy

Julkaisija:

Nokia Data

ISSN 0782-8209

Sisällys:

Muistoja putkien aikakaudelta	2
Neuraaliverkot matkivat ihmisaivoja	3
Työryhmäverkot ja keskuskoneet yhteen	6
Tiimi hajauttaa ja yhdistää	7
Tandem ja Tiimi yhteen	8
Liikettä linjoilla	10
Kilpailijat saman pöydän ympärillä	12
Käytössä koettua	14
SyliMikko kulkee mihin vaan	15
Mikro ei ole tuottavin investointi	16
Väriä elämään	19
Learning Ada — Part 5	21
SQL Windows-kurssi — Osa 3	24
Tuoteuutiset	30
Lukijakysely 2/89	31
Mikkonen	32

Muistoja putkien aikakaudelta

Kun John von Neumann kehitti 40-luvulla kuuluisan tietokonearkkitehtuurinsa, hän tuskin uskoi sen vielä yli 40 vuotta myöhemmin hallitsevan markkinoita. Kuitenkaan esimerkiksi MikroMikko 4 ei ole perusrakenteeltaan kovinkaan kaukana von Neumannin, Eckertin ja Mauchlyn 40-luvulla suunnittelemasta Edvacista (valmistui 1952).

von Neumannin arkkitehtuurin tärkein etu on sen yksinkertainen joustavuus. Se on pysynyt voimissaan elektroniputkien ajalta aina VLSI-piireihin asti. Vaikka sen puutteet on tunnettu jo pitkään, silkka markkinavoi-
mien inertia on pitänyt sen voimissaan. Resurssit eivät ole riittäneet vaihtoehtojen tutkimiseen, kun vanhaakin on voitu kehittää.

Nyttemmin tilanne on kuitenkin muuttumassa, sillä puolijohteiden fyysiset rajat alkavat tulla vastaan, mutta tietojenkäsittelykapasiteettia tarvitaan aina vain lisää. Eri puolilla maailmaa kehitetäänkin rinnan optisia piire-

jä, rinnakkaissuorittimia, bio-piirejä ja neuraaliverkkoja. Kun nämä pääsevät Edvac-vaiheestaan eteenpäin, edessä on melkoinen myllerrys.

Tähän asti tietokoneet ovat kyyenneet lähinnä helpottamaan vanhoja rutiineja kuten tietoliikennettä ja toimistotyötä. Suorituskyvyn räjähdysmäinen kasvu tuo tietotekniikan myös aivotyöhön. Jo nyt jotkin asiantuntijajärjestelmät pystyvät samaan tai parempaan kuin koulutettu ihminen.

Toivottavasti olemme oppineet viimeisistä kymmenestä vuodesta jotain, sillä tähän asti tietoyhteiskuntaan siirtyminen on ollut yhtä hallitsematonta rakennemuutosta. Kehitys jatkaa kuitenkin kehittymistään, ja meidän sukupolvemme tietotaito alkaa vanhentua sillä hetkellä kun jätämme koulun penkin. Asenteet kehittyvät vielä hitaammin kuin tiedot.

Tekniikka tuskin säilyttää asemansa kehityksen tahdin määrääjänä. Ihminen on tulevaisuuden jarrumies — hyvässä tai pahassa.

J. J. Kasvi



Neuraaliverkot matkivat ihmisaivoja

Tietokonetekniikassa on käynnissä valankumous. Toistaiseksi kysymys on ajattelutavan muutoksesta. Muutaman vuoden kuluttua neurotietokoneet eli keinotekoiset hermoverkot vaikuttavat jo käytännön elämässä.

Laboratoriot ja tutkimuslaitokset maailman eri puolilla soveltavat uutta tietotekniikkaa kuvankäsittelyyn, puheentunnistukseen, robotiikkaan ja laskennallisesti vaativiin optimointiongelmiin. Mikropiirivalmistajat julkistavat erikoiskomponentteja neurotietokoneiden rakennusosiksi. Ensimmäiset kaupalliset sovellukset ovat jo käytössä.

Miksi neuraaliverkkoja rakennetaan?

Biologiset hermoverkot ovat hitaita. Hermosolussa viestit kulkevat muutamia kymmeniä senttimetrejä sekunnissa. Tietokoneen elektroniset piirit ovat miljoonia kertoja nopeampia. Siitä huolimatta biologiset hermoverkot pystyvät hämmästyttäviin saavutuksiin. Ihminen pystyy tunnistamaan valokuvassa näkyvät kasvat noin sekunnin kymmenesosassa. Nopeimmatkaan keinotekoiset hahmontunnistulaitteet eivät pysty samaan. Hermoverkkojen ilmeisin ominaisuus, niiden erittäin suuri rinnakkaisuus, on siis erittäin tehokas tapa kompensoida prosessointielementtien hitautta.

Ihmisen aivoissa on useimpien arvioiden mukaan ainakin tuhat miljardia hermosolua (10¹²). Kukin hermosolu on kytkeytynyt keskimäärin noin kymmeneentuhanteen muuhun hermosoluun. Hermosoluja tuhoutuu tuhansittain joka päivä, mutta aivojen vikasietoisuus pysyy erinomaisena.

Biologiset hermoverkot oppivat. Niitä ei tarvitse ohjelmoida. Monet toteutetuista keinotekoisista hermoverkoista ovat pystyneet yleistämään niille annetut esimerkkitapaukset ilman, että ihmisen olisi ollut pakko kuvata esimerkkien yhtäläisyyksiä ja

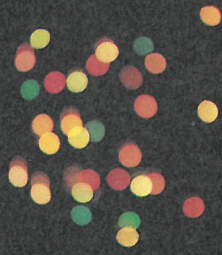
eroavuuksia. Tällainen yleistäminen on kaiken oppimisen pohjana, mutta sitä on erittäin vaikea toteuttaa tavanomaisin tekoälyn menetelmin.

Hermosolut ovat melko yksinkertaisia. Nykyaikaisella VLSI-teknikalla voidaan valmistaa mikropiirejä, joissa on suuri määrä keinotekoisia hermosoluja. Federico Faggin ja Carver Mead, viime vuosikymmenellä puolijohdevalmistuksen mullistaneet tutkijat ovat perustaneet yrityksen, joka jo nyt valmistaa piistä silmän verkkokalvon prototyyppijä. He uskovat, että esimerkiksi lasisilmään voidaan rakentaa keinotekoinen silmä, vaikkakin kehitys vie vielä useita vuosia.

Muutaman viime vuoden ajan keinotekoisien hermoverkkojen tutkimukseen on sijoitettu huomattavia rahasummia. Jotkut ovat pitäneet keinotekoisia hermoverkkoja USA:n vastauksena japanilaisten viidennen sukupolven tietokoneprojektille. USA:n sotilaallista tutkimusta rahoittava DARPA päätti viime syksynä käynnistää kaksivuotisen kartoitustiprojektin, johon se sijoittaa 33 miljoonaa dollaria. Alunperin sen suunnitelmassa oli lähes 400 miljoonan dollarin tutkimusprojekti, mutta alan tutkijat varoittelivat, että tällaisen rahamäärän kylväminen voisi olla tutkimukselle suuremmaksi haitaksi kuin hyödyksi.

Myös eurooppalaiset ovat olleet aktiivisia keinotekoisien hermoverkkojen tutkijoita. Saksan liittotasavalta käynnisti kymmenvuotisen projektin 67 miljoonaa dollaria vastaavalla summalla. ESPRIT:n puitteissa harrastetaan keinotekoisien hermoverkkojen tutkimusta noin 15 miljoonalla dollarilla vuodessa. Vaikka rahoitus on toistaiseksi pientä, sen vaikutukset moninkertaistuvat yksityisten rahoittajien ja te-

Tietokoneiden kuudes sukupolvi?



ollisuuden sijoitusten seurauksena. Suomella on oma Tekesin tukema neuraaliverkko-ohjelman-
sa laajan FINSOFT-hankkeen osana.

Alan johtavat asiantuntijat arvioivat, että keinotekoisia hermoverkkoja on muutaman viime vuoden kuluessa ryhtynyt tutkimaan noin neljätuhatta tutkijaa eri puolilla maailmaa. Luku on kasvanut niin nopeasti, että sen paikkansa pitävyyttä on vaikea arvioida. Syksyllä 1989 oikea luku lienee lähempänä kahdeksaa kuin kuutta tuhatta.

Neuraaliverkkoja on tutkittu jo pitkään

Keinotekoisia neuraaliverkkoja on tutkittu 1940-luvulta asti. Tietokoneen syntyäikoina ihmisen aivoja mallitettiin ”loogisten neuronien” avulla. Jos keinotekoiseen hermosoluun tuli riittävästi ärsytystä, se aktivoitui. Hermosolun mallina oli epälineaarinen kynnysselementti. Hermoverkon käyttäytymisen ratkaisivat hermosolujen kynnysfunktio ja hermosolujen välisten kytkentöjen eli synapsien voimakkuudet.

Jo 1940-luvun alussa Warren S. McCulloch ja Walter Pitts osoittivat, että yksinkertaisten loogisten hermosolujen verkko pystyi laskemaan kaikki laskettavissa olevat asiat. Myöhemmin tätä ajatusta kehitti John von Neumann suunnitellessaan digitaalista tietokonetta ja logiikan uusia muotoiluja.

Keinotekoisien hermoverkkojen ensimmäinen kukoistuskausi oli 1950-luvun loppupuolella. Tällöin Frank Rosenblatt rakensi Perceptronin nimellä tunnettuja loogisia hermoverkkoja, joiden tavoitteena oli kirjoitetun tekstin tunnistaminen. Viisikymmenluvulla alettiin kehittää myös oppivia hermoverkkoja ja biologisten hermoverkkojen teorioita.

Hermoverkkoja oli kuitenkin vaikea tutkia ja rakentaa. Sopivia komponentteja ei ollut saatavilla eikä tutkijoilla ollut tehokkaita tapoja tutustua malliensa ominaisuuksiin.

1960-luvun puolivälissä hermoverkkotutkijoiden innostus alkoi laantua. Yli vuosikymmenen kestänyt lamakausi alkoi Marvin Minskyn ja Seymour Papertin kirjoittettua teoksen *Perceptrons: an Introduction to Computational Geometry*. Siinä he todistivat useita loogisia hermoverkkoja koskevia lauseita, ja väittivät, että oli monia yksinkertaisia laskutoimituksia, joista tavallisimmat keinotekoiset hermoverkot eivät selviä. Minskyn ja Papertin kirja oli viimeinen naula neurotietokoneiden pakkauslaatikossa. Tutkimusrahoitus loppui, alaa koskevia artikkeleita oli vaikea saada julkaistuksi, ja neurotietokoneista tuli pahamaineisia. Vain muutammat tutkijat eri puolilla maailmaa jatkoivat neurotietokoneiden kehittelyä. Yksi tunnetuimmista ja sitkeimmistä alan harrastajista on suomalainen Teuvo Kohonen.

Neurotietokoneiden uusi aalto nousee

Nykyinen keinotekoisien hermoverkkojen myötätuuli alkoi tämän vuosikymmenen alussa. Omalta osaltaan siihen vaikuttivat tavonomaisten tietokoneiden tehon kasvu ja hinnan romahtaminen. Tutkijat saivat käyttöönsä laitteita, joilla monimutkaisiakin hermoverkkojen malleja pystyttiin kohtuullisin kustannuksin tutkimaan.

Malleja simuloitaessa havaittiin, että niillä oli monia kiinnostavia ominaisuuksia. Merkittävä tapahtuma oli tunnetun amerikkalaisen fyysikon John Hopfieldin USA:n tiedeakatemian julkaisuun vuonna 1982 kirjoittama artikkeli, joka ehdotti keinotekoisii neuraaliverkkoihin perustuvaa assosiativista muistia. Hopfieldin artikkeli sai paljon julkisuutta, osittain myös siksi, että sen käyttämä terminologia oli helposti ymmärrettävissä kaikille fysiikan perusteet tunteville lukijoille.

Ensimmäisen laajan kansainvälisen hermoverkkokonferenssin järjesti IEEE vuonna 1987. Viime

vuonna IEEE:n konferenssissa oli pari tuhatta osallistujaa. Kesäkuussa IEEE järjesti jälleen hermoverkkokonferenssin Washingtonissa, tällä kertaa yhdessä kansainvälisen hermoverkkotutkijoiden seuran kanssa. Osallistujia oli 2200.

Mitä hermoverkot sitten oikein ovat?

Keinotekoisien hermoverkkojen malleja on useita. Yhteistä eri malleille on, että ne rakentuvat yksinkertaisista prosessointielementeistä eli ”neuroneista”. Nämä keinotekoiset hermosolut on kytketty toisiinsa verkoiksi.

Maailmalla tutkitaan yleisimmin niin kutsuttuja Back-Propagation -verkkoja. Nämä verkot ovat yksinkertaisten 50-luvun Perceptronien yleistäyksiä. Perceptronissa oli tyypillisesti vain kaksi hermosolukerrosta, mutta lisäämällä kerrosten lukumäärää hermoverkko saadaan toteuttamaan monimutkaisiakin tehtäviä.

Engelmana monikerroksissa hermoverkoissa oli se, miten hermosolujen välisten kytkentöjen voimakkuuksia pitäisi säätää, jotta verkko saataisiin käyttäytymään halutulla tavalla. Useat tutkijat ovat viime vuosina kehittäneet oppimisalgoritmeja, joilla tällaisia monikerroksisia Perceptroneita pystytään opettamaan. Algoritmit perustuvat siihen, että verkolle annetaan malliesimerkkejä. Jos verkko reagoi väärin esimerkiksi, sen kytkentöjä muutetaan tunnetulla tavalla. Riittävän monen esimerkin jälkeen verkko tekee vain vähän virheitä. Monikerroksisten verkkojen sisällä oleviin keinotekoisii hermosoluihin ei ole suoraa yhteyttä, joten oppiminen tapahtuu propagoimalla solujen välisten kytkentöjen muutoksia verkon läpi. Virheiden vaikutukset lasketaan output-kerroksesta taaksepäin kohti input-kerrosta, mistä oppimismenetelmä on saanut nimensä ”back-propagation of error”.

Toinen yleisesti tutkittu hermoverkkomalli on Hopfieldin nimel-

lä kulkeva verkko. Se voidaan rakentaa kytkemällä esimerkiksi joukko operaatiovahvistimia verkoksi, jossa kunkin vahvistimen ulostulo on kytketty kaikkien vahvistimien sisäänputoluihin. Tällaisen täydellisesti kytketyn vahvistinverkon ominaisuudet mää- räytyvät vahvistinten välisten kyt- kentöjen voimakkuuksista. Vah- vistinten välisiin kytkentävästus- ten voimakkuuksiin voidaan tal- lentaa joukko verkon tasapainoti- loja. Kun verkon sisääntuloon kytketään jokin alkutila, se nope- asti hakeutuu läheiseen tasapai- notilaansa. Tällaista verkkoa voi- daan käyttää esimerkiksi sisältö- hakuun perustuvana muistina.

Myöhemmin Hopfield kehitteli verkkoaan yhdessä David Tankin kanssa. He osoittivat, että neu- raaliverkolla voidaan ratkaista esimerkiksi optimointiongelmia. Hopfield ja Tank ratkaisivat ver- kollaan mm. laskennallisesti vai- keaksi tunnetun kauppamatkus- tajan ongelman.

Maaailmalla tunnetaan erittäin hyvin myös Kohosen mukaan ni- metty itsejärjestyvä verkko. Sii- hen perustuen on rakennettu mm. puheentunnistustilaitteistoja ja ke- hitetty teorioita biologisten her- moverkkojen ominaisuuksista.

Neurotietokoneiden markkinat kasvavat

Neurotietokoneita on jo jonkin aikaa voinut ostaa kaupallisina tuotteina. Vuonna 1988 markki- nat olivat yhteensä noin 20 mil- joonaa dollaria. Alan liikevaihi- don arvioidaan kasvavan tänä vuonna noin 50 miljoonaan dol- lariin. Suurin osa kehitystyöstä tapahtuu kaupallisten markki- noiden ulkopuolella, joten tark- koja tietoja neuraalikoneisiin käytetyistä rahoista ei ole saatavilla.

Yleisimmin keinotekoisia her- moverkkoja simuloidaan tavalli- sissa työasemissa. Tutkimusta tehdään myös paljon transputer- pohjaisissa rinnakkaiskoneissa, tavallisissa supertietokoneissa ja esimerkiksi kymmeniätuhansia suorittimia sisältävissä hyperkuu-

tioissa. Parhaillaan kehitetään myös VLSI-hermoverkkoja ja op- tiikkaan perustuvia neurotietoko- neita. Intel julkaisi kesällä mikro- piirin hermoverkkojen rakenta- miseen ja erikoisprosessoreja myyvät ja kehittävät sen lisäksi ai- nakin Motorola, Texas Instru- ments, Fujitsu, NEC, Hitachi, Mitsubishi ja Toshiba. Eräät la- boratoriot ja tutkijat kehittävät myös biomolekyyliin perustu- via verkkokoneita.

Kaupallisia sovelluksia on tois- taiseksi käytössä vain vähän. So- tilassovelluksina hermoverkkoja yritetään käyttää etenkin erilaisissa hahmontunnistustehtävissä. TRW on esitellyt Mark IV neuro- tietokoneellaan tehdyn sovelluk-

Liikevaihdon arvioidaan kasvavan tänä vuonna noin 50 miljoonaan dollariin.

sen, joka luokittelee lentokoneita niiden liikkumistavan mukaan hyökkääviin ja rauhanomaisiin. Neurotietokoneita käytetään tiet- tävästi ainakin tutkasignaalien analysointiin ja sukellusveneiden äänien tunnistamiseen. Amerik- kalainen Behavioristics Inc. on myynyt rline Marketing Tactician -järjestelmäänsä lentoyhtiöille, jotka käyttävät sitä lentolippujen myynnin optimointiin.

Parhaillaan on koekäytössä neuraaliverkkoihin perustuvia järjestelmiä, joiden tehtävänä on valvoa lentoturvallisuutta. Nämä järjestelmät on opetettu haista- maan muoviräjähteitä. Neuraali- koneita käytetään menestykselli- sesti myös japaninkielisten kanji- merkkien tunnistukseen ja käsin- kirjoitetun tekstin lukemiseen. Niitä yritetään käyttää myös sää- tötehtäviin robotiikassa ja tilas- tollisten riippuvuuksien oppimi- seen pörssi- ja valuuttakursseista. Siemens käyttää neuraaliverkkoja autonlämmittimien laadunval- vontaan ja AT&T valvoo piirilevy- jen asennusvirheitä videokame- raan kytketyllä neuraaliverkol- laan. Ford on kertonut käyttä- neensä neurotietokoneita auto-



jensa maalauksen valvontaan. Neuraaliverkkoja käytetään myös esimerkiksi luottojen arvioinnis- sa. Ainakin American Express on ilmoittanut käyttävänsä keinote- koisia hermoverkkoja, vaikka- kaan se ei suostu kertomaan tark- kaa sovellusalaan.

Tietokoneiden kuudes sukupolvi?

Nykyiset tietokoneet ovat histori- allisen kehityksen tulosta, mutta kehityksen päälinjat ovat 1940- luvulta. Digitaaliset tietokoneet pystyvät laskemaan yksinkertai- sia laskutoimituksia ja differenti- aaliyhtälöitä. Ne pystyvät käsitte- lemään merkkijonoja.

On kuitenkin mahdollista, että monet tekoälytutkimuksen esiin nostamat ongelmat eivät ratkea digitaalisella tietokoneella. Näis- tä tietokoneiden teoreettisista ra- joista kertoo WSOY:n äskettäin julkaisema kirja Ajatuksia syn- nittävät koneet, joten tyydyn vain kertaamaan muutamia usein esi- tettyjä kuudennen tietokonesu- kupolven tunnusmerkkejä.

Kuudennen sukupolven tieto- koneet ovat adaptiivisia infor- maationkäsittelylaitteita. Niissä on erittäin paljon prosessoreita. Kuudennen sukupolven tietoko- neiden ohjelmointi muistuttaa ti- lastollista fysiikkaa. Laitteiden komponentit toimivat optiikan ja biokemian perustalla. Ensimmäi- sessä vaiheessa, vuonna 1992, kuudennen sukupolven tietoko- neet toimivat tavallisten digitaalisten tietokoneiden ja ihmisen ra- japintoina, ennenkaikkea säätö- järjestelminä ja tilastollisen dan- tan analysoinnissa.

Vuonna 1992 ?! Aivan. Tekno- logian eturintama kulkee aina vuosia volyyminmarkkinoiden edellä. Vaikka hermoverkkotutki- joilla on vielä erittäin merkittäviä ongelmia ratkottavanaan, tekno- logian rakennusosat ovat jo ole- massa. Ehkä niiden yhdistelmänä syntyy tietokoneiden kuudes su- kupolvi. Tai vasta toinen. ■

Tiimin SNA liittää

Työryhmäverkot ja keskuskoneet yhteen

SNA-tuoteperhe toimii yhtä hyvin sekä Nokia MS LAN Manager että MS-DOS-verkoissa. Työryhmäverkko voidaan liittää suoraan 3174/3274-ohjaimeen, esimerkiksi Alfaskop 74:ään. Keskuskoneliitäntä on mahdollista toteuttaa myös etäislinjoilla modeemien kautta.

Työasemaohjelmisto toteuttaa 3278/3279-määrittelyjen mukaiset toiminnot. Ohjelmistoon voidaan sisällyttää myös useita lisäohjelmia pitämään huolta grafiikasta ja erilaisista sovellusliittymistä.

Päätteen ja kirjoittimen emulointi

Pääte-emulointia varten työasemiin tarvitaan 3270 SNA LAN Workstation -ohjelma varsinaiseen 3270-emulointiin. Ohjelmasta on myös entry level-versio sekä versio yksittäismikroille. Järjestelmän parhaita puolia ovat monipuolinen grafiikka ja pieni muistin tarve.

Tuoteperhe noudattaa 3278/79-määrittelymallien 2 (24x80), 3 (32x80), 4 (43x80) ja 5 (27x132) mukaisia teollisuusstandardeja.

MS-DOS-istunnon lisäksi työasemassa voi olla enimmillään jopa neljä muuta istuntoa, joiden vaihto ja aktivointi on toteutettu niin kutsutulla hot key -näppäimellä. Luonnollisesti myös tiedostojen siirto työaseman ja keskuskoneen välillä on mahdollista.

Grafiikka on toteutettu vektorigrafiikalla, joka toteuttaa 3192 G:n mukaiset toiminnot. PS Grafiikka noudattaa puolestaan 3279 S3G PS -määrittelyä.

Työaseman ja keskuskoneen välistä yhteistyötä voi terästä käyttämällä SRPI- ja LLAPI-ohjelmointivälineitä.

3287-kirjoittimen (LU 1 ja LU 3) emulointi voidaan toteuttaa joko työasemassa tai tietoliikennepalvelimessa. Tulostus oman työaseman kirjoittimelle säästää keskuskoneen kapasiteettia.

Työryhmäverkkojen tultua on monessa organisaatiossa kysytty, voiko verkon SNA-yhteydet hoitaa keskiteysti? Yksi vastaus on Tiimin SNA-tuoteperhe MS-DOS-palvelimelle, joka soveltuu NetBIOS-yhteensopiviin verkkoihin kuten Tiimi Token Ringiin ja Ethernetiin. Lisäksi SNA-yhteyksiin voidaan liittää myös yksittäisiä MikroMikkoja.

SDLC kauas ja koaksiaali lähelle

Tietoliikennepalvelin vastaa kaikesta keskuskoneen suuntaan tapahtuvasta liikenteestä eli luo, ylläpitää ja lopuksi katkaisee yhteyden. Tiimissä voi olla useita tietoliikennepalvelimia, jollaisen voi rakentaa esimerkiksi MikroMikosta.

Tiimin SNA-tuoteperheessä työasemaohjelmistoja käytetään samalla tavalla riippumatta siitä, millainen tietoliikennepalvelin verkossa on. Vain istuntojen lukumäärä vaihtelee. Istuntokapasiteettiinhan lasketaan mukaan sekä näyttöjen että kirjoittimien lukumäärä.

Myös tuotteiden asennuksessa on jonkin verran eroja. Linjana voi olla joko etäis-SDLC tai lähilinja koaksiaalilla.

SDLC-linjalle kytkemiseksi käytetään 3270 SNA/SDLC Gateway Software -pakettia SNA/SDLC- tai SNA/DLC-liitännällä. Koaksiaali-kytkentää varten tietoliikennepalvelinohjelmistoksi on tarjolla 3270 SNA/COAX Gateway Software -paketti. Ohjelmistot toteuttavat PU 2.0-määrittelyt. Jopa 128 LU:ta voidaan käyttää (LU 1, 2 tai 3).

Tiimi voidaan kytkeä myös suoraan ohjaimeen, esimerkiksi S74:ään. Koaksiaali-kytkennässä istuntojen määrä on kuitenkin rajoitettu viiteen. Yhteyskäytäntönä on tällöin DFT.

Nokialainen verkotusmalli on pyramidi. Pyramidin alimpina rakennuspalikoina ovat perusjärjestelmän tuotteet kuten mikrot, verkkokortit ja verkkokäyttöjärjestelmät. Niiden päälle rakennetaan tietoliikenne ja yleissovellukset. Pyramidin huipulla sijaitsevat toimialakohtaiset ohjelmistot, jotka voivat olla joko yleisohjelmia tai erikoissovelluksia, jotka vaativat usein myös tietoliikennepalveluita.

Jos keinot eivät riitä otetaan konstit avuksi

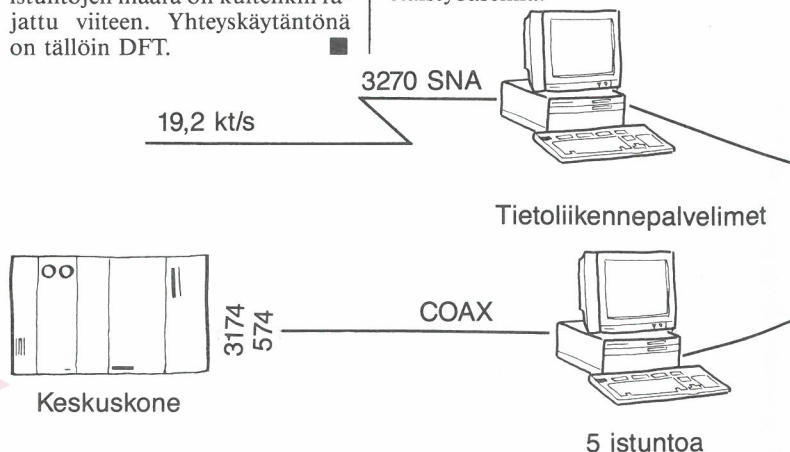
Keskuskoneyhteytenä voi olla lähi- tai etäislinja, joista suurimman osaan löytyy jo sopiva Tiimin tuoteperhe. Kaikki standardit noudattavat laitteet voivat kommunikoida Tiimin kanssa.

Tuoteperheistä löytyy SNA-yhteyksiin jopa useita vaihtoehtoja: 3270, 5250, SDLC, DLC ja Coax. Palvelimet voivat olla joko MS-DOS- tai OS/2-maailmasta. Samoin X.25-yhteyksille tai muuhun avoimeen liikenteeseen on tarjolla vaihtoehtoja painotusten mukaan. Tavallisia tahdistamattomia yhteyksiä voidaan toteuttaa esimerkiksi AACS-tuoteperheellä (Softari 1/89).

Elleivät perusmallit riitä, organisaation tarpeisiin voidaan rakentaa ratkaisut omana tuotekehitysohjelmistona.

Tiedon kuljettava myös muihin verkkoihin

Tietoliikennettä tarvitaan myös verkkojen väliseen liikenteeseen. Tiimit voidaan kytkeä paikallisesti tai etäislinjan kautta toisiinsa. Tuotteilla on mahdollista liittää myös erilaisia verkkoja (Token Ring, Ethernet) toisiinsa. Yhteyksien toteuttamiseksi Nokialla on eri vaihtoehtoja kuten toistimet, sillat, NetBIOS-sillat ja reitittimet. Tiimiin voidaan liittää myös etäistyoasemia.



Tiimin SNA-tuoteperhe MS-DOS-tietoliikennepalvelimelle on ratkaisu SNA-tietoliikenteen ongelmiin.

Perusjärjestelmän, yleissovellusten ja tietoliikenteen päälle rakennetaan toimialakohtainen ratkaisu.



Avoimuus on liitettävyyden ohella päivän sana, ja verkkoja, keskuskoneita ja päätettä yhdistetään toisiinsa kaikkialla. Tällaisessa tilanteessa järjestelmän tietoliikenne-ratkaisut nousevat ratkaisevaan asemaan, sillä niiden on selvittävä paitsi nykyhetken myös tulevaisuuden kommunikointitarpeista.

Tuotteita ja ratkaisuja on sekä NetBIOS- että MAC-tasoisia. NetBIOS-tasoisista silloista esimerkkinä voi mainita paikalliseen liittämiseen tarkoitetun NetBIOS Interconnect Programin, jonka etuna on se, että verkkotyyppi ja yhteyskäytäntö voivat olla sillan molemmiin puoliin erilaisia. Etäisiittäminen onnistuu Remote NetBIOSin avulla.

Mikro, työasema ja pääte yksissä kuorissa

Verkon käyttäjille tietoliikenne mahdollistaa yhteydet työryhmäverkon ulkopuolelle. Työryhmäverkoissa samaan työasemaan saadaan yksittäismikron, verkkotyöaseman ja päätteen toiminnot. Työaseman käyttö on helpompaa kun ympäristö on aina tuttu. Henkilökohtaiset tietokoneet tarjoavat suoritustehoa ja runsaasti hyviä henkilökohtaisen työn tuottavuutta parantavia välineitä kuten esimerkiksi hiirtä käyttävät perussovellukset.

Yhdenmukaisen käyttäjäliittymän kautta näyttää siltä kuin so-

Lyhenneviidakosta toimiviksi yhteyksiksi

Tiimi hajauttaa ja yhdistää

vellus ja tietoa olisivat osa käyttäjän henkilökohtaista tietojärjestelmää. Perinteiseen pääte-käyttöön verrattuna ratkaisu on tehokkaampi, koska lisänä ovat mikron mukanaan tuomat ominaisuudet kuten grafiikka ja helpokäyttöiset ohjelmat. Työasemien monipuolisuuden ansiosta käyttäjälle voidaan toteuttaa ympäristö, jossa hänen ei tarvitse tietää, missä kulloinkin sovellus ja tietoa sijaitsevat.

Venyy venyy vaan ei pauku

Työryhmässä sovellukset on hajautettu siten, että niiden suoritus tapahtuu siellä, missä se on mielekkäintä ja tehokkainta. Järjestelmän aarkkitehtuurin kannalta se merkitsee sitä, että osa sovelluksista toimii käyttäjän omassa työasemassa ja osa työryhmän palvelimissa tai keskuskoneissa.

Monipuolisten tietoliikenneyhteyksien etuihin kuuluu myös keskuskoneen kuormituksen laskeminen. Kun suuri osa tietojenkäsittelyä voidaan tehdä työasemissa, säästetään keskuskoneen kapasiteettia. Koska massamuitit ja laskentateho ovat mikrotekniikalla olennaisesti edullisempia, tulokset näkyvät myös kustannuksissa. Ilman tehokkaita tietoliikennetarkoituksia tällainen hajauttaminen ei olisi järkevää.

Näin saavutetaan joustava työskentely-ympäristö, joka mahdollistaa myös usean käyttäjän sovellusten toteuttamisen. Myös

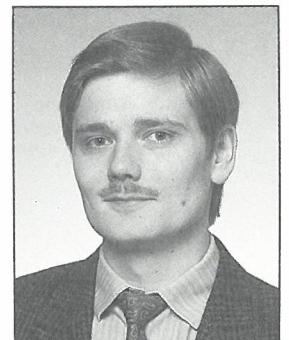
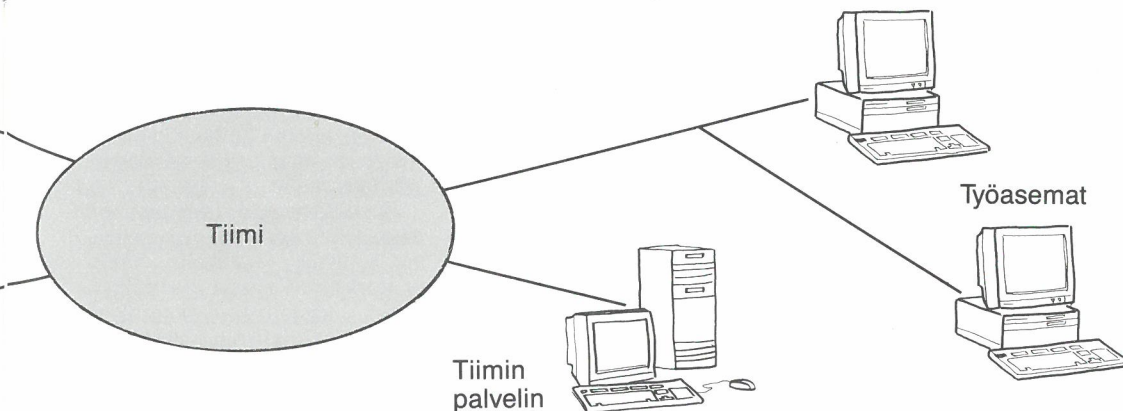
työasemien käyttäjäystävällisiä graafisia käyttöliittymiä voidaan käyttää hyväksi, vaikka itse sovellus sijaitsee palvelimissa. Palvelinhan voi olla esimerkiksi keskuskone, jonka kanssa työasema vaihtaa tietoja SQL:n tai APPC:n avulla.

MS-DOS on tosiasia, joka on huomioitava

Järjestelmien hajauttamistarve on ollut erityisesti viime vuosina melkoinen. Nyt ollaan siirtymässä toisen sukupolven järjestelmiin, joissa sovelluksia ei enää tehdä ainoastaan yhdelle koneelle, vaan sovellus tai sen osia hajautetaan keskenään yhteydessä oleviin laitteisiin. Näin syntyy luontevasti henkilökohtaisista tietokoneista rakennettavia työryhmiä, jotka tietoliikenteen avulla yhdistetään keskenään tai ulkopuolisiin resursseihin.

Nykyisin on paljon käytössä MS-DOS-perusteisia ratkaisuja. Eräs vaihtoehto on rakentaa verkot Nokia MS LAN Managerin avulla.

Nokian tietoliikennetarkoituksia on suunniteltu siten, että kasvu MS-DOS-pohjaisista ratkaisuista OS/2-maailmaan onnistuu ilman ongelmia. Esimerkiksi uusimmissa tietoliikenneohjelmistoissa käytetään MS-DOS-työasemia ja MS OS/2-palvelimia. MS-DOSia tullaan käyttämään runsaan ohjelmistokantansa vuoksi työasemissa vielä pitkään. ■



Olli Nurminen.

Tandem ja Tiimi yhteen

Kolme arkkitehtuuria, kolme tietokantaa

Niin Tiimi kuin Tandemkin tuntevat Netbios-rajapinnan eli yhteyden rakentaminen näiden kahden järjestelmän välille on ollut vain ajan kysymys. Nyt tuo yhteys on valmis, ja se avaa niin Tandemille kuin Tiimillekin aivan uusia uria.

Tandemin liittäminen työasemaverkkoon mahdollistaa useita sovellusarkkitehtuuri- ja tiedonhallintavaihtoehtoja. Niin tieto kuin sovelluksetkin voidaan sijoittaa sekä järjestelmän että käyttäjän kannalta edullisimmalle tasolle.

Työryhmäverkko ja sovellusarkkitehtuuri

Tandemin Multilan-järjestelmä liittää Netbios-rajapinnan omaavan työasemaverkon Tandemiin. Se tarjoaa kaksi käyttötapaa: 6530-pääte-emulointi ja ohjelmasta-ohjelmaan keskustelu. Verkko voi olla esimerkiksi Ethernet tai Token Ring ja työasema voi olla vaikka MikroMikko 3 tai 4. Ohjelmasta ohjelmaan -keskustelu on sovelluskohtaista. Multilan-järjestelmään kuuluu myös tiedostopalvelin, jolloin työasemassa ajettava sovellus käyttää Tandemin levyille sijoitettuja MS-DOS-tiedostoja läpinäkyvästi.

Periaatteessa sovellusarkkitehtuureja on kolmen tyyppisiä: tyhmiä päätteitä tukevat sovellukset, joiden käyttöliittymä- ja tiedonhallinta-osat ovat keskuskoneessa, yhteistoimintasovellukset, joiden käyttäjää lähellä oleva osa on työasemassa ja tietokantaa lähellä oleva osa on keskuskoneessa sekä lopuksi työasemassa kokonaan olevat sovellukset, jotka käyttävät jossain verkossa olevaa tietokantapalvelintä. Kullakin arkkitehtuurilla on omat käyttöalueensa ja vahvat puolensa. Tiimi-työryhmäverkon ja Tandemin yhteisellä on mahdollista toteuttaa jokainen niistä.

Työasema leikkii Tandemin päätteitä

Perinteisesti sovellukset on tehty tavallisille päätteille, Tandemin kyseessä ollessa Tandemin 6530-päätetyypin kanssa yhteensopiville päätteille. Tiimi-työryhmäverkkoa käytettäessä työasemat voivat emuloida 6530-päätettä, jolloin kaikki sille tehty sovellukset ovat käytettävissä.



Käyttäjä voi vaihtaa omista henkilökohtaisista sovelluksistaan kuten tekstinkäsittelystä suoraan Tandemilla toimivaan sovellukseen, esimerkiksi sähköpostiin. Emulaattoriin kuuluu myös tiedostonsiirto-ohjelmisto, joten käyttäjä voi liittää teksturilla tekemänsä kirjeen sähköpostista lähtevään viestiin tai tulostaa vastaanottamansa tiedoston työasemaan liitettyllä laser-kirjoittimella.

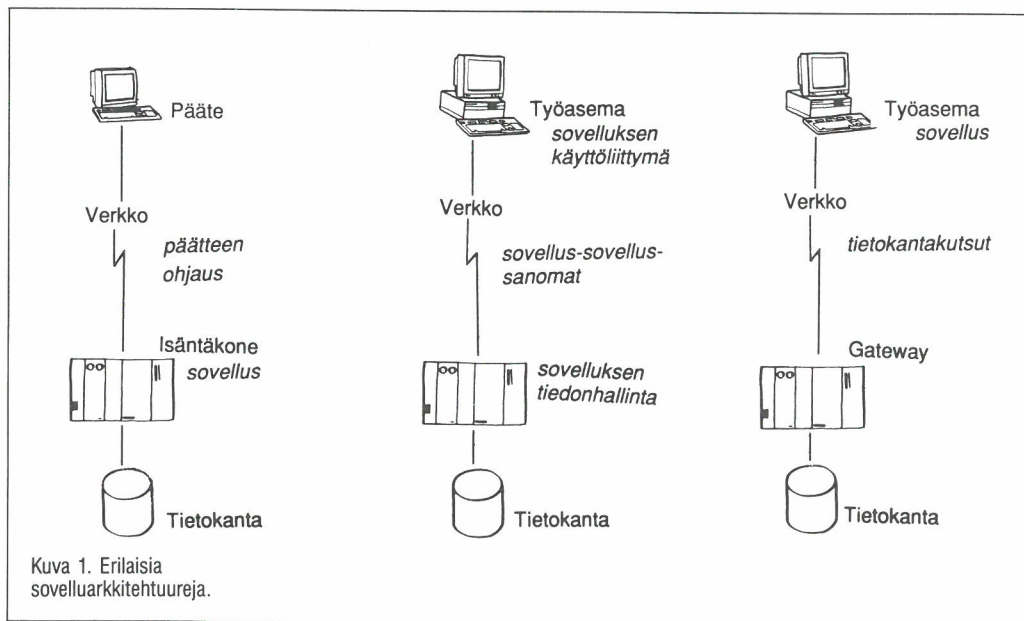
Tandem ja Information Builder Inc ovat perustaneet yhteisen projektin nimeltä Focus sovelluskehittimen siirtämiseksi Tandemille. Focus käyttää 6530-päätteitä tai niitä emuloivia työasemia. Kehitettävät sovellusohjelmat pyörivät Tandemissa. Focus on saatavilla Tandemille vuoden vaihteen jälkeen.

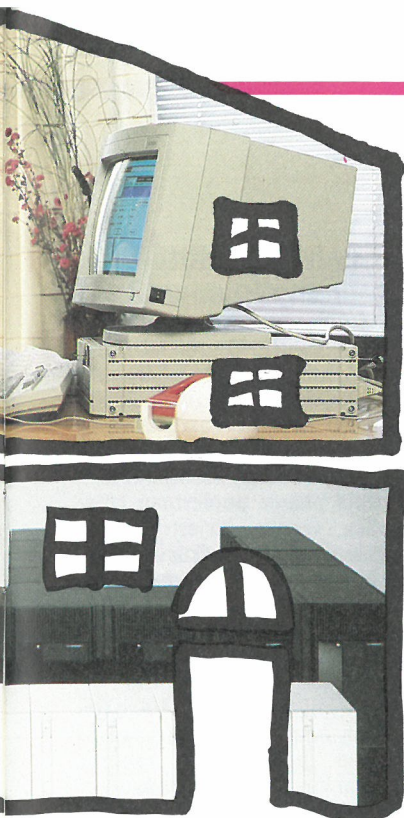
Liittymä päätteessä sovellus Tandemissa

Kehittyneen käyttöliittymän tarve on yhä suurempi myös tilanteissa, joissa varsinainen sovellus toimii edelleen Tandemissa. Käyttöliittymä voi olla toimintaperiaatteeltaan joko perinteinen merkkipohjainen tai moderni graafinen järjestelmä.

Tandemin IDS (Intelligent Device Support) tukee ratkaisuja, joissa työasema hoitaa käyttöliittymän ja muu osa sovelluksesta jää Tandemille.

Käyttöliittymän toteutus voidaan tehdä sovellukseen parhaiten sopivalla tekniikalla. Useissa merkkipohjaisissa käyttöliittymissä käytetty työkalu on DScreen. Graafiseen käyttöliittymään voidaan vuorostaan käyttää SQLWindows-järjestelmää. Nokia Datalla on kehitetty yleis-





käyttöiset C-kieliset rutiinit, joiden avulla SQLWindows-sovellus voi kommunikoida Netbios-verkon yli Tandemin IDS-sovelluksen kanssa. Rutiinit ovat vapaasti kaikkien kiinnostuneiden saatavilla.

Työaseman monipuoliset välineet ja tehokkuus antavat mahdollisuudet korkeatasoisen käyttöliittymän tekemiseen. Tandemille valmiiksi käännetty sovellukset antavat puolestaan lyhyet vasteajat ja suuren tapahtumien käsittelykapasiteetin, joten massiiviset tuotantosovellukset sopivat parhaiten tällä arkkitehtuurilla toteutettaviksi.

Sovellus työasemassa Tandem palvelimena

On myös tilanteita, joissa sovellus pyörii kokonaan työasemassa ja ainoastaan tietokanta on ulkopuolisessa tietokantapalvelimessa. Työasemassa käytettävä sovelluskehitin mahdollistaa nopean sovellusten kehityksen ja muokkaamisen, joka on tarpeen varsinkin päätöksenteon tukena käytettävissä järjestelmissä.

Nokia Datan toimesta Tandemille ollaan toteuttamassa SQL-Base-tyyppistä tietokantapalvelintä. Sovellus pyörii kokonaan työasemassa esimerkiksi SQLWindowsilla tai C-kielillä ja DScreenillä toteutettuna ja käyttää tietokantaan kohdistuvia SQL-funktioita. Tandemissa oleva SQLBase Gateway vastaanottaa syntyneet tietokantakomennot, kääntää ja suorittaa ne sekä palauttaa vastaukset takaisin työasemassa olevalle sovellukselle. Työasemasovellus ei periaatteessa edes tiedä, että tietokanta on Tandemilla.

Samanlaisen Gateway-ratkaisuun perustuvat myös Tandemille työn alla olevat Ingres- ja Oracle-toteutukset. Niiden työvälineet ovat käytettävissä työasemissa ja tietokanta perustuu Tandemin NonStop SQL-tiedonhallintajärjestelmän käyttöön. Nämä työvälineet ovat ennakkotietojen mukaan saatavilla Tandemille ensi keväänä.

Tietokanta on kolmessa tasossa

Työryhmäverkkojen ja Tandemin käyttö antaa mahdollisuudet toteuttaa todellisia on-line enterprise ratkaisuja.

Henkilökohtaisessa käytössä olevat tiedot kuten oma myynnin toteutuma ovat omassa työasemassa oman tiedonhallinnan hoidossa, jolloin vastuu niiden oikeellisuudesta ja varmistuksista on yksin käyttäjällä.

Työryhmän tiedot ovat Tiimi-työryhmäverkon ja siinä olevien sovellusten hoidossa. Tietokanta on työryhmän tietokantapalvelimen hoidossa ja ryhmä vastaa tietojensa paikkansapitävyydestä sekä varmistamisesta.

Yritystason tietokannat ovat Tandemin NonStop SQL tiedonhallinnan hoidossa ja yrityksen atk-osaston omistuksessa. Sieltä jokainen saa ajan tasalla olevan kuvan yrityksen liiketoiminnan nykyyhteisestä, mutta ylläpitovastuu on yritystason sovelluksilla, jotka pyörivät Tandemilla.

Käyttöliittymä kaikkiin sovelluksiin

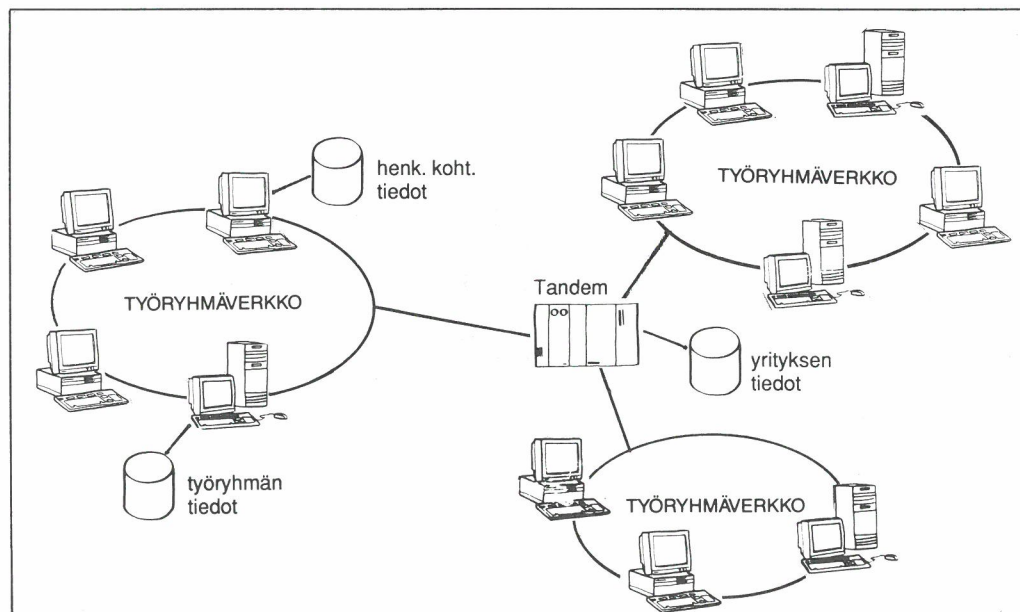
Edellä kuvattuja tekniikoita käyttäen on mahdollista luoda käyttäjälle Windows-ympäristö, joka kattaa kaikki käyttötavat.

Käyttäjä voi ajaa Windowsin alla omia henkilökohtaisia sovelluksiaan, kuten Excel-taulukkolaskentaa tai käynnistää Tandemin pääte-emulaattorin ja ajaa kokonaan Tandemilla olevia sovelluksia. Vastaavasti hän voi käynnistää sovelluksen, jonka käyttöliittymä on toteutettu SQLWindowsilla ja jonka tietokantaa lähellä oleva osa on toteutettu Tandemissa tai joka on kokonaan toteutettu SQLWindowsilla käyttäen ja jossa tietokannasta huolehtii Tandemin NonStop SQL-tiedonhallinta.

Tiimi-työryhmäverkon ja Tandemin yhteenliittäminen merkitsee valinnan vapautta. Tietokannat on mahdollista sijoittaa sinne, missä on vastuu niiden sisältämien tietojen omistuksesta. Tietokantoja käsittelevät sovellukset on mahdollista tehdä käyttötarkoitukseen soveltuvin välinein. Nopeaan tiedontarpeeseen on tarjolla lukuisa joukko sovelluskehittäjiä, kun taas massiivisiin tuotantosovelluksiin on tarjolla Tandemin perinteiset sovelluskehitysmekanismit.

Tandem ja tiimi-työryhmäverkot on sovelluskehityksen ja tuotannon ympäristö, joka takaa käyttäjilleen kilpailukykyisen tietohuollon.

Kuva 2. Kolmen tason tietokanta.



Liikettä linjoilla

Tiimistä S2500:n uusi pääteympäristö

Tiimi-verkon työasemissa toimiva LANCON-WS-ohjelmisto, ja yksittäisissä mikroissa pyörivä EPCON emuloivat S2500:n älykkäitä synkronisia työasemia. Mitään toiminnallista eroa työasemien ja mikrojen välillä ei ole.

S2500 toimii verkon tapahtumakäsittelyn sovelluspalvelimena eli S2500:n tapahtumankäsittelyyn kehitettyjen sovellusten yhteiset tiedot voidaan säilyttää itse S2500:ssa. Kukin käyttäjä voi halutessaan poimia tarvitsemansa tiedot omaan mikroonsa tai palvelimeensa ja käsitellä tietoja omissa sovelluksissaan.

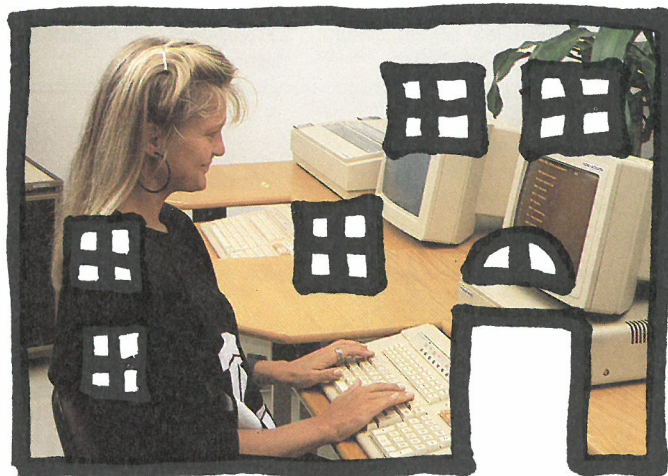
Yksityistä ja julkista tietoa

Henkilökohtaiset tiedot on mahdollista tallentaa mikron loogiselle levyasemalle (virtuaalilevy), joka on todellisuudessa S2500:n levyllä sijaitseva relatiivitiedosto. S2500:aan sijoitettu virtuaalilevy tarjoaa hyvän keinon suojata omat tiedostot asiattomilta käyttäjiltä. Samalla omien tietojen varmistus hoituu kätevästi S2500:n tietokannan päivittämisen varmistuksen yhteydessä.

S2500:n kiintolevy on fyysisestikin oikean kokoinen. Esimerkiksi mikrolle näkyvä 22237 kilotavun levy vie S2500:ssa 88070 sektoria eli käytännössä 22550 kilotavua.

Virtuaalilevy voi olla henkilökohtainen tai useat käyttäjät voivat käyttää samaa levyasemaa eli viitata samaan S2500:n tiedostoon. S2500 toimii siis eräänlaisena tiedostopalvelimena. Päivitystoiminnot hoidetaan DOS-ympäristön mukaisesti yksi käyttäjä kerrallaan. Virtuaalilevy voi olla myös pelkästään luettavissa ilman päivitysoikeuksia, jolloin yhteiset lukuarkistojen ja esimerkiksi PC-ohjelmistojen päivitykset hoidetaan S2500:n kautta.

Virtuaalilevyn vaatimat ohjelmat (EM24HD.SYS) ja S2500:n ohjelmat (PC-DEVICE, EM24-DEV ja EM24-FORMAT) toimitetaan emulointiohjelman mukana levykkeellä.



Nokia Data on kehittänyt Tiimi työryhmä-verkon ja S2500-järjestelmän välille LANCON-nimisen yhdyskäytävän. Samalla mikron emulaattorista EPCONista tulee markkinoille uusi versio 4.0.

Palveluja riittää molempiin suuntiin

S2500:n kautta Tiimin käyttäjät saavat yhteyden myös ulkopuolisiin järjestelmiin. IBM 2780/3780:aan tieto liikkuu BSC:n ja 3770:aan SNA:n kautta eräsiirtoin. Tiimin välityksellä S2500:n käyttäjä pääsee puolestaan käyttämään verkon järjestelmiä ja verkossa olevia yhdyskäytäviä kuten X.400:aa ja asynkronisia linjayhteyksiä.

LANCON mahdollistaa työryhmäverkon ja S2500-järjestelmän resurssien jaon. Verkon työasemat voivat käyttää S2500:n tiedostoja omina levyaseminaan ja tulostaa haluamansa raportit S2500:n kirjoittimille. Vastaavasti S2500-järjestelmän omat älykkäät synkroniset työasemat voivat käyttää verkon kirjoittimia ja siirtää tiedostoja työasemien ja palvelimien levyille.

S2500:n ja mikron (tai palvelimen) välillä on mahdollista suorittaa normaali tiedoston siirto (PCCOPY). Siirron voi suorittaa miltä tahansa verkon työasemalta, yksittäiseltä mikrolta tai tavalliselta S2500:n työasemalta mille tahansa mikrolle tai kirjoittimelle.

S2500:n uusi käyttöjärjestelmä (EOS S5.2) tarjoaa valmiin komennon (PC), jolla siirrytään mikron DOS-kehotteeseen. Paluu S2500-järjestelmään tapahtuu EXIT-komennolla.

Emulointiohjelmat voidaan käynnistää myös komentojonolla, jolloin komentojonon käsky siirtyvät suoraan S2500:aan.

Query poimii, ja EPPS siirtää

S2500:n vuoropuhelumuotoista relaatiomallista kyselykieltä Queryä käyttämällä käyttäjä voi poimia haluamansa tiedot useasta eri tiedostosta samanaikaisesti ja siirtää ne joko näytölle, kirjoittimelle tai EPPS-ohjelmiston avulla mikron tiedostoon. EPPS osaa myös tarvittaessa muuntaa tiedon esimerkiksi SYLK-, BASIC-, CSV-tai DOS-muotoon.

Poiminta tapahtuu valitsemalla ennalta määritellyt tietokannan osa ja antamalla poimintakriteerit. Mikäli määrittely on uusi, valitaan haluttu tiedosto ja merkitään siirrettäväksi haluttavat tiedot (kentät). Mukaan voi liittää yhteen-, vähennys-, jako- ja kertolaskuja sekä yhtä- ja erisuuruus- ehtoja loogisilla operaattoreilla yhdisteltyinä.

Saman tasoisia ja alempia tiedostoja voi yhdistää viittaamalla yhteisillä tiedoilla tietokannan eri osiin. EPPS:n avulla S2500:n sovellusohjelmien tilastot voidaan tulostaa graafisesti pylväinä, viivoina tai ympyröinä. S2500 hakee tiedot ja tallentaa ne valmiina kuvina omaan arkistoonsa.

S2500:n sovellus voi ohjata mikroyä

S2500:n COBOL-kielinen sovellusohjelma voi kommunikoida mikron sovellusten kanssa. Sekä LANCON että EPCON sisältävät ohjelmoinnille tarkoitettua rajapinnan (API). Lisäksi S2500:n sovellusohjelma voi käynnistää mikrojen ohjelmia, luoda tiedostoja mikrojen levyille, päivittää ja lukea mikrojen tiedostoja, tarkistaa mikrojen tiedostojen koot ja päiväykset, kommunikoida mikrojen sarjaporttien välityksellä, soittaa musiikkia mikrojen kaiuttimista, ... Käytännössä kaikkia mikron osia ja toimintoja voidaan ohjata ja käsitellä suoraan S2500:n sovellusohjelmasta käsin.

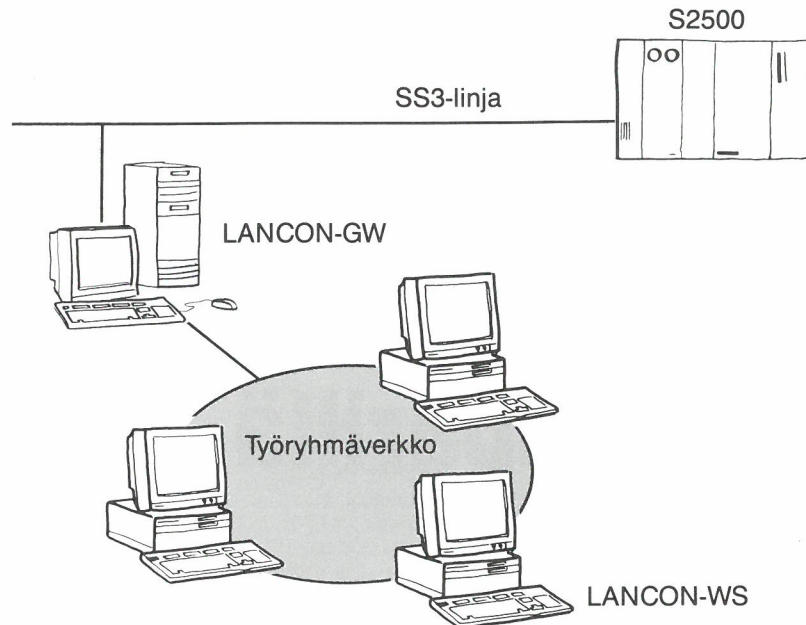
Yhdessä tiedon siirron ja poiminnan kanssa S2500:n sovellusohjelma voi esimerkiksi taustajona siirtää osan tietokannasta haluttuna ajankohtana Tiimi-verkon palvelimeen tai yksittäiseen mikroon. Mikroon ei tarvitse olla siirtoa varten kirjoitettua S2500:aan sisään.

Kaikki S2500:n tulosteet voidaan ohjata suoraan mikron levyille haluttuun tiedostoon.

Standardiliitännät eri tarpeita varten

LANCON koostuu kahdesta osasta: yhdyskäytävänä toimivasta LANCON-GW:stä ja työasemissa käytettävästä LANCON-WS:stä. LANCON-GW on Tiimi työryhmäverkkoon liitetty mikro (MM3/MM4), joka tarjoaa samanaikaisen yhteyden S2500:aan usealle eri käyttäjälle.

LANCON perustuu NetBIOS-tason kommunikointiin, ja se on tarkoitettu ensisijaisesti käytettäväksi Nokia MS LanManager-pohjaisissa Ethernet-verkoissa, joissa on yhdyskäytävän lisäksi OS/2-palvelin. NetBIOS-liitäntä mahdollistaa sekä Ethernet- että Token Ring-verkkojen liittämisen S2500:aan. LANCONin ensimmäinen vaihe kattaa Tiimi Ethernet-verkot, ja seuraava vaihe Tiimi TokenRing-verkot.



Yhdyskäytävässä käytettävää LANCON-GW-ohjelmistoa suositellaan ajettavan dedikoidusti. LANCON-GW käyttää MS-DOS-käyttöjärjestelmää. Ohjelmiston lisäksi yhdyskäytävä tarvitsee sovitimen sekä S2500-järjestelmän SS3-verkkoon että Tiimi työryhmäverkkoon. Tiimistä voi olla useita yhdyskäytäviä samaan tai useaan eri S2500:aan.

LANCON-GW:stä on kaksi eri versiota. Toinen viidelle ja toinen kuudelletoista käyttäjälle. LANCON-GW sisältää myös tarvittavat ohjaustiedot käyttäjien tunnuksista, samanaikaisten käyttäjien määrästä jne. Tärkeimmille käyttäjille voidaan määritellä kiinteät osoitteet, jolloin he saavat aina yhteyden S2500:aan muiden käyttäjien jakaessa yhteistä osoitejonoa.

Verkon DOS-työasemissa (väh. DOS 3.3) ajetaan LANCON-WS-ohjelmistoa, joka kommunoi LANCON-GW:n kanssa NetBIOSin välityksellä.

Tiimi työryhmäverkon voi liittää S2500:aan myös sekä kiinteän linjan että X.21- tai X.25-verkon

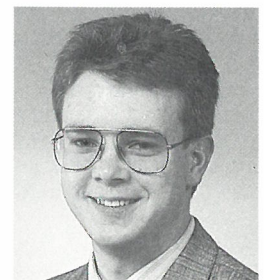
välityksellä S2500:n oman etäisohjaimen hoitaessa linjaliikennettä.

Yksittäisissä mikroissa käytetään EPCON-emulointiohjelmistoa ja SS3-verkon sovitinta.

S2500 ja Tiimi loppukäyttäjällä

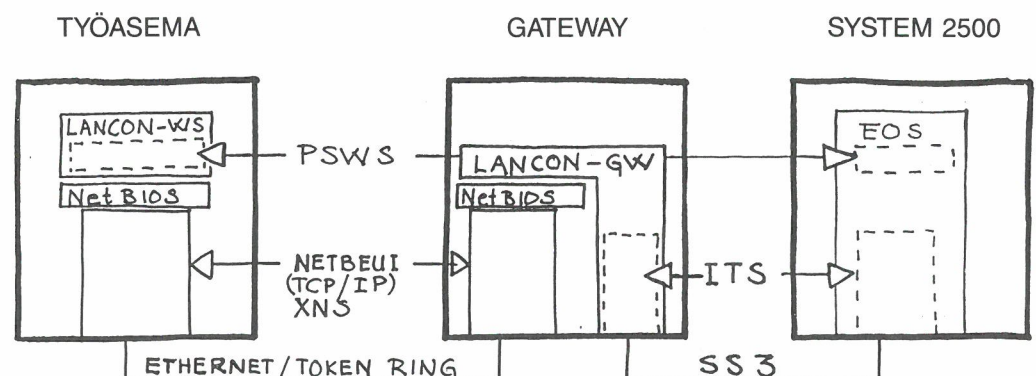
S2500-järjestelmän yleisimmät käyttökohteet ovat myynnin, materiaalihallinnon ja taloushallinnon sovellukset auto- (MERCAR) ja tukkukaupalle (ODEN). Sovellusten käyttäjille Tiimi tarjoaa henkilökohtaiset työasemat, kehittyneen työryhmäverkon ja liittymät valmiisiin sovellusjärjestelmiin ja tietokantoihin. Toimisto-Tiimi puolestaan mahdollistaa nykyaikaisen toimistoautomaation.

Jatkossa Tiimin ja S2500:n välistä liittymää kehitetään edelleen. Tavoitteena on esimerkiksi S2500:n SAGA II -asiakirjojen ja ToimistoTiimin eri sovellusohjelmien tiedostojen välisen tiedonsiirron helpottaminen.



Glen-Koskela työskentelee Nokia Datan Tuotemerkkioinnissa S2500-tuotteiden tuotepäällikkönä.

LANCONIN protokolla-arkkitehtuurissa on kolme porrasta.



Kilpailijat saman pöydän ympärillä

Löydettyään suorittavan ja ohjaavan portaan välistä Tiimille sopivan markkinaraon, Nokia Data ryhtyi etsimään sopivia yhteistyökumppaneita, joilta löytyisi sekä sovellusalueiden asiantuntemusta että mielenkiintoa OS/2:en ja SQLSystemin hyödyntämiseen.

Keijo Tuovinen Unicista kuvaa yhteistyön etuja: ”Laitteet, ohjelmistot ja yritykset uudistuvat nykyään sellaista vauhtia, ettei yksittäinen ohjelmatalo ehdi enää hallita kaikkea. Yhteistyö tarjoaa kuitenkin ratkaisun, sillä nyt kukaan voi panostaa siihen minkä todella osaa.”

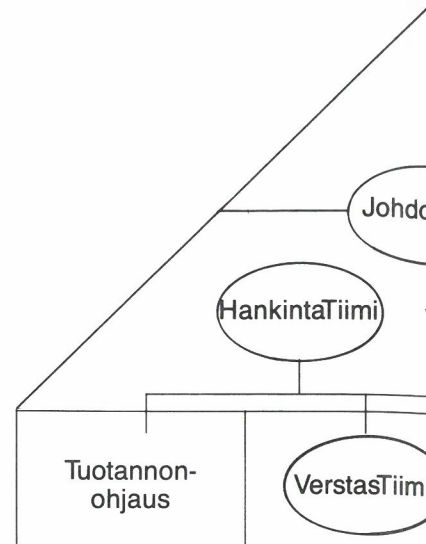
Työnjako on hahmottunut kiinnostavasti kunkin yhteistyökumppanin painopistealueen mukaan: Procons on keskittynyt ostojen ja hankintojen ohjaukseen — HankintaTiimi, Unic luo päättäjien tietotilaamon — JohdonTiimi, ja Länsitieto kokoaa markkinoinnin kannalta ratkaisevat tiedot yhteen — MyyntiTiimi. Lisäksi Nokialla on oma Versta-Tiimin nimellä kulkeva operatiivinen tuoteverstaan ohjausjärjestelmä.

Yhteinen Tiimi — sovellusarkkitehtuuri

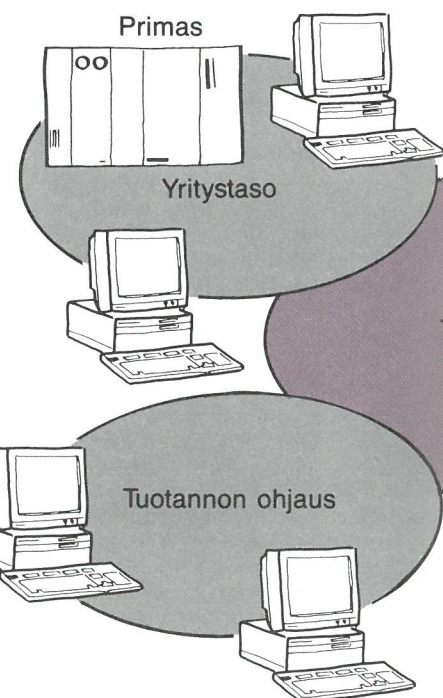
Vaikka sovelluksia on koottu useilla eri tahoilla, ne ovat saman tuoteperheen jäseniä paitsi toimintoiltaan myös toiminnaltaan. Suunnittelun pohjana on Nokian tekemä tyyliopas, joka määrittelee käyttöliittymän ilmeen ja toimintaperiaatteet. Käyttäjän on voitava siirtyä järjestelmästä toiseen ilman, että hänen täytyy opetella kaikki alusta uudelleen. Lisäksi yhteistyökumppaneiden ja

Markkinat ovat täynnään suorittavan tason operatiivisia järjestelmiä. Yksittäiset toiminnot tehostuvat, mutta kokonaiskuvan hahmottaminen on jopa entistä vaikeampaa, sillä johdon tarvitsemien tietojen kokoaminen ja käsittely tapahtuu usein kaikkien tietojärjestelmien ulkopuolella.

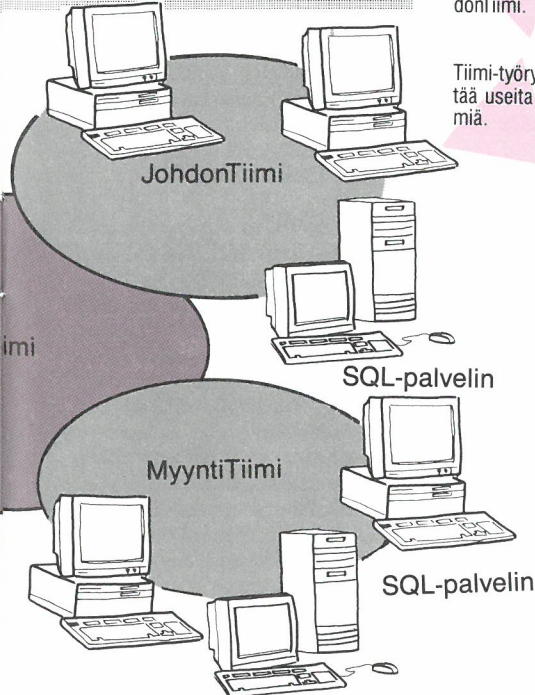
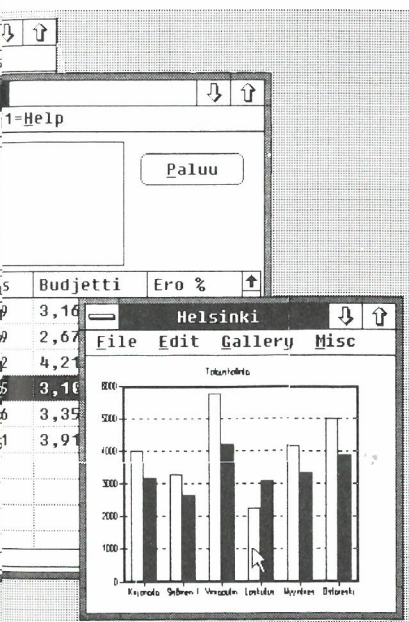
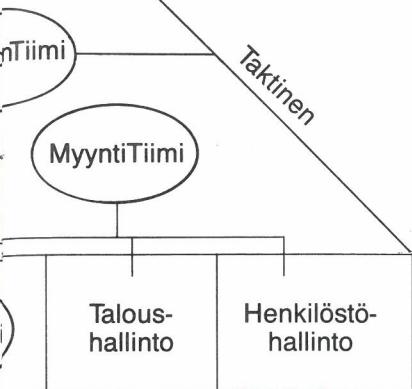
Nokia Data on rakentamassa yhdessä Proconsin, Unicin ja Länsitiedon kanssa joukkoa Tiimi-pohjaisia sovelluksia, jotka mahdollistavat myös taktisen tason suunnittelun ja ohjauksen.



JOHDON JÄRJESTELMÄ		
Arkisto	Perustiedot	Raportit Selaus
JohdonTiimi		
Arkisto	Muokkaus	Grafiikka Tasot
Tasot:		
Oy Teollisuus Ab		
Helsinki		
N:o	Tuote	Laskutus
1001	Elektroniikka	4,020.5
1002	Laakerit	3,318.7
1003	Hydrauliikka	5,765.1
1011	Sähkömoottorit	2,275.2
1012	Polttimoottorit	4,191.2
1013	Generaattorit	5,029.5



Strateginen



Operatiivinen

Eri Tiimien käyttöliittymät ovat samantyyppiset. Kuvassa JohdonTiimi.

Tiimi-työryhmäverkko voi sisältää useita eritasoisia järjestelmiä.



Procons

Nokian edustajista koostuva tekninen työryhmä valvoo osaprojektien yhtenäisyyttä.

Järjestelmät rakentuvat SQLWindowsin ja SQLBasen varaan. Sovellus pyörii työasemassa SQLWindowsin alla, ja tietokanta on palvelimessa SQLBasen takana.

Käsiteltävät tiedot päivitetään eri operatiivisen tason järjestelmistä säännöllisin väliajoin (esim. kerran päivässä). Tiimit joudutaankin räätälöimään aina kunkin organisaation olemassaoleviin järjestelmiin sopiviksi.

Tiimien 386-pohjaisten työasemien (MM4) käyttöjärjestelmänä on MS-DOS (Windows), ja palvelimissa käytetään OS/2:ta. Tulevaisuudessa myös työasemissa siirrytään OS/2-maailmaan.

Procons hallitsee hankintajärjestelmät

HankintaTiimin tekijä Procons perustettiin alun perin vuonna 1979 Valmetin ja Fiskarsin yhteistyönä. Nykyään se on Valmet Automationin tytäryhtiö, joka on erikoistunut teollisuuden toiminnanohjaus- ja CAD/CAM-järjestelmiin.

HankintaTiimin ideana on tarjota tukea sekä itse ostotilanteissa että pitkän tähtäimen hankintasuunnitelmia tehtäessä. Se kerää tietoa alemman tason ostojärjestelmiltä ja antaa informaatiota hankintasopimuksista, -ennusteista, -lähteistä ja kuljetuksista. Myös hankintaprojektien kokonaistilanteen seuranta onnistuu.

Länsitieto myy ja markkinoi

Länsitieto syntyi puolestaan Rosenlewin atk-yksiköstä vuonna 1983. Kun Rauma Repola osti Rosenlewin, Länsitieto siirtyi Modul-ryhmään. Myös Länsitiedon painopisteenä on Teollisuuden kokonaistietojärjestelmät, mutta markkinointihenkeä.

MyyntiTiimi mahdollistaa myynnin ja markkinoinnin keskitetyn suunnittelun ja ohjauksen. Yhdistetty asiakas-, tuote- ja kilpailijatietous mahdollista paitsi markkinaosuuksien ja toimialatrendien seurannan myös myyntikampanjoiden suunnittelun, toteutuksen ja analysoinnin.

Unic tislaa tietoa informaatiotulvasta

Unic on Tiimi-sovelluksia tekevä ohjelmistotaloista vanhin. Se on perustettu jo vuonna 1967, ja omimpina alueina ovat olleet taloushallinnon ja johdon järjestelmät.

JohdonTiimi helpottaa johdon työtä sen yrittäessä selvittää, mitä liikeorganisaatiossa todella tapahtuu. Perisuomalainen kirjantutkimuksen huomaa jotakin olevan tekeillä vasta kun raha kilahtaa tilille — sitä ennen on musta aukko, josta ei ole tietoja.

Ideana on asentaa mittareita (työasemia) organisaation kriittisiin pisteisiin. JohdonTiimi kokoaa mittareiden antamat tiedot yhteen, jolloin saadaan kuva koko organisaation toiminnasta — ja pullonkauloista. Järjestelmä ja mittarit voidaan asentaa myös olemassaolevien järjestelmien päälle.

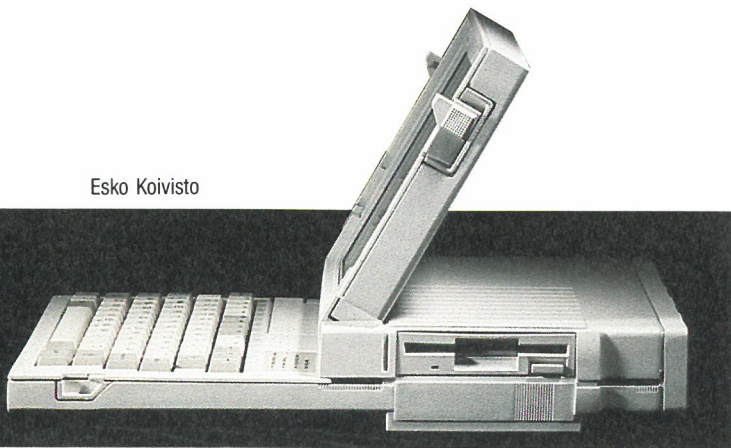
Yhteistyö on hauska toimintamalli

Yhteistyö on koettu eri osapuolien taholta toimivaksi, jopa hauskaksi toimintamalliksi. Jussi Vanhahonko Länsitiedosta ihmettelee, miten kivuttomasti kilpailijat on saatu saman pöydän ääreen.

Markkinoinnista vastaavat kaikki yhdessä siten, että Nokia Data-myy ja markkinoi kaikkia järjestelmiä ja Procons, Unic sekä Länsitieto omiaan. Yhteisrintamaa on helppo pitää yllä, sillä päällekkäisyydet on saatu ennakko-suunnittelulla karsittua pois. ■

Käytössä koettua

Esko Koivisto



Sormiini takertunut laite oli prototyyppi eikä muistuttanut esimerkiksi oheisissa kuvissa komeilevaa konetta kuin sisuksiltaan. Jopa väri oli yönmusta. Sisälmykset olivat sen sijaan samat näppäilytuntumaa myöten eli silmät kiinni ja töihin.

Kova kone

Käyttötesti muodostui matkasta Kaakkois-Suomeen ja takaisin. Pomppiminen linja-autoihin ja pois matkalaukkujen ja sylimikron kanssa tarjosi monta tilaisuutta särkeä matkatavaransa. SyliMikon metallirunko piti kuitenkin huolta siitä, että kaikista muista tavaroista on tätä nykyä kulmat sisällä paitsi koneesta itsestään. Myöskään hiekkaiset puutarhapöydät eivät pystyneet turmelemaan laitteen ulkonäköä. Tuotantoversion vaalea väri voi tosin ottaa helposti itseensä, mutta metalli kestää myös puhdistusaineiden käyttöä aivan toiseen tapaan kuin muovi.

Näppäilytuntuma on suht' jämäkkä, ja näppäimet on sijoitettu loogisesti. Ensi alkuun vain caps lock tuppasi lipsahtamaan joskus päälle, mutta sitäkin oppi äkkiä varomaan. Numeronäppäimistön puutekaan ei päässyt häiritsemään, sillä Insin ja Delin kaltaiset erikoisnäppäimet on jätetty jäljelle. Kohdistimensiirtonäppäimiä on neljä eli niistäkään ei pääse sanomaan pahaa sanaa.

Mitä tapahtuu, kun vanha laite-testaaja saa uuden SyliMikon proton puolivahingossa sormiinsa? Niinhän siinä käy, että muistivihon sivut alkavat täytyä käyttökokeuksista — vanhat paheet kuolevat hitaasti ja sitä rataa. Mutta mitä vanha mikromurskaaja oli SyliMikosta mieltä?

Hiljaa yössä

Aiemmin salkkumikroja matkoilla käyttäessäni ovat yöt osoittaneet melkoiseksi ongelmaksi. Levykeasemat kun ovat tupanneet pitämään melkoista rousketta, josta esimerkiksi junan makuuvaunussa saa pian kuulla kunniansa. Junaan asti en SyliMikolla päässyt, mutta ei kaikkea sentään tarvitse käytännössä kokeilla, kun on korvatkin annettu. Jos joku vielä tästä hurinasta valittaa niin ihme on. Saisipa vielä kaiuttimesta äänen pois niin voisi jopa pelata vapaasti.

Akutkin saa riittämään pitkälle yöhön kun muistaa asettaa näytön ja kiintolevyn virransäästötilaan. Kone myös varoittaa akkujen loppumisesta niin hyvissä ajoin, että työt ehtii saattaa sopivaan kuntoon ennen tallentamista.

Yksissä kuorissa

MikroMikko 4 m310 on siitä mukava kone, että kun sen ottaa mukaansa, muuta ei sitten tarvita. Kunhan puhelinlinja löytyy, etätyöskentely onnistuu mistä tahansa. Oheislaitteissakin löytyy aina savukkeensytytinliitäntään asti.

Mahdollisuus kytkeä SyliMikoon erillinen näppäimistö ja näyttö pakotti toki kokeilemaan eli paiskasin kotosalla vanhan myllyn nurkkaan ja jätin oheisromut pöydälle. SyliMikko sai leikkiä keskusyksikköä — ja hyvinhän se 386 leikkii ei siinä mitään. Sääli etten ehtinyt hankkia mistään Xenixiä, sitten homma olisi käynyt todella mielenkiintoiseksi. Kyllä siinä meinasi poru tulla, kun piti nostaa vanha rousku takaisin pöydälle ja viedä lelu takaisin.

Lähtökohtana SyliMikon valintaan on ollut vanhan asiakaskunnan vaatimustaso. MikroMikkoonsa tottunut — joka neljäs suomalainen mikronkäyttäjä — ei hyväksy kannettavaksi ”kakkoskoneeksikaan” käyttömukavuudeltaan tai suorituskyvyltään puutteellista laitetta.

Tekniikan kannukset

MikroMikko 4 m310:n sisällä sykkii 12,5 megahertsin taajuudella kelloitettu 80386-suoritin. Suorituskyyä riittää siis niin kotiin, kentälle kuin konttoriinkin. Käyttöjärjestelmänä on MS-DOS 3.3.

SyliMikossa on käytetty vähän virtaa kuluttavia CMOS-komponentteja. Numeronmurskauskykyä voidaan kasvattaa erillisellä aritmeettikkasuorittimella, jos laitetta käytetään esimerkiksi vektorigrafiikan tai taulukoiden käsitelyyn.

MikroMikko 4 m310 on varustettu yhden megatavun muistilla, joka voidaan laajentaa suoraan emokortille kahteen, neljään tai kahdeksaan megatavuun.

Näyttönä on elektroluminenssilla taustavalaistu 10-tuumainen VGA-nestekidenäyttö. Se kykenee tulostamaan 16 eri harmaasävyä 640×480 pisteen erottelutarkkuudella. Valkealle taustalle tulostuvien mustien merkkien luettavuus säilyy tavanomaisista nestekidenäyttöistä poiketen hyvänä, vaikka valaistusolosuhteet vaihtelevatkin. Näin saadaan hyödynnettyä nestekidenäyttöjen vähäistä virrankulutusta ilman käyttö-mukavuuden heikkenemistä.

Akku tai virtalähde

SyliMikkoa voidaan käyttää joko akulla tai verkkovirralla. Koneen mukana toimitetaan verkkovirralla toimiva virtalähde, jota voidaan käyttää sekä sisäisesti että ulkoisesti.

Maastossa virtalähteen tilalle voidaan vaihtaa akku, jonka lataus onnistuu normaalin verkkokäytön aikana. Akun kestävyyttä voi tarpeen vaatiessa pitkittää asettamalla näyttö ja kiintolevy virransäästötilaan.

Lisäksi SyliMikkoa voidaan käyttää autossa ja veneessä savukkeensytyttimen adapterin kautta erillisellä liitäntäjohdolla.

Mikonpäivien yllätysuutuus

SyliMikko kulkee mihin vaan

Näppäimistöä modeemiin

MikroMikko 4 m310:n takapaneeliin on sijoitettu monipuolinen valikoima liitäntöjä. Kiinteä näppäimistö voidaan korvata ulkoisella DIN-liitäntän kautta. Takapaneelista löytyy myös standardi rinnakkaisliitäntä sekä 9-pinnainen sarjaliitäntä.

Ulkoisia optioita varten on oma liitäntänsä. Sen kautta koneeseen voidaan liittää esimerkiksi 5,25 tuuman levykeasema tai 40 megatavun nauhavarmistusasema.

Videoliitäntään voidaan liittää ulkoinen VGA-näyttö. Lisäksi takapaneelista löytyy modeemiliitäntä, jolle on käyttöä mikäli järjestelmä sisältää optiona saatavan sisäisen modeemin.

Akun kestävyyttä voi pitkiä aikoja asettamalla näyttö ja kiintolevy virransäätötilaan.

Mukavasti mukana

MikroMikko 4 m310 on Suomen markkinoiden kevyin 386-pohjainen mikrotietokone. Alle kuusi kiloa painavan laitteen saavat muutkin kuin voimamiehet liikumaan.

MikroMikko 4 m310:n kotelo on poikkeuksellisesti kolhuilta suojaavaa iskunkestävää magnesiummetallia. Sen voi siis viedä kentälle hankaliinkin paikkoihin — pehmustettuine olkalaukkuihin — pehmustettuine olkalaukkuihin.

SyliMikko on hiljainen matkakumppani. Ainoat korvin kuultavat elonmerkit syntyvät massamuistitoimintojen aikana sekä kiinteästä 72-näppäimisestä näppäimistöä. Koon ja painon pienentämiseksi numeronäppäimis-

tö on sisällytetty kirjainnäppäimistöön. Laitteeseen voidaan kuitenkin liittää myös ulkoinen AT/E-yhteensopiva näppäimistö (esim. Nokia 102) tai erillinen numeronäppäimistö.

Kantokahva on sijoitettu koneen takaosaan. Se voidaan lukita ala-asentoon, jolloin koko laite saadaan kallistettua mukavampaan työskentelyasentoon.

Sisäisiä ja ulkoisia

MikroMikko 4 m310:ään voi valita joko 20 tai 40 megatavun kiintolevyaseman. Molempien kiintolevyjen haku aika on 27 millisekuntia. MikroMikko 4 -tuoteperheen mukaisesti myös m310:ssä on sisäänrakennettu 3,5 tuuman 1,44 megatavuinen levykeasema. Asemassa voidaan lukea, kirjoittaa ja alustaa myös 720 kilotavun levykkeitä. Tarvittaessa SyliMikoon voidaan liittää ulkoisesti myös 5,25 tuuman levykeasema.

Tien päältä ja kotoa työskentelyä varten MikroMikko 4 m310:een on saatavana sisäänrakennettu 2400 baudin modeemi. Heikkojen linjojen yli yhteyttä voi yrittää myös 1200 tai 300 baudin nopeudella. Modeemi toimii Hayes Smartmodem tavoin eli siinä on MNP tason viisi virheenetsintä ja korjaus sekä tiedon tiivistys.

Lisäksi numeroiden syöttöä helpottamaan voidaan hankkia ulkoinen numeronäppäimistö. Sisäänrakennetun kiintolevyn varmistukseen on tarjolla ulkoinen 40 megatavuntavun nauhavarmistusasema.

Virtalähteen aukossa on laajennusväyläliitäntä, johon MikroMikko 4 m310:n myöhemmin markkinoille tulevat laajennukset kuten verkkomoduli tai laajennusyksikkö liitetään.

Eli MikroMikko 4 m310 kehittyy myös jatkossa eikä jää paikalleen polkemaan. Kestävä ja kulkevana työvälineenä se säilyttää asemansa vielä pitkään. ■



Tekniset tiedot:

MikroMikko 4 m 310

Suoritin:

Intel 80386 — 12,5 MHz

Käyttöjärjestelmä:

MS-DOS 3.3

Muisti:

1/2/4/8 Mt

Massamuistit:

3,5" — 1,44 Mt levykeasema

20/40 Mt kiintolevyasema

Näyttö:

10" taustavalaistu LCD (VGA)

640x480 kuvapistettä

Näppäimistö:

72 näppäintä, 12 toimintonäppäintä

Liitännät:

Sarjaliitäntä (9-pin)

Rinnakkaisliitäntä

Oheislaiteliitäntä

Näppäimistöliitäntä RGB

Puhelinliitäntä (modeemi)

Laajennukset:

Sisäinen 2400 baudin virheen-

korjaava modeemi 5,25" —

1,2Mt levykeasema

40 Mt nauhavarmistusasema

Numeronäppäimistö

Savukkeensytyttimen pistoke

Kotelo:

Magnesiummetallia

Virtalähde:

Verkkovirta

Ladattava akku

Paino:

5,1 — 6,0 kg

lev*kork*syv:

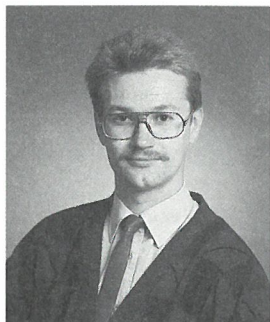
290*58*380 mm

Nokia Data tuo uutuutena markkinoille kauan kaivattua kannettavan SyliMikon. Mallinimeltään uutuus on MikroMikko 4 m310 eli se täydentää keväällä esiteltyä MikroMikko 4 -perhettä sieltä kevyemmästä päästä.

Tuhat ja yksi tapaa pilata selkensä ja silmänsä



Mikro ei ole tuottavin investointi



Timo Kurki työskentelee Teknillisen Korkeakoulun työpsykologian laboratoriossa vs. assistenttina sekä Työterveyslaitoksen työturvallisuusosastolla apulais-tutkijana.

Mikrotietokoneet ovat olleet jo jonkin aikaa toimistojen arkipäivää. Kuitenkin edelleen on vallalla käsitys, että mikron voi nostaa pöydälle kuin pöydälle ja huoneeseen kuin huoneeseen. Käyttäjästävälisessä mikrosta voi tulla epämieluisa työväline, jos sen ympäristö, kalusteet ja sijoittelu eivät ole kunnossa.

U seimmat toimistotilat ovat sellaisenaan sopivia mikrokäyttöön. Joitakin seikkoja on kuitenkin syytä ottaa huomioon ennen mikron sisäänkantamista.

Huoneen sisustusmateriaaleille on asetettava joitakin vaatimuksia. Seinäpinnat eivät saa olla liian vaaleita, koska ne voivat heijastua häiritsevästi varsinkin tummapohjaisten näyttöpäätteiden kuvaputken pinnasta. *Grandjeanin* suosittelamia heijastussuhteiden arvoja pintamateriaaleille on esitetty taulukossa 1. Heijastussuhde määritellään pinnalle tulevan ja sen lähettämän valaistusvoimakkuuden suhteena.

Jos valaistus on muuten kunnossa ja mainittuja suosituksia noudatetaan, toimistossa esiintyvät valotiheysarvot (luminanssit) ovat tasapainossa. Silmälle on erittäin rasittavaa joutua jatkuvasti mukautumaan eri kirkkaussasteisiin. Sisustus, jossa on esimerkiksi mustaa nahkaa, kromia ja hohtavan valkoisia pintoja, on kaikkea muuta kuin soveltuva kunnan tietotyöhön.

Tietokoneiden ja niiden oheislaitteiden lämpökuorma on myös otettava huomioon. Tämä ei ole tavallisten mikrotietokoneiden yhteydessä kriittinen tekijä, mutta esimerkiksi jotkut CAD-laitteet saattavat tuoda työtilaan jo-

pa kaksi kilowattia lämpötehoa. Vaikutus on sama kuin kahdella työtilassa jatkuvasti pyörivällä hiustenkuivaajalla. Tällöin on otettava avuksi koneellinen jäähdytys. Jäähdytysjärjestelmän tulisi olla keskitetty, sillä huonetilaan sijoitettu paikallinen jäähdytyskone aiheuttaa vetoa ja melua.

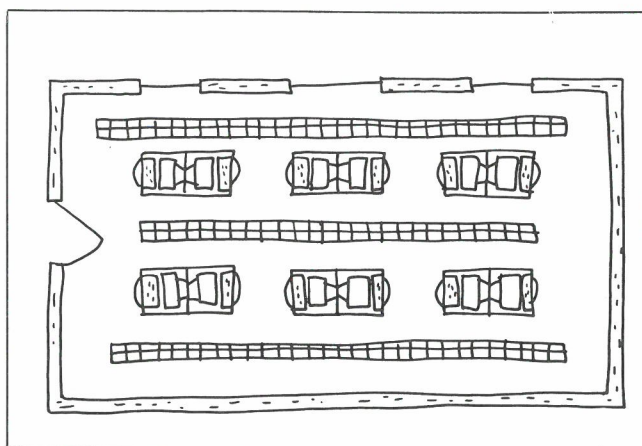
Valaistus on päätettyön murheenkryyni

Näyttöpäätetyö asettaa valaistukselle ristiriitaisia vaatimuksia: valoa on oltava riittävästi papereiden lukemiseen, mutta liiallinen valo vähentää etenkin tummapohjaisilla näytöillä merkkien luettavuutta. Lisäksi liian kirkas valaistus aiheuttaa usein kiusallisia heijastuksia kuvaruodun pinnalle.

Työtilassa tulisi välttää myös suuria valontheysarvoja. Nämä vaatimukset asettavat luonnollisesti vaatimuksia myös valaistukselle. Suositeltava valaistusvoimakkuus tummapohjaisille näyttöpäätteille on 500 luksia, jos työskenneltäessä on luettava myös papereita. Muuten valaistusvoimakkuus voi olla 300 luksia. Vaaleapohjaisille näyttöpäätteille voidaan suositella 500 — 800 luksin valaistusvoimakkuuksia eli miltei normaalia toimistovalaistusta. Luminanssien suhteiksi näkökentässä suositellaan 1:3 — 1:5.

Mainittujen ongelmien ratkaisemiseksi on kehitetty monenlaisia valaistusratkaisuja. Epäsuora valaistus on yksi vaihtoehto. Siinä valo heijastetaan työkohteeseen katon tai erillisen varjostimen kautta. Tällöin valaisimien pinnat eivät ole näkyvissä ja näin ne eivät voi näkyä heijastuksina kuvaruodussa. Muuten mukavan valaistuksen antavan järjestelyn haittapuolena on suurempi valaistustehon tarve sekä vaatimukset huonetilan katolle ja korkeudelle.

Toinen näyttöpäätetiloissa käytetty ratkaisu on niin kutsuttujen matalaluminanssivalaisimien käyttö. Tällaisissa, yleensä loisteputkivalaisimissa, valo on suunnattu kapeana keilana alaspäin. Valaisinta sivulta katsottaessa loisteputket eivät ole näkyvissä,



Kuva 1. Tietokoneet voidaan sijoittaa suurempaan työtilaan vaikka näin.

Katot	70 %
Ikkunat (kaihtimet ja verhot)	50 %
Seinä näyttöpäätteen takana	40—50 %
Seinä näyttöpäätettä vastapäätä	30—40 %
Lattiat	20—40 %

Taulukko 1. Suositeltavat heijastumissuhteet eri pintamateriaaleille.

- Valonlähde ei saa olla työntekijän näkökentässä hänen tehdessään normaaleja työtehtäviä.
- Valonlähteiden tulee olla niin suojattuja, etteivät luminanssiarvot näkökentässä ylitä 200 cd/mFD.
- Katseen suunnan ja valonlähteen välisen kulman tulee olla enemmän kuin 30 astetta. Jos suurissa tiloissa ei päästä tähän, on käytettävä parempaa valaisinten suojauksia.
- Loisteputket tulee asentaa työntekijän kummallekin puolelle katseen suuntaiseen jonoon.
- On parempi käyttää useampaa heikkotehoista lamppua kuin yhtä suuritehoista.
- Lamppujen valo ei saa heijastua silmiin työpinnoista.
- Heijastavien materiaalien käyttöä tulisi välttää työpöydällä ja työvälineissä.

Valaistuksen suunnittelijan tarkastuslista.

Katseluetäisyys	500	700	900
Jatkuva tekstin luku	3,0	3,5	4,5
Tekstin silmäily	3,3	4,7	6,0
Standardiehdotuksen suositeltava alaraja: 20 kulmaminuuttia	2,9	4,1	5,2
Standardiehdotuksen ehdoton alaraja: 18 kulmaminuuttia	2,6	3,7	4,7

Taulukko 2. Merkkien vähimmäiskoot (mm) eri katseluetäisyyksillä kirjallisuuden ja SFS:n standardiehdotuksen mukaan.

vaan ne ovat ritilän suojassa ja näin ne eivät suuressakaan maisemakonttorissa näy häiritsevinä heijastuksina kuvaruodussa tai häikäise kirkkaudellaan näkökentässä.

Valaisimien tyypin lisäksi valaistusolosuhteisiin vaikuttaa valaisimien ja tietokonepöydäpisteiden sijoittelu. Suuressa maisemakonttorissa valaisimien sijoittelulla on keskeinen merkitys. Valaisimet tulisi sijoittaa niin, etteivät ne näkyisi näkökentässä eivätkä heijastuisi näyttöpäätteen pinnasta. Maisemakonttorissa on käytettävä valaisintyyppinä esimerkiksi matalaluminanssivalaisimia ja sijoitettava ne työpisteiden sivuille jonoon. Huoneessa näyttöpäätteen tulee sijoittaa suorakulmaisesti ikkunoihin nähdä. Kuvassa 1 on esitetty näyttöpäätetyöpisteiden sijoittaminen maisemakonttoriin ja kuvassa 2 yksittäiseen huoneeseen.

Kalusteet ovat hyvin tuottavia investointeja

Ikkunan edessä olevalle kirjoituspöydälle nostettu mikrotietokone on valitettavan tavallinen. Kun monitori vielä nostetaan keskusyksikön päälle alkaa ”täydellinen” työpiste olla valmis. On helppo ymmärtää, ettei käyttökelpoisia kalusteita haluta heittää menemäänkään, mutta kunnollisten kalusteiden hankkiminen saattaa pidemmällä tähtäimellä olla hyvinkin kannattavaa. Sairaushoitoa ja heikko työteho ovat merkittäviä kustannustekijöitä. Onkin nurinkurista, että tietokoneiden hintojen laskettua säästyneitä markkoja ei käytetä työpisteiden ergonomian parantamiseen, vaan niillä hankitaan lisää koneita.

Kalusteiden tärkein ominaisuus on niiden muunneltavuus erilaisten käyttäjien ja työskentelytapojen mukaan. Riittävät säädöt takaavat kalusteiden sopivuuden erikokoisille ihmisille ja erilaisiin työtehtäviin. Jos samaa työpistettä käyttää päivittäin useampi kuin yksi työntekijä, säätöjen tulisi olla niin helppoja ja nopeita, että ne viitsitään tehdä. Kunnollisessa päätepöydässä on monitorille ja näppäimistöille eril-

Kirjallisuutta:

Grandjean, E. (1987) **Ergonomics in computerized offices**. London, New York, Philadelphia: Taylor & Francis.

Grandjean, E. (1984) **Postural problems at office machine work stations (introductory paper)**. Teoksessa Grandjean, E. (toim.) *Ergonomics and health in modern offices*. 445-455. London and Philadelphia: Taylor & Francis.

Lehtelä, J. (1986) **Mikrotietokoneiden ja tekstinkäsittelylaitteiden ergonominen arviointi**. Työterveyslaitos, Helsinki.

Lehtovaara, J. (1987) **Näyttöpäätetyön valoteknisiä ongelmia**. Sähkö, 60, 18-21.

Mandal, A. (1984) **What is the correct height of furniture?** Teoksessa Grandjean, E. (toim.) *Ergonomics and health in modern offices*, 471-476. London and Philadelphia: Taylor & Francis.

Mandal, A. (1987) **The influence of furniture height on backpain**. Behaviour and Information Technology, 6(3), 347-352.

Sihvonen, T., Baskin, K., Järvinen, H., Riikonen, A. ja Hänninen, O. (1987) **Tietotekniikka toimistotyössä ja niska-hartiaseudun kuormittuminen**. Suomen Lääkäri-lehti, 42(25), 2283-2290.

Suomen Standardisointiliitto SFS r.y. (1986) **SFS-käsikirja 72 Näyttöpäätetyöskentely toimistoympäristössä**. Vaatimukset ja varustus. Helsinki: Kyrrii oy.

Suomen Valoteknillinen Seura r.y. (1986) **Valaistussuositukset, sisävalaistus**. Julkaisu n:o 9-1986. Jyväskylä: Gummerus.

liset säädettävät tasot. Monitorin korkeuden tulee olla sellainen, että kuvapinnan yläreuna on silmien tasalla. Keskusyksikön sijoittaminen monitorin alle ei yleensä ole hyvä ratkaisu, sillä katselukohde tulee tällöin liian korkealle. Jos etäisyys on yli 50 senttiä, merkkien koko ei normaaleissa näyttöpäätteissä riitä. Taulukossa 2 on esitetty merkkien koko eri katseluetäisyyksillä.

Pelkkä pöytä vain monitorille ja näppäimistöille ei riitä, sillä yhä useampaan mikrotietokoneeseen

kuuluu hiiri tai muu osoitinlaite, joka vaatii oman liikuttelutilansa. Harva tietotyökään on luonteeltaan sellaista, ettei siinä tarvita papereita. Niiden tutkimiselle on varattava tilaa. Jos työ sisältää tiedon syöttöä tai vertailua paperidokumenttien kanssa, on kuvaruudun etäisyydelle sijoitettu teline erittäin hyvä apuneuvo. Sama etäisyys sekä paperille että kuvaruudulle pienentää silmän tarkennusjärjestelmän työtä ja vähentää näin silmien rasittumista.

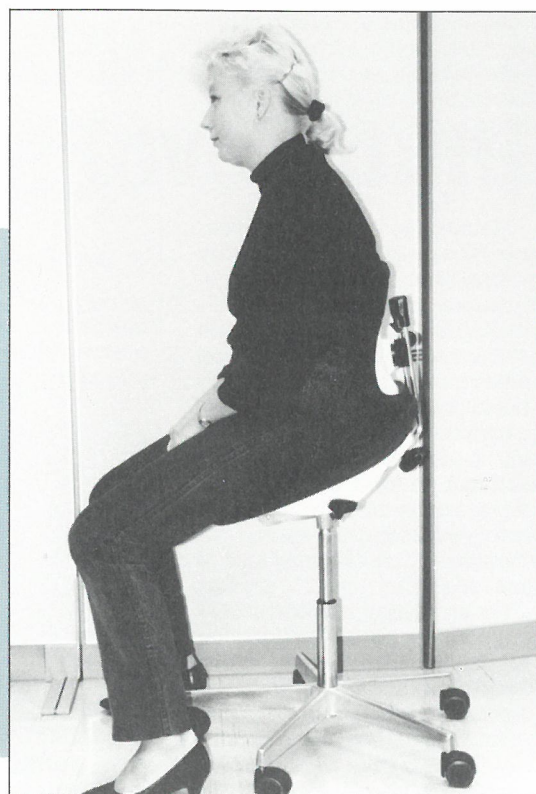
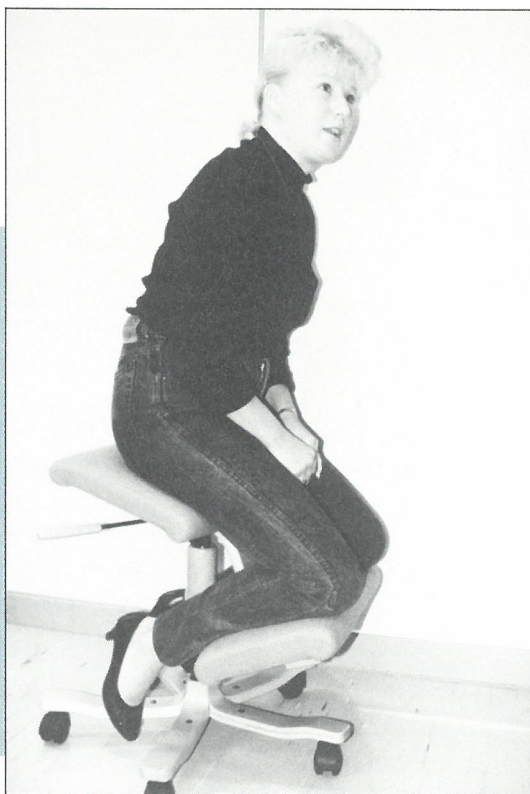
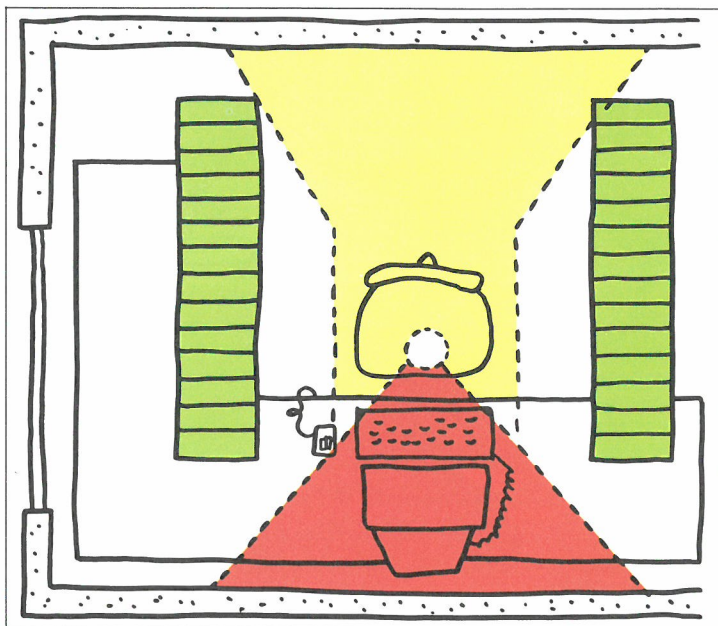
Insinööri työskentelee vaikka päällään

Moni tietotekniikan ammattilainen tai käyttäjä saattaa uppoutua työhönsä niin, ettei kiinnitä työskentelyympäristöönsä tai -asentoonsa mitään huomiota. Illalla kuitenkin ihmetellään hartioiden kipuja. Usein niska-hartiaseudun kipujen syynä ovat jännittyneet

Työympäristön suunnitteluun kannattaa ottaa mukaan asiantuntijat — työntekijät.

ja yksitoikkoiset työskentelyasennot. Ei voida antaa yhden ja ainoan oikean työskentelyasennon suositusta, mutta tärkeintä olisi oppia vaihtamaan työasentoja. Jotkut edistyneimmistä tutkimuksista esittävät ratkaisuksi toimistoja, jotka muistuttavat enemmän kuntosalia kuin tavanomaista konttoria. Tällaisessa toimistossa esimerkiksi tekstinkäsittelyä tehtäisiin laitteessa, joka simuloisi portaissa kiipeämistä.

Vaikka edellämämainitun kaltaiset visiot ovat ehkä enemmän utopiasa kuin tulevaisuutta, tosiasiaksi jää, että ihminen liikkuu normaalissa toimistotyössä aivan lii-



Kuva 2. Tietokoneen ja valaisimien sijoittaminen työhuoneeseen.

Kuva 3. Polvituolessa paino jakautuu säärillem, polville, reisille ja pakaroillem.

Kuva 4. Mandalin tuoli puolestaan siirtää osan painosta jaloille.

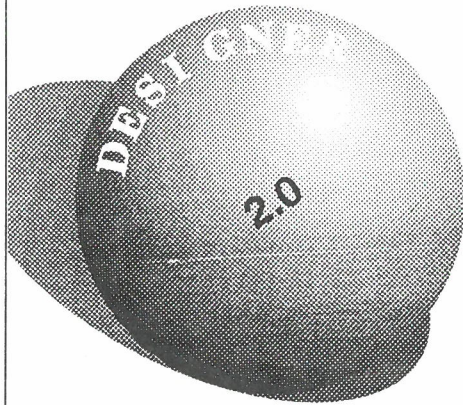
an vähän ja istuu liikaa paikallaan "sähköisen silmän" vangitsemana. Ratkaisuksi tähän ongelmaan tarvitaan monipuolisesti säädettävää työpistettä, ja huomion kiinnittämistä omaan työskentelyyn. CAD-työhön suositellaan jopa seisomatyöskentelyn mahdollistavaa työpistettä. Normaalisissa toimistotyössä työasentoa voidaan vaihdella muuttamalla selkänöjan kallistusta, tuolin tai näppäimistötason korkeutta jne.

Istumisenkin on filosofinen kysymys

Mahdollisuuden erilaiseen työskentelyyn antavat erilaiset tuolit. Viime aikoina on markkinoille tullut uusia erilaisiin istumisfilosofioihin perustuvia istuimia. Yksi näistä on Norjassa alkunsa saanut niin kutsuttu polvituoli, jossa istutaan osittain säärien ja polvien varassa. Tällaisessa tuolissa ei ole erillistä selkänöjaa, sillä selän luonnollinen notko on helppo ylläpitää. Polvituolissa joutuvat selän lihakset kaipaamaansa työhön. Polvituoliin totutteluun menee kuitenkin aikaa. Maanmittaushallituksessa järjestetyssä kokeilussa koeaika oli kuukausi ja vasta sen loppupuolella oltiin tuoliin täysin totuttu.

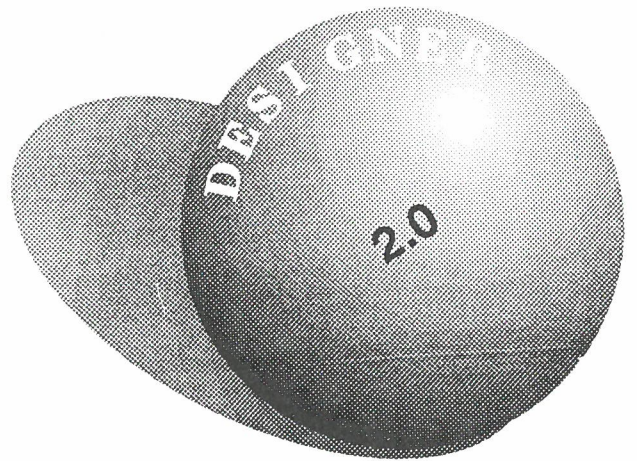
Mandalin kehittämää istumisasennoista edustavat tuolit, joilla istutaan normaalia korkeammalla ja etureunastaan kallistetussa asennossa. Tällainen istumisasento säilyttää myös selän luonnollisen notkon paremmin kuin tavallinen työtuoli. Pöydän korkeuden tulee tällöin olla huomattavasti korkeampi kuin normaalisti, ja pöytälevy on hiukan kallistettava. Osa painosta siirtyy jalkojen kannatettavaksi.

Yleisten suositusten antaminen on vaikeaa. Työympäristön kehittämisen ehdoton ja tärkein lähtökohta on työ itse. Analysoimalla työtä ja selvittämällä sen vaatimukset päästään paljon parempaan lopputulokseen kuin jonkun suosituskokeelman orjallisella noudattamisella. Työympäristön suunnitteluun kannattaa ottaa mukaan asiantuntijat — työntekijät. Heillä on varmimmin tiedossa ne seikat, joihin tulee työympäristön suunnittelussa kiinnittää eniten huomiota. Lisäksi osallistuminen oman työn kehittämiseen lisää työntekijän sitoutumista huomattavasti enemmän kuin konsulttityönä tehty kallis työympäristösuunnitelma.



Designer 2.0:n häivytysväritys muuttaa sävyä pehmeästi.

Väriä elämään



Designer 2.0 piirtää ja sävyttää ►

Micrografxin suosittu Designer piirto-ohjelman versio 2.0* on ilmestynyt markkinoille. Uusi versio ymmärtää useita uusia tiedostomuotoja sekä helpottaa vapaalla kädellä piirtämistä Bezier-käyrien avulla. Aiemmasta poiketen myös kuvien värisävyjä voidaan liu'uttaa tasaisesti.

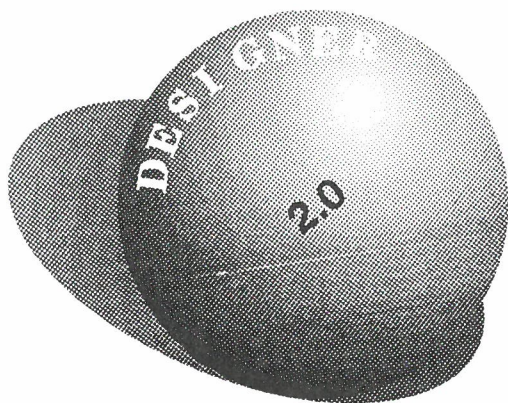
Uudesta Designerista löytyy kaikki piirto-ohjelmalta odotettavat ominaisuudet: kuvanlukijalla luettujen kuvien muokkaus, värien käsittely, Bezier-käyrät, kuviokirjastot ja nippu tulostinvaihtoehtoja.

Vektoreista varjojen kautta käyriin

Designerilla monimutkaisetkin kuviot muodostuvat yksittäisistä piirtoalkioista. Uudessa versiossa myös rasterikuvia voidaan käyttää esityksen pohjana. Kuvat luetaan paperilta kuvanlukijalla esimerkiksi TIF-muotoon (Tag Image File Format), minkä jälkeen ne ladataan Designeriin. Designer muuntaa saamansa rasterikuvat automaattisesti vektorimuotoon (auto tracing). Lopputuloksen osia voidaan värittää ja muokata kuin mitä Designerin omia piirtoalkioita tahansa.

Varjostusten teko on ollut piirto-ohjelmilla perinteisesti työlästä, koska jokaista sävyä varten on täytynyt piirtää oma kuvio. Uuden Designerin häivytysväri-työtoiminnolla väri saadaan liukumaan toiseksi kuvion sisällä ►

*) Designerin ykkösversio esiteltiin Softarissa 3/88.



oheisen kuvan tapaan. Häivytyks voi olla säteittäistä tai lineaarista, mutta sävyjen muuttumisnopeuteen ei päästä vaikuttamaan, vaan muuttuminen on aina tasaista.

Kaikki hiiren avulla piirtäneet ovat huomanneet, miten vaikeaa hyvien kuvien piirtäminen on. Viiva ei läheskään aina synny siihen, mihin käsi on sen asettavinaan. Vapaan piirtelyn avuksi Designer 2.0 tarjoaa Bezier-käyrät. Käyrän käännepisteeseen liittyy tangenttijana, jonka päätepisteitä siirtämällä itse käyrän muotoa voidaan muuttaa. Tämä esitystapa on yleistymässä myös muissa grafiikkaohjelmissa.

*Nyt käyttäjä voi valita
minkä tahansa
valikkokomennon
työkalualueelle ja
poistaa vähiten käytetyt
piirrosvälineet tilaa
viemästä.*

Ikkunasta räätälöintiin

Koska Designer toimii Microsoft Windows -käyttöliittymän alaisena, sen käyttö on muihin Windows-sovelluksiin tottuneelle helppoa. Mikäli vaikeuksiin kuitenkin törmätään, apua voi etsiä hierarkkisesti organisoidusta opastustoiminnosta. Ohjeet ovat englanninkielisiä, sillä ohjelmaa ei ole käännetty suomeksi.

Designerin voi räätälöidä omien tarpeidensa mukaan näytön vasemmassa laidassa olevaa työkalualueetta muokkaamalla. Toimintojen valitseminen työkalualueelta on myös valikoiden napsuttelua nopeampaa. Aikaisemmassa versiossa piirrostyökalut ja muutama yleisin toiminto oli ikonisoitu työkalualueelle. Nyt käyttäjä voi valita minkä tahansa valikkokomennon työkalualueelle ja poistaa vähiten käytetyt piirrosvälineet tilaa viemästä.

Muunnoksista leikepöydälle

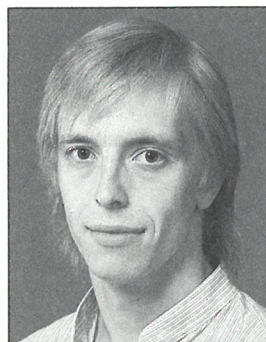
Kuvien siirrettävyys ohjelmasta toiseen on aina ollut eräs grafiikkaohjelmien käyttäjien pahimpia päänsärkyjä. Siirtämisen helpottamiseksi Designer sisältää kuvien esitysmuotoa muuttavat Export ja Import -toiminnot. Nyt Designeriin voidaan lukea Ansitextin lisäksi seuraavia grafiikkatiedostoja: Micrografx Graph, Z-soft Image File (PaintBrush), TIFF, Windows Metafile, Computer Graphics Metafile (CGM) ja AutoCAD DXF. Vastaavasti Designer osaa itse luoda seuraavia tiedostoja: Encapsulated PostScript (EPS), EPS ilman TIFF-esitystä, HP Graphics Language (HP-piirturien ohjauskie- li), Z-Soft Image File, TIFF, Windows Metafile, AutoCAD DXF ja Computer Graphics Metafile.

Tiedostojen ohella Designeriin voidaan siirtää tietoa Windowsin leikepöydän (Clipboard) kautta. Näin esimerkiksi Microsoft Excelin kuvaajat saadaan Designeriin edelleen muokattaviksi.

Keinojen loppuessa otetaan konstit avuksi

Designer on piirrosohjelman hankintaa harkitsevalle tutustumisen arvoinen ohjelma. Se sisältää lähes kaikki kilpailijoiden ominaisuuudet sekä lisäksi joukon omia, vuosien varrella hyviksi havaittuja piirteitä. Designer osaa esimerkiksi käyttää Expanded-muistia monimutkaisten ja isojen kuvien käsittelemiseen.

Kaikkien kuvien tuottamiseen uusi Designerkaan ei sovi. Piirakat, pylväät ja muu liikegrafiikka on edelleen tehtävä jollakin muulla ohjelmalla, esimerkiksi Micrografxin Graph Plussalla. Se on Designerin kanssa molempiin suuntiin yhteensopiva eli osaa sekä kirjoittaa että lukea Designerin tiedostoja.



Jari Luokkala on perehtynyt julkaisuohjelmiin Nokia Datan tuotekehityksen ohjelmistotryhmässä.

Up to now we have been dealing with data abstraction. In part 5 we concentrate on operational abstraction, that is to sub-programs. We will also show how run time error situations (called exceptions in Ada) are handled.

References

- 1/ Reference Manual for the Ada Programming Language, January 1983.
- 2/ Booch, G.: Software Engineering with Ada, Benjamin/Cummings, 1987.
- 3/ Barnes, J.G.P.: Programming in Ada, Addison-Wesley, 1989.
- 4/ Cohen, N.: Ada As a Second Language, McGraw-Hill, 1986.

Learning Ada

Part 5

A subprogram is a good example of abstraction. The programmer knows its name and purpose. By supplying the actual parameters in a subprogram call, he or she gets the results without needing to know how the processing occurs. This kind of abstraction can be called operational to distinguish it from data abstraction, which is implemented with packages in Ada.

There are two kinds of subprograms: procedures and functions. A procedure typically manipulates a global data structure and possibly returns some values in its output parameters. A procedure call is a statement in Ada. A function is used to compute a value, which it returns with a return statement; functions cannot have output parameters in Ada. A function call can be part of a complex expression but it cannot be used as a statement.

In the program text subprograms, like packages, have two parts: specification and body. The specification identifies the name of a subprogram, the number and types of its parameters, and its return type if it is a function. This is all that is needed to call a subprogram.

The body of the subprogram contains the statements that define the algorithm itself. The specification and body can be separated, as we have seen earlier in examples of packages in which the subprogram specification is written in the package specification, and the subprogram body in the package body.

A procedure is specified as follows:

```
procedure SWAP_CHARACTERS (  
  A, B: in out CHARACTER;  
  -- This procedure swaps the contents  
  -- of two CHARACTER-type variables.
```

The procedure has two parameters of the type CHARACTER-type. The interchange is implemented in the procedure body:

```
procedure SWAP_CHARACTERS (  
  A, B: in out CHARACTER) is  
  TEMPORARY: CHARACTER := A;  
begin  
  A := B;  
  B := TEMPORARY;  
end SWAP_CHARACTERS;
```

Here is one way of calling this subprogram:

```
A: CHARACTER := 'X';  
B: CHARACTER := 'Y';  
...  
SWAP_CHARACTERS (A, B);
```

During the procedure call, the value of A becomes 'Y' and that of B becomes 'X'.

Subprogram Parameter Modes

Procedure definitions and calls are very similar to those in Pascal. But note however, the keyword **in out** in the declaration of formal parameters. In Ada we define the direction of parameter passing. Three modes are possible: **in**, **out**, or **in out**. The keyword **in** means that a formal parameter in a subprogram is a constant and permits the associated actual parameter only to be read. The programmer can therefore rely on the value of the actual parameter not changing when the subprogram is executed. The **in** mode is the default mode.

The **out** mode means that a formal parameter is a variable, and enables the value of the corresponding

actual parameter to be written. Only the boundaries of a formal parameter can be read (by using attributes), but the value cannot be read within the procedure, which seems to be rather odd. Note, that if a procedure does not update an **out** parameter, the value to be returned is undefined.

The third alternative, **in out**, allows the programmer to use and modify the value of the actual parameter in a procedure.

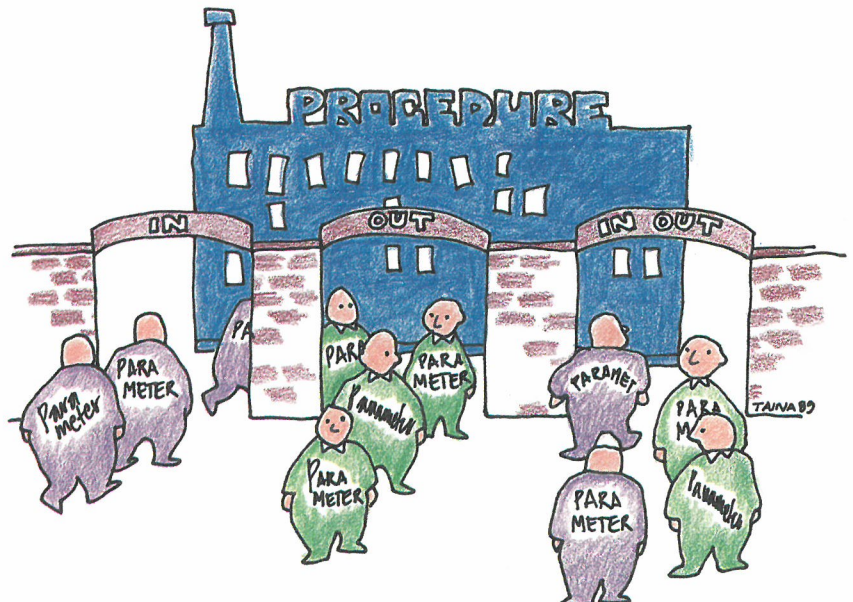
Note, that functions cannot have **out** or **in out** parameters.

In many programming languages, subprogram parameters can be passed by copy or by reference, depending whether the value itself or a pointer to it is copied into the subprogram.

The Ada standard specifies that parameters of scalar types (enumeration, integer, and real types) are transmitted by copy. At the start of a subprogram call the value of the actual parameter is copied into the formal parameter (modes **in** and **in out**). At the end of the subprogram execution the formal parameter value is copied back into the actual parameter (modes **out** and **in out**).

But if the type of the parameter is a record or an array type, the parameter passing mechanism depends on the implementation. The programmer cannot know what is copied into the subprogram. If a pointer is copied, the subprogram accesses the variable in the calling program — not a variable in the subprogram. The actual parameter is modified (unless its mode is **in**).

If the programmer passes an array or record variable into a subprogram and also refers globally to the same variable within the subprogram, the results can be unexpected (depending on the implementation). The simple rule is: never use the names of actual parameters within a subprogram.



Overloading of Subprograms

Ada subprogram identifiers can be **overloaded**. This means we can use the same name for different subprograms within the same program, provided there is some difference between the declarations of the subprograms: i.e. the number of parameters, or at least the types of the parameters are different:

```
procedure PUT (X : INTEGER);  
procedure PUT (X : STRING);
```

This is a typical example of overloading; we have several subprograms doing a similar job for values of different types (predefined or programmer's own types).

Functions as well as procedures can be overloaded. Since predefined operators like "=" and "+" are in fact functions, this means that operators, too, can be overloaded:

```

function "+" (A, B: in MATRIX__TYPE)
  return MATRIX__TYPE is
    R: MATRIX__TYPE;
begin
  ROWS:
    for I in MATRIX__TYPE'RANGE (1) loop
      COLS:
        for J in MATRIX__TYPE'RANGE (2) loop
          R(I, J) := A(I, J) + B(I, J);
        end loop COLS;
      end loop ROWS;
    return R;
end "+";

```

Here we have overloaded the predefined operator `++` to add up two matrixes. `MATRIX_TYPE` is a user-defined data type. Now we can elegantly write:

```
F1, F2, F3 : FLOAT;
M1, M2, M3 : MATRIX__TYPE;
```

F3 := F1 + F2; — Add floating point numbers
M3 := M1 + M2; — Add matrices

Default Parameters

Subprogram parameters can have default values in Ada. For example we can control the number of iterations to be performed when calculating a square root:

```
function SQRT (  
    X      : float;  
    EPSILON : float := 0.001) return float;
```

EPSILON is used in the function body to specify the accuracy of the result. The value of the function call `SQRT (2.0)` is 1.414, but the value of `SQRT (2.0, 0.0000001)` is 1.4142135.

You can also define your own string output operation:

```
subtype STRING5 is STRING (1..5);
procedure PUT (
    A,B,C,D,E,F,G,H,I,J: in STRING5 := ''
```

Now you can output one to ten strings with a single statement. The number of parameters can thus be pseudovisible. For example:

```
PUT ("Value", of  ", S1, " is ", S2, ".  ");
```

assuming that S1 and S2 are strings of length 5.

Note, that it is easier to implement this particular output operation by using directly the PUT_LINE procedure in the TEXT_IO package. This is done by concatenating the strings with the '&' operator:

```
TEXT_IO.PUT_LINE ("Value of " & S1 & " is " & S2 &
" . ");
```

Exceptions

In most programming languages program execution comes to a stop in the event of a hardware malfunction, an overflow in an arithmetic operation or the like. In Ada there is a special mechanism to handle these situations.

Run time errors are called **exceptions** in Ada, and the programmer must take them into account in his or her program. The purpose of exceptions is to indicate events that do not occur during normal operation: *File not found* (when opening a file), or *buffer overflow* are exceptions, but reaching the end of a linked list is not.

In the best case we want the program to remedy the situation. In the worst case, in which total recovery is impossible, such as a peripheral device failure, we would like to continue processing with a reduced capability. Raising an exception brings the error condition to our attention. We can then handle the exception in given parts of the program.

The following exceptions are predefined: `CONSTRAINT_ERROR`, `NUMERIC_ERROR`, `STORAGE_ERROR`, `TASKING_ERROR`, and `PROGRAM_ERROR`. The programmer can also define his own exceptions. Exceptions are easy to declare:

OVERFLOW : **exception;**
UNDERFLOW: **exception;**

If the programmer finds, in programming a stack package, that a pop operation can be applied to an empty stack, he or she can write the statement:

```
raise UNDERFLOW;
```

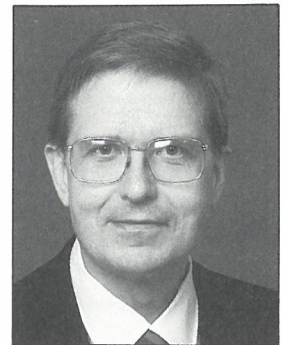
If this statement is executed, the execution of the current block is terminated, and the program control is moved to the nearest exception handler. An excep-



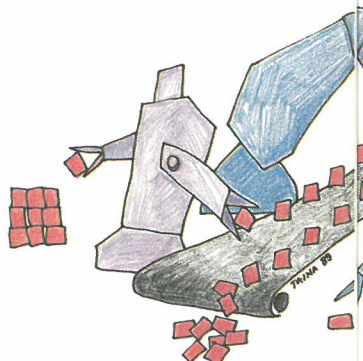
CONSTRAINT_ERROR: Violation of a range constraint, an index constraint, or a discriminant constraint. Selecting a component of, or indexing a null access value.

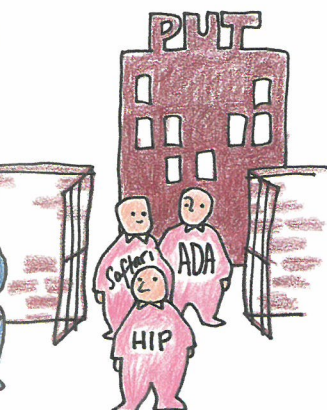
NUMERIC__ERROR: Numeric overflow, divide by zero.

STORAGE__ERROR: Storage overflow during the evaluation of an allocator, during the



Veikko Kehä is a Software Engineering Manager in Nokia Data Product Management & Development.

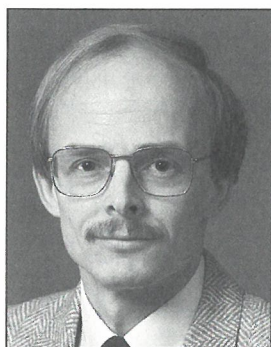




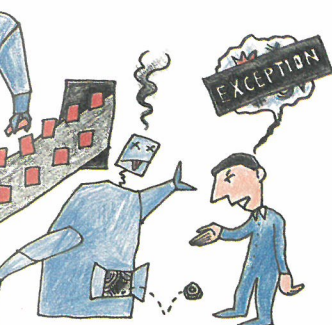
elaboration of a declarative item, or during the execution of a subprogram call.

TASKING__ERROR: Intertask communication error.

PROGRAM__ERROR: Calling of a subprogram, whose body has not been elaborated. Reaching the end of a function without encountering a return statement.



Lassi Juutinen is a Senior Software Engineering Specialist in Nokia Data Product Management & Development.



tion handler has a form similar to a case statement. A **when** clause defines the action in response to a particular exception. For example:

```
with STACK__PACKAGE, TEXT__IO;

procedure STACK_HANDLER is
  -- Declaration part
begin
  -- Statement part
exception
  when STACK__PACKAGE.OVERFLOW =
    TEXT__IO.PUT ("Stack overflow");
  when others =
    TEXT__IO.PUT ("Unknown exception");
  raise;
end STACK_HANDLER;
```

The system first checks to see if there is an exception handler at the end of the current program block (e.g. a block statement, or a subprogram body). If a handler is found, the statements associated with it are executed, and the control moves to the outer block (in the case of a block statement) or calling subprogram (in the case of a subprogram). Execution of the block that raised the exception is abandoned, and there is no way to return to the exact point in the program that raised the exception.

If no handler is found, the exception is propagated to the calling block. The search continues, and if no suitable handler is found even in the main program, the program ends with an error message.

Note: when looking for exception handlers the system moves backwards in the dynamic call chain.

The **raise** statement without parameters in the above exception handler raises the same exception again and propagates it to an upper level.

Exceptions should usually be handled at the lowest possible level. But in some cases the lowest possible level is not the best level for the purpose. If we have a stack package implementing the stack operations, it is not a good idea to handle a stack overflow in that package (e.g. by emitting an error message). Only the user of the package can decide what should be done if an overflow occurs. The package declares the exception in its visible part, and the user program handles it.

Pros and Cons of exceptions

One advantage of exceptions is that they can be used to separate error situations almost completely from the normal processing. If you want, you can write your program for correct data first and add the error handling later. This results in a program code much easier to read and maintain than with most other languages, in which status variables are returned by subprograms.

But Ada exceptions also involve problems: it is impossible to know exactly at what point a predefined exception was raised. If a **CONSTRAINT__ERROR**

is detected in a hundred-line subprogram, it is hard to locate the error and decide what should be done.

One solution is to keep program blocks small. For example, the call of the file opening procedure should be in its own block and have its own exception handler.

Problems also arise when exceptions and subprogram parameters are combined. If an exception is raised in a subprogram, the values of output parameters may be undefined after the subprogram call. An exception handler, if there is one, can assign values to output parameters and define the function return value, but the Ada compiler does not check to see if such statements exist.

Solution to the Exercise in Part 4

Here are the data-type definitions needed to implement a graph of nodes in which each node is connected to other nodes with 0..n edges. The maximum number of nodes is dynamic, so we have to use access types.

type ELEMENT__TYPE is new INTEGER; — — Say...

type NODE__TYPE;
type GRAPH__TYPE is access NODE__TYPE;

type EDGE__LIST__TYPE;
type EDGE__LIST__POINTER__TYPE is access
 EDGE__LIST__TYPE;
type EDGE__LIST__TYPE is
 record
 NEXT : EDGE__LIST__POINTER__TYPE;
 EDGE : GRAPH__TYPE;
 end record;

type NODE__TYPE is
 record
 INFO : ELEMENT__TYPE;
 EDGES : EDGE__LIST__POINTER__TYPE;
 end record;

If P1, P2 and P3 are variables of the **GRAPH__TYPE**, we can write

```
P1 := new NODE__TYPE'(11, null);
P2 := new NODE__TYPE'(22, null);
P3 := new NODE__TYPE'(33, null);
P1.EDGES := new EDGE__LIST__TYPE'(null,P2);
P1.EDGES.NEXT := new EDGE__LIST__TYPE'(null,P3);
```



The solution above is more general than required. If we want to limit the number of edges to two, it must be checked when the data structures are used in the program. ■

SQLWindows-kurssi

Softarin numerossa 1/89 alkoi neliosainen SQLWindows-kurssi. Kahdessa edellisessä numerossa olemme saaneet aikaiseksi jäsenrekisterisovelluksemme valikkorakenteen ja keskusteluikkunan uuden jäsenen tietojen syöttämistä varten. Nyt opetamme, miten keskusteluikkunaan syötetyt tiedot viedään tietokantaan sekä miten tietoja luetaan kannasta list box -rakenteeseen.

Jos et ole käyttänyt SQLWindowsia artikkelisarjan edellisen osan jälkeen, olisi ehkä syytä silmäillä aikaisemmat osat läpi. Sen lisäksi kannattanee sulkea Outline-ikkunassa oleva SAL-kielinen ohjelma niin, että siitä näkyy vain ylin otsake: Application Description... Tämän jälkeen yritä etsiä Outlinen otsikoita avaamalla kaikki muutokset, joita olet itse sinne tehnyt. Normaalisti ne löytyisivät helposti merkitsemällä ne Pääikkunasta painamalla hiiren vasenta näppäintä kyseisten olioiden kohdalla tai Search-valikon avulla. Etsiessäsi niiden koodia Outlinesta, tulee samalla SAL-kielisen ohjelman perusrakenne kerratuksi.

Tarkista samalla, mitä SAL-kielisessä vakiorakenteisessa ohjelmassa on määritelty jo valmiiksi. Etsi esimerkiksi viime kerralla käyttämämme SalMessageBox-funktion määrittely, joka löytyy kaikkien muiden valmiiden määrittelyjen tavoin kohdasta Global Declarations. Tarkastele erityisesti sen alaotsakkeita "Returns" ja "Parameters". Parameters-kohdassa on lueteltu järjestyksessä parametrien tyypit muodossa Number: WORD, joista ensimmäinen kertoo SQLWindowsissa käytettävän tyyppin ja toinen Windows SDK:ssa C-kielillä ohjelmoitaessa käytetyn tyyppin. Tämän kurssin aikana olemme kiinnostuneita vain ensin mainitusta.

Sulje Global Declarations -otsake ja tutki myös, mistä löytyvät

MB-Ok-vakion sekä hWndForm-muuttujan määrittelyt. Sulje välillä turhia otsakkeita, jotta varmasti tiedät, missä kohtaa sovelusta olet.

Viime kerralla saimme aikaiseksi valikon "Jäsenistö", jonka vaihtoehdosta "Uusi jäsen..." käynnistyi kuvan 1 mukainen keskusteluikkuna. Jos kyseinen Dialog Box ei ole näkyvässä, se löytyy merkitsemällä rivi Dialog Box... ja sitten View-valikosta valitsemalla vaihtoehto Show Window.

Koska uuden jäsenen vasta syötetyt tiedot muuten häipyisivät taivaan tuuliin, lisäämme koodia, jonka avulla tiedot tallennetaan tietokantaan. Samalla selvitämme muuttujien määrittelyt, Wintalk-apuohjelman ja SQL-lauseiden upottamisen SAL-kieliseen ohjelmaan. Tämän jälkeen piirämme vielä list boxin ja lisäämme koodia, jonka avulla voimme lukea tietokannasta kaikkien jäsenten nimet list boxiin selattaviksi.

Kenttien käsittely

Kenttiä voi käyttää muuttujien tavoin kentän nimen avulla. Kentän nimi löytyy Outlinesta kenttäoli- on ensimmäiseltä riviltä, esimerkiksi Data Field: nro. Kentällä on muuttujaan nähden eräs ero: kun kentälle annetaan arvo (esim. Set nro = 1), sen arvo näkyy näytöllä.

Data Field -kenttiin voi käyttää ja syöttää uuden arvon. Avaa

vaikkapa kaikki nro-kenttään liittyvät rivit Outlinesta. Sen voi tehdä merkitsemällä Data Field: nro -rivin ja valitsemalla sen jälkeen Outline-valikosta "Expand all levels" -vaihtoehdon.

Data Field -kenttään syötetyn tiedon tyyppi voidaan määritellä otsakkeessa "Data" olevaan kohtaan "Data Type". Järjestelmä tarkistaa automaattisesti kuitenkin vain syötteen maksimipituisuuden ("Maximum Data Length"), muut tarkistukset joutuu ohjelmoimaan itse. Tarkistuksia varten Data Fieldille voi määritellä tyyppin lisäksi Display Settings -kohtaan myös käytettävän formaatin (Format) ja maan asetukset (Country, profiili). Esimerkiksi rahasummalle voi määritellä tyyppiä Number, formaatiksi Currency ja maaksi (Country) Finland, jolloin syötteeksi kelpaisivat sekä "123" että "123 MK".

Kokeillaan esimerkiksi eri formaattivaihtoehtoja. Ne saadaan kätevästi esiin painamalla Format-rivin alussa hiiren oikeaa näppäintä ja sen jälkeen näppäinmistöstä nuoli ylös -näppäintä.

Tämäntapaiset tarkistukset joutuu siis ohjelmoimaan itse. Tällöin koodataan Data Fieldin Message Actions -kohtaan reagentti esimerkiksi SAM-AnyEdit -sanomaa (= jokaisen kenttään kohdistuneen muutoksen jälkeen), jolloin funktion SalFmtIsValidField avulla tarkistetaan, onko syöte kentälle asetetun formaatin ja maan tavan mukainen. Kenttien tarkistuksesta tulee toimituksen mukana hyvä esimerkkiohjelma FMTDEMO.APP.

Radio Button ja Check Box -kentät saavat Boolean- eli totuusarvon (TRUE tai FALSE) sen mukaan, ovatko ne "päällä" vai "pois päältä". Niihin voidaan myös viitata kenttien nimillä, kuten Boolean-muuttujiin.

Background Text ja Group Box -kenttiin ei sen sijaan voi viitata muualta, sillä niissä on nimien sijaan Title-tyyppisiä, suoraan näytöllä näkyviä tekstejä.

Koska tietokannassamme ei ole Boolean-kenttiä, avuksi tarvitaan muuttujia, joiden avulla tiedot saadaan talteen. (kuva 2.)

Kuva 1: Keskusteluikkuna uuden jäsenen tietojen syöttämistä varten.

Osa 3 — Tietokannan käsittely

Muuttujien määrittely

Muuttujat voidaan määritellä otsikon "Variables" alle. Näitä otsikkoja ohjelmassamme on useitakin. Valitaan niistä jokin ja otetaan View-valikosta Outline Options näkyviin. Se näyttää kaikki mahdolliset muuttujatyypit, kun Variables-otsikko on merkittynä. Kelataan Outline Optionsin List boxia alas päin, jotta kaikki vaihtoehdot saadaan näkyviin. Valitaan sieltä kokeeksi esimerkiksi Number-tyyppi, jolloin saadaan valmis muuttujamäärittelyrivin alku, joka voidaan kirjoittaa itse loppuun:

Number: nKoe

Tällä tavalla alkavat kaikki muuttujamäärittelyt: ensin tyyppin nimi, sitten kaksoispiste ja muuttujan nimi. Muuttujan nimeämisessä kannattaa käyttää nimen alkuosaa, joka ilmaisee muuttujan tyyppin.

String-tyypeille muuttujille pitää vielä antaa pituus, joka kirjoitetaan muuttujan nimen perään sulkeisiin, esimerkiksi

String: strOsoite (30)

Tosin pituuden voi korvata tähdellä, jolloin tila varautuu dynaamisesti tarpeen mukaan.

Yksilotteisia taulukoita voi määritellä lisäämällä hakasulkeissa taulukon alkioiden lukumäärän nimen perään:

Number: nSaldot [100]

Myös merkkijonotaulukot ja dynaamiset taulukot ovat sallittuja.

Muuttujat määritellään mahdollisimman lähelle käyttöpaikkaansa, esimerkiksi tässä tapauksessa Dialog Boxin Variables-kohtaan. SAL-kielessä, toisin kuin useimmissa muissa kielissä, muuttujat määritellään usein vasta niiden käyttöpaikan jälkeen. SAL-kielessä sisennys kuitenkin ilmaisee, että ne ovat nimenomaan Dialog Boxin muuttujia.

Jos määritellään globaaleja muuttujia (Global Declarations: Variables), niiden nimet kannattaa aloittaa vaikka g-kirjaimella, jotteivät ne mene sekaisin minkään ikkunan omien muuttujien kanssa.

Katsotaan seuraavaksi, minkä muotoisena keskusteluikkunaan syötetyt tiedot voidaan viedä tietokantaan.

Tietokanta hallintaan

Toimituksen mukana tulee WINTALK.EXE, joka vastaa SQL-TALK.EXE:ä, mutta on Windows-ohjelmana kätevämpi käyttää. Luomme sen avulla tietokantaan jäsenrekisterisovelluksellemme sopivan taulun. Poistu SQL-Windowsista ja käynnistä siis Windowsista WINTALK.EXE.

Ensimmäisenä tehtävänä on ottaa yhteys tietokantaan eli valikosta kohta Session ja siitä Connect. Sen jälkeen valitaan DEMO-tietokanta, johon luodaan taulu JASENET. Kun yhteys

on otettu, Input Window -ikkunaan voidaan kirjoittaa SQL-lauseita ja suorittaa niitä Run-valikon avulla, jolloin tulokset tulevat Output Window -ikkunaan. Output Window:n alla on yksi tilanne-rivi, johon punaisella kirjoitetut virheilmoitukset tulevat.

Kirjoita nyt seuraava SQL-lause, jolla luodaan DEMO-tietokantaan taulu JASENET sarakkeineen. Lause kirjoitetaan yhdelle riville, vaikka se onkin ohessa jouduttu jakamaan peräti kuudelle riville.

```
create table JASENET (  
  NRO integer,  
  NIMI char(30),  
  KULJIN char(1),  
  VARI1 char(1),  
  VARI2 char(1));
```

SQL-lauseet päätetään Wintalkissa aina puolipisteellä. Nyt voit suorittaa lauseen valitsemalla Run-valikosta Current Command tai pikavalinnalla SHIFT-F2, jolloin järjestelmä suorittaa sen SQL-lauseen, jolla kohdistin on.

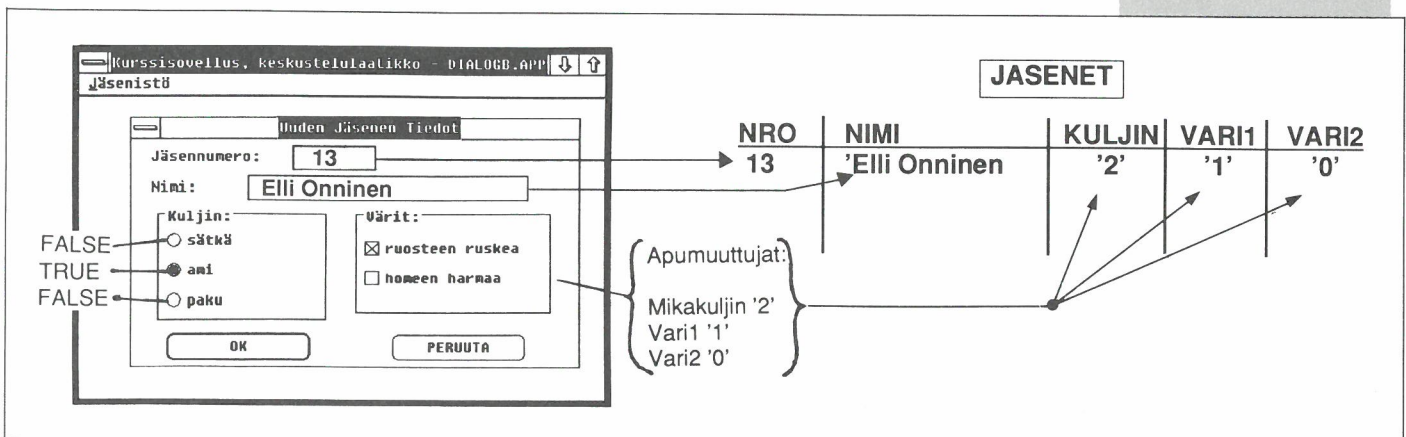
Voit tarkistaa JASENET-taulun sarakkeiden pituudet, tyypit jne. käyttämällä seuraavanmuotoista yhdelle riville kirjoitettavaa SELECT-lauseetta:

```
select * from SYSCOLUMNS  
where TBNAME='JASENET';
```

Suorita se taas viemällä kohdistin lauseen kohdalle ja painamalla SHIFT-F2.

Seuraavaksi voit kokeilla viedä joitakin rivejä JASENET-tauluun

Kuva 2: Esimerkki tietojen tallentamismuodosta.



► SQLWindows

seuraavanmuotoisella yhdelle riville mahtuvalla INSERT-lauseella:

```
insert into JASENET  
values(1,'joku nimi','1','0','1');
```

ja tarkistaa tietojen perille meno SELECT-lauseella:

```
select * from JASENET;
```

Huomaa kuitenkin, että skandimerkit eivät täsmää SQLWindows-sovelluksen tuottamien skandien kanssa.

Kun taulu on saatu kuntoon, voit poistua WINTALKista. Poistuttaessa järjestelmä kysyy, tallennetaanko Input Window -ikkunaan kirjoittamasi "script" tiedostoon, jolloin voisit käyttää sitä toistekin (File: Open...). Lisäksi kytketään pois yhteys DEMO-tietokannasta (Disconnect).

Wintalk on kätevä apuneuvo paitsi järjestelmän tukihenkilölle, myös SQLWindows-sovelluskäyttäjälle. Sen avulla voi testata ohjelmassa tarvittavat SQL-lauseet ja sitten siirtää ne suoraan Edit-valikon Copy-komennolla talteen ja noukkia ne Edit-valikon Paste-valinnalla SAL-kieliseen ohjelmaan. Tosin SAL-ohjelmasta on valittava juuri oikeanlainen rivi Pasten kohteeksi, ennenkuin järjestelmä sallii toimenpiteen.

Tiedot kantaan

Seuraavana tehtävänä on keskusteluikkunaan syötettyjen tietojen vieminen tietokantaan. Yksinkertaistetaan tehtävää vielä sillä tavoin, että jätetään kenttien tarkistukset tekemättä. Numero- ja nimi-kenttien tiedot pystytään viemään sellaisinaan, mutta Kuljin ja Väri-kentät vaativat erikoiskäsittelyn, sillä ne viedään kantaan merkkimuotoisina kuvan 2 mukaisesti.

Kuljinvaihtoehdot on koodattu seuraavasti: sätkä = '1', ami = '2' ja paku = '3'. Väri:n (ruosteen ruskea) arvoksi tulee '1', jos ko. väriä löytyy kulkimesta, ja '0' muuten. Väri2 on koodattu samoin.

Jatketaan ohjelman muokkaamista. Wintalk kannattaa lopettaa, sillä SQLWindowsilta saattaa muuten loppua muisti. Tässä vai-

Application Description: SQLWindows-kurssin sovellus

Design-time Settings

Global Declarations

Form Window: MainForm

Dialog Box: AboutBox

Dialog Box: UusiJasen

Title: Uuden Jäsenen Tiedot

Display Settings

Contents

Background Text: Jäsennumero:

...

Pushbutton: OK

Title: OK

Window Location and Size

Keyboard Accelerator: Enter

Message Actions

On SAM_Click

If satka=TRUE

Set MikaKuljin='1'

If ami=TRUE

Set MikaKuljin='2'

If paku=TRUE

Set MikaKuljin='3'

If ruoste=TRUE

Set Vari1='1'

Else

Set Vari1='0'

If home=TRUE

Set Vari2='1'

Else

Set Vari2='0'

Call SqlImmedate(INSERT INTO JASENET
(NRO,NIMI,KULJIN,VARI1,VARI2 VALUES
(:nro, :nimi, :MikaKuljin, :Vari1, :Vari2))

Call SalEndDialog(hWndForm,1)

Pushbutton: Cancel

Title: PERUUTA

Window Location and Size

Keyboard Accelerator: (none)

Message Actions

On SAM_Click

Call SalMessageBox('Enpä välitä antamistasi
tiedoista!', 'Pyysit peruuttamaan, siis: MB Ok)

Call SalEndDialog(hWndForm,2)

Variables

String: MikaKuljin(1)

String: Vari1(1)

String: Vari2(1)

Message Actions

Kuva 3: Tietojen tallettamiseen tarvittava koodi.

heessa ohjelma kannattaa tallentaa tekstimuotoon (File: Save As, laajennusosaksi .APT) ja lukea se uudelleen tekstimuotoisena käsiteltäväksi (File: Open). Tämä toimenpide selvittää ohjelman rakenteen, eivätkä turhat järjestelmän kaatumiset pääse häiritsemään. Tee näin aika ajoin ja poistu välillä SQLWindowsista, niin päiväsi tulevat olemaan onnellisia!

Kuvitellaan, että käyttäjä haluaa tietojen menevän tietokantaan vasta, kun hän painaa OK-painonappia. Toisin sanoen voimme kirjoittaa kaiken koodin sen toimintakohtaan. Muitakin ratkaisuja olisi, mutta tämä lieene helpoin. Siis ennen kuin poistumme keskusteluikkunasta, tarkistamme, mitkä ovat kenttien sisällöt, asetamme apumuuttujien arvot ja viemme tiedot tietokantaan.

Koodirivejä pitää siis lisätä On-lauseen ja Call SalEndDialog -kutsun väliin, mikä osoittautuu oletettua hankalammaksi tehtäväksi. Kun View-valikosta ottaa Outline Options -ikkunan ja valitsee sieltä If-lauseen On-lauseen ollessa merkittynä, ilmestyykin uusi rivi Call-lauseen perään eikä edelle. Onneksi Alt-näppäimen ja nuolinäppäimien avulla merkittyjä rivejä voidaan aina siirtää ylös-, alas tai sivuillepäin. Merkitse siis jälkeensä Call-lause ja siirrä se Alt-näppäimen ja nuoli alas -näppäimen avulla oikealle paikalleen.

Radio Button ja Check Box -kenttien arvojen tarkistukseen tarvitaan ensinnäkin String-tyypiset apumuuttujat (MikaKuljin, Vari1 ja Vari2), jotka voivat kaikki olla yhden merkin mittaisia, kuten kannassa.

Set-lause löytyy Outline Optionsin List boxista kelaamalla sitä alaspäin. Set-lauseen muoto on

Set muuttuja = uusi arvo

jossa muuttujan tilalla voi olla kenttäkin.

Lisäksi tarvitaan If-lause muodossa:

If ehto

! then-osa sisennettynä

Else If ehto

! jälk. If:n then-osa sis.

Else

! else-osa sisennettynä

jossa ehto voi olla esimerkiksi seuraavan näköinen

```
fldOsoite = 'HELSINKI'
```

Ensimmäistä If-lauseetta vastaava Else If alkaa samalta sarakkeelta kuin If. Outline Optionsia käytettäessä merkitään If ja valitaan Else vasemmanpuoleisesta List boxista.

If-lauseetta rakentaessaan kannattaa olla tarkkana, ettei vahingossa jää yksinäistä Else-osaa ilman vastaavaa If:iä. Guptan Release Noteissa varoitetaan, että tällainen tilanne saattaa sotkea jopa ohjelmätiedostot. Tallenna siis tiedostosi välillä tekstimuotoon eri nimelle ja poistu SQLWindowsista ja mene kahville. Palatessasi käynnistä SQLWindows käyttäen tätä juuri tallentamaasi, tekstimuotoista tiedostoa. Huomaa If-lauseesta vielä, että If-rivillä saa olla vain ehtolause ja Else-sanalla perässä ei saa olla samalla rivillä mitään muuta.

Upotus SAL-kieleen

Tietokantaan vieminen tapahtuu tässä vaiheessa helpoiten funktion SqlImmediate avulla. Se ottaa vastaan vain yhden parametrin eli suoritettavan SQL-lauseen merkkijonona (= heittomerkit ympärille). Muilta osin se käyttää oletusarvoja (tietokantana DEMO, käyttäjänä SYSADM ja salasana SYSADM). Näitä oletusarvoja voi muuttaa määrittelemällä kohtaan Global Declarations: Variables Sql-alkuiset muuttujat, jotka ovat siellä valmiina kommenttimerkin takana. Näiden muuttujien arvojen kyselemiseen ja asettamiseen on useissa esimerkkiohjelmissa valmiita koodinpätkiä. Tällä kertaa annamme näiden muuttujien olla määrittelemättä.

Etsi nyt kohta, johon aiot kirjoittaa SqlImmediaten kutsun. Kutsurivin saa oikeaan kohtaan merkitsemällä Else-rivin ja valitsemalla Call-lauseen Outline Optionsin vasemmasta List boxista.

SqlImmediaten parametriksi merkkijonona kirjoitettava SQL-lause on muuten standardin mukainen, mutta ohjelmassa määriteltyjen muuttujien ja kenttien nimien eteen kirjoitetaan kaksoispiste. Seuraavassa esimerkki tar-

vittavasta INSERT-lauseesta (yksi rivi):

```
insert into JASENET
(NRO, NIMI, KULJIN,
VARI1, VARI2)
values (:nro, :nimi, ...)
```

Riviä voi jatkaa painamalla rivin lopussa Ctrl-Enter. Huomaa, ettei rivin perään tule puolipistettä, vaikka Sqltalkissa ja Wintalkissa se tarvitaan.

Jos et heti muista, miten esimerkiksi kenttiin viitattiin, lukaise kertaalleen edellä olevat kohdat, joissa kerrotaan kentistä ja muuttujista. Jos ei siitäkään ole apua, kurkista valmista ratkaisua kuvasta 3.

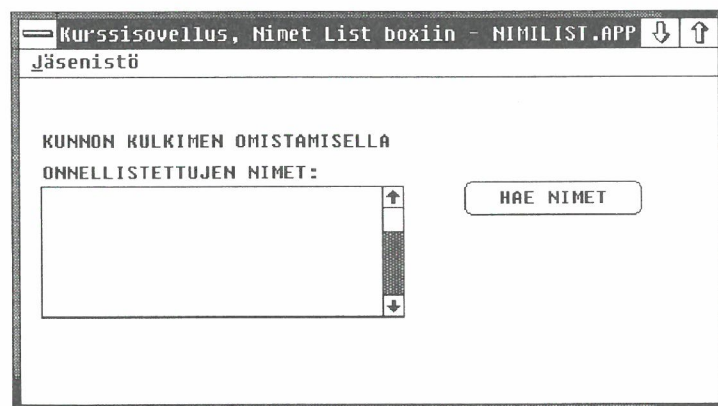
Tehtyäsi ohjelmaan edellä esitetty muutokset, voit taas kokeilla ajaa sitä asettamalla ensin animaation päälle Run-valikon Slow Animate -vaihtoehdolla. Sen jäl-

tä tarkistaa Sql-funktioiden palauttama arvo ja puuttua virhetilanteisiin ohjelmallisesti.

Tietojen tietokantaan menemisen voi tarkistaa Wintalkilla edellä esitettyyn tapaan. Poistu kuitenkin ensin SQLWindowsista. Sitten voit lisätä PERUUTA-painonappiin toiminnaksi SalMessageBox-funktion kutsun, jolla ilmoitat, ettei tietoja viedä tietokantaan, koska käyttäjä ei sitä kerran halua.

List box -merkkijonot

Seuraavaksi piirrämme kuvan 4 mukaisen list boxin. List boxia voi käyttää tulostusalustana esimerkiksi ohjelman tapaan tai selaukseen sekä myös valintaan. Jos piirtäminen ei onnistu, on syynä yleensä se, että SQLWindows on User-tilassa. Ottamalla User Mo-



Kuva 4: Painonappin "HAE NIMET" avulla luetaan tiedot tietokannasta List boxiin.

keen käynnistyy käännös ja suoritetaan Run: User mode:lla. Aina ennen käännöstä kaikki turhat ikkunat kannattaa sulkea tai ikonisoida muistitilan vapauttamiseksi.

Käännösvaiheessa ei vielä tule virheilmoituksia SQL-lauseista, vaan SQLBase tarkistaa lauseen vasta, kun kyseinen SQL-funktio suoritetaan. Yritä siis valita Jäsenistö-valikosta Uusi jäsen -vaihtoehto, syöttää tietoja ja painaa OK-painonappia. Ohjelma luultavasti toimii moitteettomasti, jos se ei anna virheilmoituksia. Näinhän optimistiohjelmoijat aina luulevat.

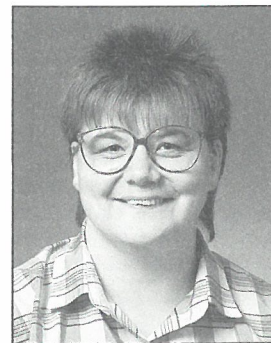
Jos SQL-virheilmoituksia kuitenkin tulee, niissä mainittu Position tarkoittaa, monennenko merkin kohdalla SQL-lauseen alusta lukien virhe havaittiin. Todellises-

sa sovelluksessa on tietenkin syy-

den Run-valikosta pois päältä SQLWindowsin saa Design-tilaan, jonka merkinä Pääikkunaan ilmestyy taustapisteytys.

Nyt on tarkoitus piirtää Pääikkunaan eli siihen ikkunaan, joka tulee ensimmäiseksi käyttäjälle näkyviin. Jos tämä Form Window on kadoksissa, merkitään Outlinesta sen ensimmäinen rivi (Form Window:...) ja pyydetään ikkuna näkyviin View: Show Window.

Piirrämme myöskin painonappin "HAE NIMET", sillä valikon kohta "Jäsenluettelo" säästetään myöhempiä tehtäviä varten. Muistathan, että piirrettäessä jotakin ikkunan (tässä tapauksessa Pääikkunan) Contents-kohtaan työ aloitetaan valitsemalla Outlinen valikosta Draw piirrettävän olion tyyppi. Sen jälkeen hiirikohdistin viedään ikkunan halut-



Tiina Mikonmäki vaikuttaa Mikonmäki Oy:ssä p. (915) 49 388. Hän toimii verailevana kouluttajana mm. Nitecissä aiheinaan C-kieli, SQLWindows ja OS/2 Presentation Manager.

tuun kohtaan ja painetaan hiiren vasen näppäin alas. Kun olio on saatu sopivan kokoiseksi, päästetään hiiren näppäin ylös. Painonappia piirrettäessä kannattaa tässä vaiheessa kirjoittaa sen näytöllä näkyvä teksti, sillä teksti näkyy merkittynä ja järjestelmä on editointitilassa. Kirjoitettua tekstiä paina Enteriä niin pääset nuoltilaan.

Kun olet saanut Pääikkunan kuvan 4 mukaiseksi, on aika lisätä koodia, jotta saamme oliot henkiin. Ota taas avuksesi View-valikosta Outline Options.

List box -funktioissa on vielä eräs puute. Useimmiten, kun haluamme viitata johonkin olioon, voimme käyttää sen Template-nimeä. Template on se nimi, joka esiintyy ohjelmakoodissa olion ensimmäisellä rivillä, esimerkiksi List Box: lbNimet. List boxeihin ei kuitenkaan voi viitata tällä nimellä, vaan tarvitaan aina List boxin kahva.

Jokaisella oliolla on oma, yksilöllinen kahvansa (kuten autoilla rekisterinumerot), jonka avulla Windows tunnistaa sen. Itse asiassa SAL-kielisen ohjelman vakiorungossa on valmiiksi määritellyt kaksi muuttujaa tähän tarkoitukseen, hWndForm ja hWndItem. Etsi niiden määrittelyt!

Näiden muuttujien avulla Windows ilmoittaa sovellukselle, missäpäin käyttäjä on menossa. Sovelluksessa ei siis anneta niille uusia arvoja Set-lauseella, vaan pelkästään käytetään niitä, sillä Windows asettaa niille arvot. hWndForm kertoo kullakin hetkellä sen korkeimman tason ikkunan (Form Window = Dialog Box) kahvan, joka on aktiivinen ja hWndItem kertoo, missä tämän aktiivisen ikkunan kentässä käyttäjä parhaillaan työskentelee.

Koska List box -funktioissa tarvitaan aina List boxin kahva, on se otettava talteen omaan Window Handle -tyyppiseen muuttujaan hWndItem-muuttujasta, kun List box on aktiivinen. Oliohan on aktiivinen, kun reagoidaan johonkin olion saamaan sanomaan. Jokainen olio saa ainakin sanoman SAM-Create juuri, kun oliota ollaan piirtämässä näytölle. Niinpä jokaisella List boxilla on vähintäinkin seuraavanlainen koodi toiminta-kohdassa:

```
On SAM-Create
Set hWndNimet = hWndItem
```

List boxin toimintakohdan löytää helpoiten merkitsemällä List boxin Pääikkunassa. Muuttuja hWndNimet pitää tietysti määritellä mahdollisimman lähelle käyttöpaikkaansa Window Handle -tyyppiseksi.

Nyt on aika kirjoittaa koodia, jotta saamme ohjelmamme toimimaan seuraavasti:

Kun käyttäjä napauttaa painonappia HAE NIMET, sen toimintana on hakea taulusta JASENET kaikkien jäsenten nimet ja näyttää ne List boxissa. Painonapin toiminnan saa soveltamalla otaksien "Sql-funktioiden käyttö" sekä "Tietojen lukeminen" alla olevia esimerkkejä.

SQL-funktioiden käyttö

Seuraavaksi tarkoituksena on lukea tietokannan JASENET-aulusta kaikkien jäsenten nimet ja tulostaa ne list boxiin selailtaviksi. Ensi kerralla luemme taulusta jäsenten kaikki tiedot table window -rakenteeseen, mutta tällä kertaa tyydymme pelkkiin nimiin, sillä list boxin rivit ovat yksittäisiä merkkijonoja.

Tietokannasta lukeminen onkin hieman monimutkaisempaa, varsinkin jos lukemisen tuloksena on monta riviä, niinkuin tässä tapauksessa. Kuvassa 5 on kaavio, jossa kuvataan eri vaihtoehtoja tietokannan käsittelyyn sopiviksi funktioiksi.

Kuvasta 5 voi myös havaita, että aikaisemmin käyttämämme SqlImmediate teki sisäisesti Connectin. Toisin sanoen suorituksen nopeuden kannalta pitäisi sen sijaan käyttää erillisiä funktioita SqlConnect, SqlPrepare ja SqlExecute, jolloin hidasta Connectia ei tarvittaisi kuin kerran ja toistuvat asiat voitaisiin tehdä SqlPrepare ja SqlExecute-pareilla.

SqlConnect ottaa yhteyden tietokantaan ja palauttaa parametrina annetun muuttujan arvoksi tietokannan kahvan, joka sitten välitetään seuraavissa funktioissa ensimmäisenä parametrina kertomaan, mistä tietokannasta ja käyttäjistä on kysymys. Sovelluksella voi olla vain viisi Connectia voimassa yht'aikaa, joten turhaan niitä ei kannata tehdä. Ehkä paras paikka SqlConnect-funktioille olisikin Global Declarations: Application Actions -kohdassa reagoitaessa sanomaan SAM-AppStartup ja SqlDisconnect sanoman SAM-AppExit tullessa. Esimerkiksi

```
Call SqlConnect(hWnd)
```

Muuttuja hWnd on määriteltävä ja sen tyyppinä on Sql Handle. Kyseisen muuttujan nimenä voi olla joku muukin, mutta sen nimi kannattaa aloittaa merkillä h = handle ja lyhenteellä Sql = tietokantaan liittyvä eli tekstillä hWndSql = tietokannan kahva.

Tämän jälkeen tarvitaan vielä SqlPrepare, jolle annetaan para-

metrina SqlConnectissa käytetty kahva ja toisena parametrina käännettävä SQL-lause merkkijonona. SqlPrepare tarkistaa SQL-lauseen, optimoi ja kääntää sen. Esimerkiksi seuraava rivi (yksin sellainen):

```
Call SqlPrepare(hWnd,
'INSERT INTO ASIAKKAAT ...')
```

Seuraavaksi SqlExecute ottaa parametrina vastaan vain SqlConnectissa ja SqlPreparessa käytetyn kahvan ja suorittaa SqlPreparen valmisteleman SQL-lauseen. Esimerkiksi

```
Call SqlExecute(hWnd)
```

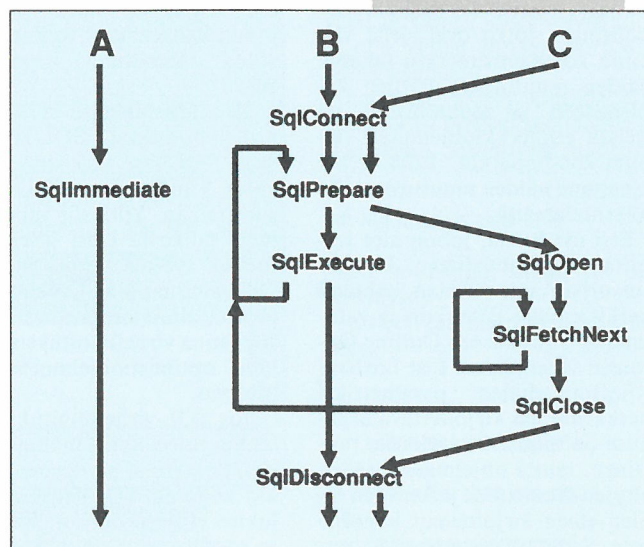
Jos kyselyn tuloksena on monta riviä, ei SqlImmediate-funktiota eikä SqlConnect-SqlPrepare-SqlExecute-SqlDisconnect -funktioita voida käyttää. Silloin tarvitaan SqlExecuten sijaan työtilan avaus, haku rivi kerrallaan sekä työtilan sulkeminen.

Tietojen lukeminen

Kuvan 5 sarakkeessa C on esitetty, mitä funktioita tarvitaan, kun tietokannasta tehdään kysely (SELECT), joka tuottaa tuloksena useita rivejä. Nyt esimerkiksi aiomme lukea tietokannan taulusta JASENET kaikkien jäsenten nimet. Tässä tapauksessa tarvitsemme SqlConnectin ja SqlPreparen lisäksi funktioita SqlOpen, SqlFetchNext ja SqlClose. Etsi näiden funktioiden määrittelyt, avaa ne kokonaan ja vertaa seuraavia esimerkkikutsuja parametrin määrittelyihin.

Näistä SqlPrepare kääntää SELECT-lauseen, jonka INTO-osassa kerrotaan, minkä muuttujien tai kenttien arvoksi aina yhden rivin sarakkeiden sisällöt tullaan lukemaan, esimerkiksi

Kuva 5: Tietokannan käsittelyyn tarvittavat funktiot.




```
Call SqlPrepare(hSql,
'SELECT ASNRO, ASNIMI FROM
ASIAKKAAT INTO
:nAsNro, :strAsNimi')
```

Seuraavaksi avataan työtila (kursori) ja suoritetaan kysely SqlOpen-funktiolla. Verkkoversiossa työtila varautuu palvelinkoneelta, ja funktion vaikutuksesta kyselyn tuloksena olevat rivit si-joitetaan sinne. SqlOpen-funktio liittyy aina samalle kahvalle aikai-
semmin tehtyyn SqlPrepareen. SqlOpen-funktion toisena para-
metrina annetaan työtilalle eli kursorille jokin nimi, joka on va-
paasti valittavissa oleva merkki-
jono. Sitä voitaisiin käyttää myö-
hemmin SELECT-lauseen WHE-
RE CURRENT OF -osassa. Esi-
merkiksi

```
Call SqlOpen(hSql, 'NROTJANIMET')
```

Nyt vastausrivit ovat valmiina palvelinkoneen työtilassa ja seu-
raavaksi niitä aletaan lukea ohjel-
mamuuttujien arvoiksi yksi rivi
kerrallaan. Siihen tarvitaan sil-
mukkaa ja SqlFetchNext-funk-
tiota:

```
Loop
If SqlFetchNext(hSql, nInd) =
TRUE
! nyt 1 vastausrivin
! tiedot ovat
! muuttujien nAsNro
! ja strAsNimi
! arvoina, ja tähän lisättäisiin
! koodia niiden
! käsittelyä
Else
! rivit loppuivat, joten
! poistetaan silmukasta
Break
```

Silmukka loppuu, kun sisen-
tyt rivit loppuvat. Silmukasta
poistutaan Break-lauseella, kun
SqlFetchNext-funktio palauttaa
FALSE-arvon ilmaisten, että
kaikki rivit on jo luettu työtilasta.

SqlFetchNextin toisena para-
metrina on annettava muuttuja,
sillä funktion määrittelyssä kysei-
sen parametrin tyyppinä on Re-
ceive Number. Sana "Receive"
muuttujan tyyppin edessä tarkoit-
taa, että kyseisenä parametrina
annetun muuttujan arvo muut-
tuu funktion suorituksen aikana.
Yleensä tällaisessa muuttujassa
saadaan funktiolta jotain tärkeää
paluutietoa SqlConnect-funktion
antaman tietokannan kahvan ta-
paan. SqlFetchNext-funktiossa
toisen parametrin arvo on kiin-
nostava vain, jos tehdään monen
käyttäjän järjestelmää ja määri-
tellään käyttäjien eristystasoksi
Cursor Stability. Tällöin nInd-
muuttujan paluuarvot 2 ja 3 il-
maisevat, onko joku muu käyttä-
jä jo ehtinyt päivittää tai poistaa

tietoa sen jälkeen, kun vastausri-
vit on saatu työtilaan.

Kun kaikki rivit on luettu työti-
lasta, käytetään vielä funktiota
SqlClose vapauttamaan työtila
esimerkiksi seuraavaa SqlOpenia
varten:

```
Call SqlClose(hSql)
```

Tämän jälkeen tarvitaan vielä
SqlDisconnect, jollei kyseistä
kahvaa enää tarvita seuraavaa
SqlPreparea varten.

Kun rivi on silmukassa luettu
tietokannasta muuttujaan (esim.
strNimi, pituus 15 merkkiä), se
näytetään List boxissa SalList-
Insert-funktiolla. SalListInsert-
funktion ensimmäisenä paramet-
rina annetaan List boxin kahva,
seuraavana monenneksiko riviksi
tämä rivi vietään (—I=loppuun)
ja viimeisenä parametrina joko
vakiomerkkijono tai merkkijo-
nomuuttuja, joka sisältää List
boxiin vietävän rivin.

Kun olet saanut koodin kirjoj-
tetuksi, testaa taas sen toimivuut-
ta animaation avulla. Jos saat
jonkin olion kohdalla virheilmoi-
tuksen "Illegal symbol", on si-
nulta unohtunut olion Template-
nimi.

SQLWindows hakee tietokan-
nasta kaikki nimet, vaikkei näy-
töllä näkyisi kuin muutama.
Tarkkaile Windowsin toimintaa
myös siinä tilanteessa, että käyt-
täjä selailee List boxin rivejä.

Jos koodauksessa tulee ongel-
mia, voit kurkistaa kuvassa 6 ole-
vaa toimivaa ratkaisua.

Entä sitten?

Tässä vaiheessa osaat jo luulta-
vasti muokata valikon, saada kes-
kusteluikkunan toimimaan, vie-
dä tietoja tietokantaan, lukea ni-
tä sieltä ja näyttää rivejä List bo-
xissa. SQLWindowsin oliokäsite
alkanee olla hahmottunut.

Jos kiinnostusta riittää, voit
kokeilla muitakin List box -funk-
tioita. Esimerkiksi lisätä yhden
List boxin, jonka avulla käyttäjä
voi tehdä valinnan (esim. vain pa-
kujen omistajien nimet halutaan
nähdä). Siihen tarvitset funktiota
SalListQuerySelection. Onnea
matkaan!

Seuraavalla, viimeisellä kerral-
la uppoudumme vielä mielenkiin-
toiseen, SQLWindowsin omaan
olioon nimeltä Table Window.
Sen avulla saamme näytölle tieto-
kannan tauluja sarakkeineen,
pystymme editoimaan niitä ja
päivittämään tietokantaa. ■

Kuva 6: Nimien lukemiseen tar-
vittava koodi.

Application Description: SQLWindows-kurssin sovellus

Design-time Settings

Global Declarations

External Functions

Constants

Variables

Window Handle: hWndNULL

Window Handle: hWndForm

Window Handle: hWndItem

Number: wParam

Number: lParam

! Database name: default = DEMO

! String: SqlDatabase(8)

! Authorization name: default = SYSADM

! String SqlUser(8)

! Password: default = SYSADM

! String: SqlPassword(18)

Sql Handle: hSQL

Internal Functions

Application Actions

On SAM_AppStartup

Call SqlConnect(hSQL)

On SAM_AppExit

Call SqlDisconnect(hSQL)

Form Window: MainForm

Title: Kurssisovellus, Nimet List Boxiin - NIMILIST.APP

Icon File: (none)

Display Settings

Menu

Contents

Background Text: LAATUKULKIMEN OMIMISELLA

Background Text: ONNELLISTETTUIJEN NIMET

List Box: NimiLista

Window Location and Size

Multiple selection? No

Sorted? Yes

Vertical Scroll? Yes

Message Actions

On SAM_Create

Set hWndList=hWndItem

Pushbutton: pHaenimet

Title: HAE NIMET

Window Location and Size

Keyboard Accelerator: (none)

Message Actions

On SAM_Click

Call SQLClose(hSQL)

Call SqlPrepare(hSQL,'select

nimi into :strNimi from jaset')

Call SqlOpen(hSQL,'JASENQRY')

Call SalListClear(hWndList)

Loop

If SqlFetchNext(hSQL, nFetch) = FALSE

Break

Call SalListInsert(hWndList, -1, strNimi)

Call SalListSetSelect(hWndList,0)

Variables

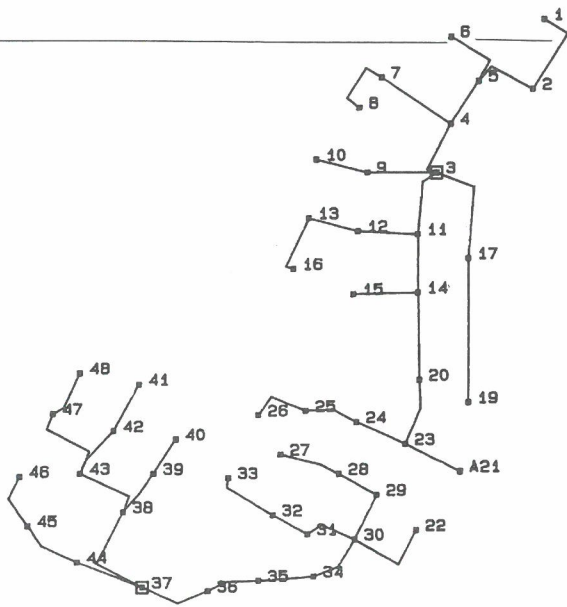
Window Handle: hWndList

String: strNimi(30)

Number: nFetch

Message Actions

Dialog Box: AboutBox



MA120 MA185 2MA185

0-4 % 4-5 % 5-6 % 6-7 % 7-8 % 8- %

NOKIA OY / VERSOFT OY

PJ-VERKKOPIIRROS

VELHO (TM)

1: 2500

Velho laskee sähköverkon

Energialaitosten tietojärjestelmään Nelliin kuuluva Velho on sähköverkkoyritysten suunnittelu- ja hallintaohjelmisto, joka sisältää kaikki verkoston suunnittelussa, seurannassa, tilastoinnissa ja huollossa tarvittavat toiminnot.

Ohjelmisto laskee koko jakeluverkon (pien- ja keskijänniteverkko) tai sen osien sähkötekniikan tilan ja antaa listan raja-arvot ylittävistä kohteista. Solmupisteiden kuormitustiedot haetaan Asta-asiakastietojärjestelmästä.

Uutta verkkoa suunniteltaessa annetaan lähtötietoina suunniteltavan verkon kaikki mahdolliset johtoreitit sekä tulevien kuormituspisteiden arvioidut kuormitukset. Ohjelmisto osaa myös mitoitaa keski- ja pienjänniteverkojen pylväät, orret ja harukset sekä laskea riippumat ja vapaaväli eri sääolosuhteissa.

Lisätietoja: Matti Mäkelä ja Tommi Kokkola, puh. (931) 509 111.

Yritysgrafiikkaa Boeingilla

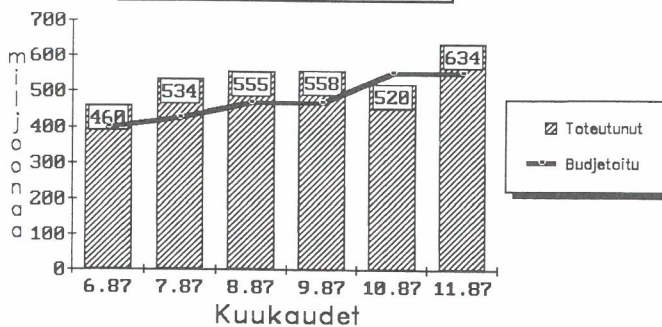
Boeing Graph on uusi monipuolinen mutta helppokäyttöinen yritysgrafiikkaohjelmisto, jolla on mahdollista tuottaa kaksi- ja kolmiulotteisia kaavioita kaksi- tai nelivärisenä. Kolmiulotteisia kuvia voidaan pyörittää, akselien suhteita muuttaa ja samaan kuvaan on mahdollista esittää useita eri kuvaajia.

Ohjelmisto osaa lukea Lotus 1-2-3:n, Multiplanin ja Symphonyn taulukkotiedostoja sinällään sekä muiden ohjelmien SYLK-, DIF- ja ASCII-tiedostoja. Boeing Graphin virtuaalimuisti mahdollistaa suurienkin tietomäärien yhtäaikaisen käsittelyn.

Tulostukseen voidaan käyttää erilaisia matriisi-, mustesuihkua ja lasertulostimia. Paperitulosteen eroteltutarkkuus on riippuvainen tulostimen eikä näytön tarkkuudesta.

Lisätietoja: Käyttötuki, puh. (90) 567 2200.

LIKEVAIHTO loppuvuosi 87



Microsoft Chart tekee grafiikkaa

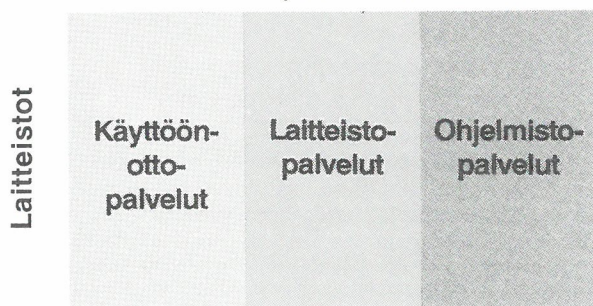
Microsoftin Chart on yritys- ja tieteelliseen käyttöön tarkoitettu grafiikkaohjelmisto. Käyttöliittymä noudattaa Microsoftin vakiintunutta käytäntöä eli esimerkiksi Multiplanin käyttäjä löytää helposti tutut komennot näytön alareunasta.

Normaaleista yritysgrafiikkaohjelmistoista poiketen Chart tarjoaa hyvät apuvälineet myös tieteellisten ja tilastollisten esitysten tekemiseen. Esimerkiksi regressiokäyrät voidaan laskea automaattisesti.

Lähtötiedot voidaan lukea joko näppäimistöltä tai tiedostosta (Lotus 1-2-3, Symphony, Multiplan, dBASE, DIF, ASCII) ja myös tulostus voidaan ohjata tiedostoon, joten Chartin kuvat voidaan siirtää suoraan esimerkiksi Venturaan. Kuvat voidaan myös linkittää tiettyyn lähdetiedostoon, jolloin kyseiseen tiedostoon tehtävät muutokset heijastuvat kuvaajaan automaattisesti.

Lisätietoja: Käyttötuki, puh. (90) 567 2200.

Ohjelmistot



KÄYTTÖTURVA

Käyttöturva turvaa toimivuuden

Nokia Data tarjoaa toimittamilleen mikrotietokone-, lähiverkko-, Unix- (Mikko 100, Mikko 200 & Sun), S2500- ja DPS 6 -järjestelmille uutta yhdistettyä käyttöturvapalvelua.

Käyttöturva koostuu kolmesta eri osasta: käyttöönotto-, laitteisto- ja ohjelmistopalveluista. Esimerkiksi Unix-järjestelmien käyttöönottopalvelut sisältävät verkon kaapeloinnin suunnittelun ja toteutuksen, esiasennuksen, asen-

nuksen ja koulutuksen. Laitteistopalvelut muodostuvat eri tasoisista laitehuoltosopimuksista (Ke 1, Ke 2, Ko 1 ja Ko 2) sekä atk-tarviketoimituksista. Ohjelmistopalvelu koostuu puolestaan käyttöneuvonnasta, ohjelmistojen käyttö- ja palvelusopimuksesta sekä koulutuksesta.

Lisätietoja lähimmästä Nokia Datan asiakaspalvelupisteestä, puh.

Kamu suunnittelee yhdyskuntatekniikkaa

Nokia Data ja kajaanilainen Cae-Matic Oy ovat kehittäneet yhteistyössä avoimessa työasemaverkossa toimivan yhdyskuntatekniikan suunnittelujärjestelmän Kamun. Järjestelmä sisältää liittymät myös olemassa oleviin tietorekistereihin.

Kamu koostuu seitsemästä osasta. Mittausohjelmisto hyödyntää pisterekisteriä ja automaattista takumetriiliäntä maastotutkimusten hyödyntämiseksi. Kaava- ja verkostosuunnittelun lähtötiedot saadaan maaperätutkimusohjelmistosta, joka tulostaa pohjatutkimuskarttoja, kairausdiagrammeja, rakeisuus-käyriä ja pituusleikkauksia.

Kaavasuunnitteluohjelmiston avulla voidaan laatia uusia kaavoja ja kaavamuunnoksia jo kaavoitetuille alueille. Ohjelmisto osaa laskea automaattisesti esimerkiksi eri alojen pinta-alat ja kohteiden etäisyydet.

Katu- ja tiesuunnittelua sekä liikenneteknistä suunnittelua varten on omat ohjelmistonsa, jotka hyödyntävät esimerkiksi mittausohjelmiston antamia tietoja. Eri tielinjavyötehtöjen vertailu onnistuu näin helposti myös kustannusten osalta.

Vesihuolto- ja kaukolämpösuunnittelu vastaavat kunnallistekniikasta ja mahdollistavat järjestelmien yksityiskohtaisen suunnittelun ennen toteutusvaihetta.

Lisätietoja: Tekniset järjestelmät, puh. (90) 567 4043 tai Cae-Matic Oy Kajaani, puh. (986) 140 736.

Plussaa DeskJetille

DeskJet-kirjoittimesta on tullut markkinoille uusi Plus-malli, joka on muuten yhtenevä vanhan DeskJetin kanssa, mitä nyt tulostaa tekstiä tuplasti ja grafiikkaa viisi kertaa nopeammin.

Tulostusnopeuden lisäksi sisäänrakennettua merkistövalikoiden on parannettu. Merkkien enimmäiskorkeus on 30 pistettä, minkä lisäksi kirjoittimesta löytyy puoli megatavua muistia itse määriteltäviä merkistöjä varten. Myös suhteutettujen merkistöjen tasoa on parannettu.

Lisätietoja: Pirkko-Riitta Ta-sanko, puh. (90) 5671.

Uusia tietoliikenne-ohjelmistoja

MikroMikkojen tietoliikennehyteydet paranevat uusien ohjelmistojen myötä.

5251/11 ja 5251/12 liittävät MikroMikko 3:n tai 3TT:n IBM System 34, 36 tai 38 -koneeseen. MikroMikko toimii tällöin päätteenä, ja Mikon kirjoitin IBM-koneen kirjoittimena. Ohjelmistot tukevat IBM PC Support ja FSU-ohjelmia ja sisältävät oman tiedostonsiirto-ohjelman. Siirtäminen pääteistunnosta MS-DOS-sovelluksen käyttöön onnistuu joustavasti. Toimitus sisältää myös tietoliikennekortin.

3270 BSC ja 3270 SNA/SDLC liittävät MikroMikko 3:n tai 3TT:n IBM- tai NAS-suurkoneen päätteeksi. MikroMikko toimii tällöin IBM 3271 51-C -pääteohjaimena, johon on liitetty yksi 3278/79-pääte. Ohjelma toimii kiinteällä tai valinnaisella SNA- tai BCS-linjalla 1200–9600 baudin nopeudella. Toimitus sisältää myös tietoliikennekortin.

Ascom IV, Ascom TTY ja Ascom VDU 202 mahdollistavat puolestaan tahdistamattoman tiedonsiirron sekä MikroMikko 3-sarjan koneilla (IV) että vanhoilla kakkosilla (TTY ja VDU 202). VDU 202 sisältää VDU 202-emuloinnin.

Lisätietoja: Jouko Salo, puh. (90) 567 2819.

Kahdeksan sivua minuutissa

QMS:n valmistama PS810 on PostScript-yhteensopiva 300 dpi:n laserkirjoitin, jonka tulostusnopeus on kahdeksan sivua minuutissa.

PostScript-ohjelmien tulkitsemisen lisäksi PS810 osaa emuloida sekä HP LaserJet +:aa, HP 7475A -piirturia että Diablo 630:tä.

Merkistöjä on vakiona 35 kappaletta, ja kirjainten koot voivat olla neljästä pisteestä ylöspäin.

Liitännöissä löytyy sekä sarjattä rinnakkaisliitäntä, ja kirjoitinta voidaan käyttää myös SUN-ympäristössä.

Lisätietoja: Pirkko-Riitta Ta-sanko, puh. (90) 5671.

Uusi Lotus 1-2-3 Release 3 tulostaa grafiikkaa

Lotus 1-2-3 3.0 on taulukkolaskentaohjelma, jossa voidaan käyttää kolmiulotteisia taulukkoita, grafiikkaa sekä tietokantatoimintoja. Kolmiulotteisten taulukoiden avulla voidaan järjestää tiedot eri tasoille, jolloin mm. taulukoiden yhteenlasku ja linkittäminen on helppoa. Jokaisessa taulukossa voi olla 256 tasoa, minkä lisäksi useita eri taulukoita voidaan käsitellä työtilassa samanaikaisesti. Ohjelma sisältää aiempaa versiota paremmat grafiikka ja tulostusominaisuudet esityskelpoisen grafiikan tuottamiseksi.

Osittainen uudelleenlaskenta laskee vain muuttuneet tiedot. Mikäli taulukkoa muutetaan laskennan aikana laskenta keskeytyy hetkeksi, minkä jälkeen laskenta jatkuu huomioiden tapahtuneet muutokset.

Lotus 1-2-3 3.0 on yhteensopiva kaikkien 1-2-3-versioiden kanssa.

Ohjelma on suunniteltu 80286- tai 80386-suorittimilla varustettuihin laitteisiin. Sama ohjelmapakkaus tukee sekä DOS- että OS/2-käyttöjärjestelmiä.

Lisätietoja: Jaakko Saarela, puh. (90) 5671.

Uutiset kiinnostivat eniten

Softari 2/89:n lukijakyselyyn vastasi peräti 144 lukijaa. Tulokset olivat sikäli yllättäviä, että Tuoteuutiset osoittautuivat ylivoimaisesti suosituimmaksi jutuksi. Niiden määrää on vain valitettavan vaikea lisätä.

Seuraava lista sisältää muidenkin juttujen sijoitukset, ja pakko tunnustaa, että osa niistä on varsin yllättäviä...

1. Tuoteuutiset
2. Puhuu suomea ja ymmärtää hiirtä (WP 5.0)
3. 90-luku voi alkaa (MM 4)
4. Välkky, ei välkky, välkky, ei välkky, ... (ergonomia)
5. Muistia millä mällätä
6. Enemmän kuin käyttöliittymä (Presentation Manager)
7. SQLWindows-kurssi — Osa 2
8. Excel oppoaa järjestelmiin
9. Verkkotietokanta on kolmen kauppa (SQL-System)
10. Ventura laajeni uusiin ympäristöihin
11. Käyttäjät resurssien tasalle (B. Gates)
12. Lyhyt historia — pitkä tulevaisuus (M. Andersson)
13. Pureeko se (C++)
14. Tien päällä taas (Saarisen Tapsa)
15. Pankkijärjestelmän sydän (NBF)
16. Learning Ada — Part 4

Luvatut t-paidat arvottiin vastanneiden kesken, ja Onnetar suosi seuraavia lukijoita: Juha Helminen, Sirkka Rautanen, Hannu Salo, Kirsi Mäöhönen ja Anssi Järvenpää. Onnea ja menestystä!



00003 HELSINKI

Sopimus 00380/166

VASTAUSLÄHETYS

PL 780/10

Softari

Nokia Data

postimaksun
maksaa
Nokia

Softari 389

Palvelukortti

Softari-kysely

(Anna arvosana (—, 0 tai +) jutusta ja kerro, haluatko aiheesta lisätietoja)

	—	0	+	Tahdon tietoa
Neuraaliverkot matkivat ihmisaivoja ...	☐	☐	☐	☐
Työryhmäverkot ja keskuskoneet	☐	☐	☐	☐
yhteen	☐	☐	☐	☐
Tiimi hajauttaa ja yhdistää	☐	☐	☐	☐
Tandem ja Tiimi yhteen	☐	☐	☐	☐
Liikettä linjoilla	☐	☐	☐	☐
Kilpailijat saman pöydän ympärillä	☐	☐	☐	☐
Käytössä koettua	☐	☐	☐	☐
SyliMikko kulkee mihin vaan	☐	☐	☐	☐
Mikro ei ole tuottavin investointi	☐	☐	☐	☐
Väriä elämään	☐	☐	☐	☐
Learning Ada — Part 5	☐	☐	☐	☐
SQLWindows-kurssi — Osa 3	☐	☐	☐	☐
Tuoteuutiset	☐	☐	☐	☐
Lukijakysely 2/89	☐	☐	☐	☐
Mikkonen	☐	☐	☐	☐

Vastanneiden kesken arvotaan viisi MM4-T-paitaa.

Jos voitaa paitaa, kokoni on: _____

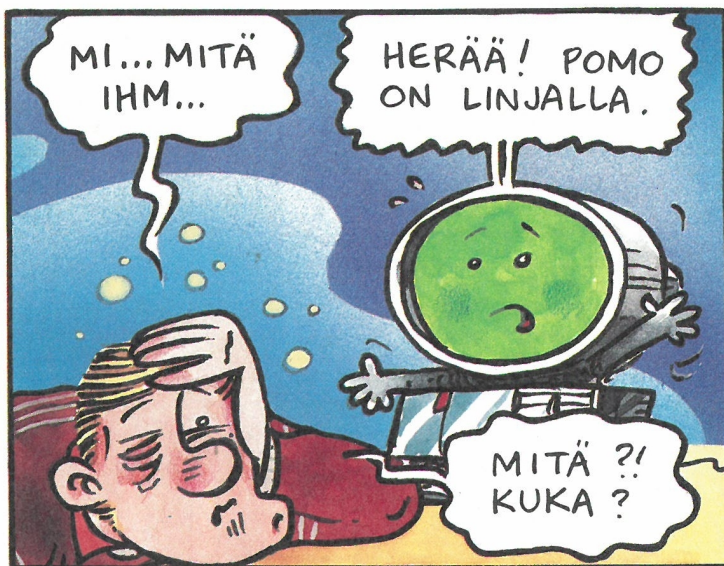
- ☐ Korjaan yllämainitut osoitetiedot.
☐ Poistan yllämainitut osoitetiedot
☐ Lisään allamainitut osoitetiedot
☐ Netin ☐ Softarin rekisteristä/rekisteriin
☐ Haluan lehden kotiosoitteeseeni

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



8/0010

MIKKONEN



Waller