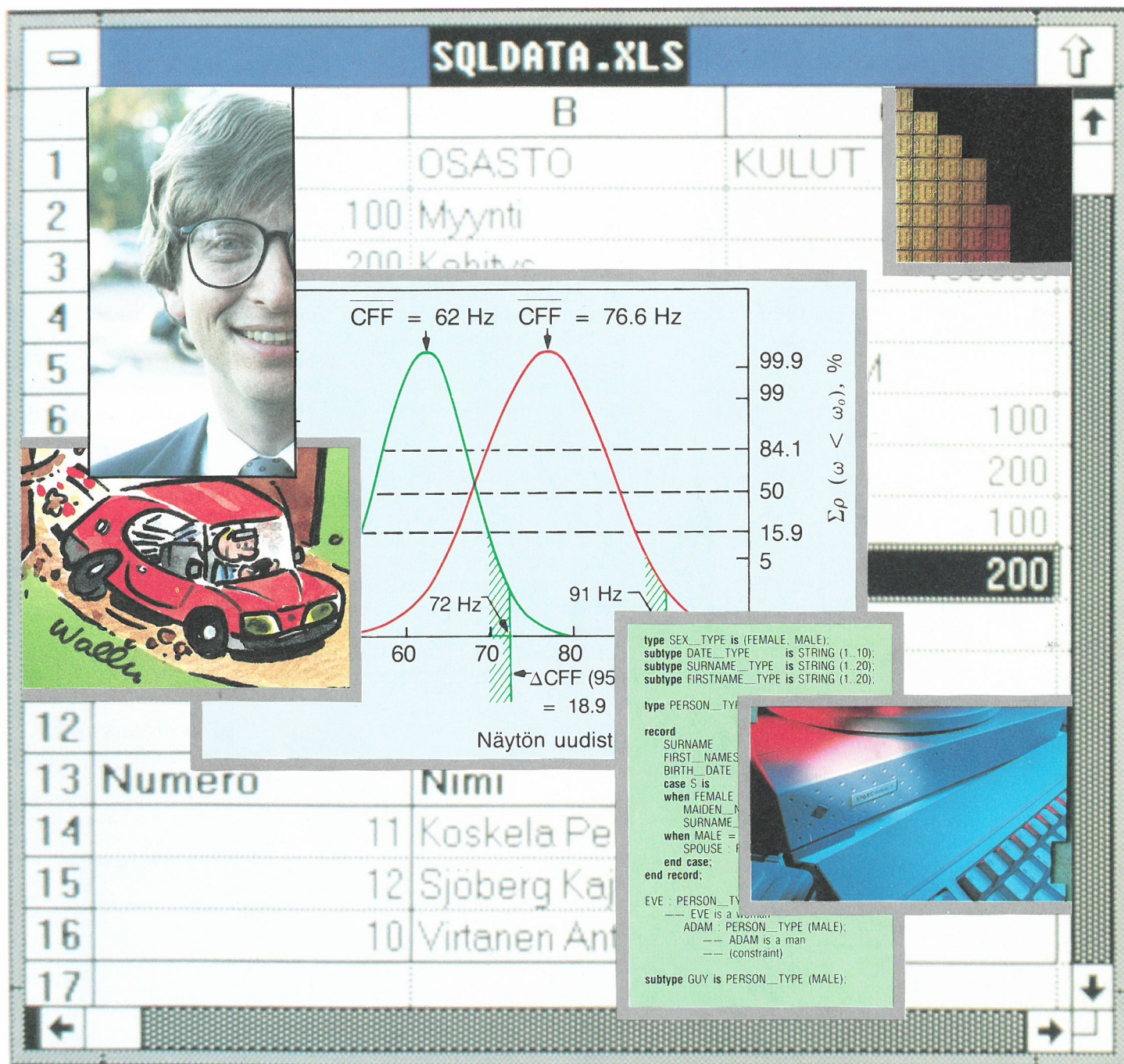


Softari

NOKIA DATAN

OHJELMISTOLEHTI 2/89



**Taistelupari: Microsoft OS/2
ja MikroMikko 4**

Nokia Datan ohjelmistolehti

Päätoimittaja:
Terhi Ogbeide

Toimitus:
Jyrki J. J. Kasvi

Softaritimi:
Tapio Sulkava, Outi Kuusela,
Anne Typpö, Olli Nurminen,
Kirsti Lindfors.

Ulkoasu:
Jyrki J. J. Kasvi

Kuvitus:
Taina Sonninen, Wallu

Kuvakollaasi, ideointi:
Jyrki J. J. Kasvi

Toteutus:
Liikekirjapaino

Osoitteenmuutokset:
Takakannen palvelukortilla

Toimituksen osoite:
Nokia Data
Softari
PL 780
00101 HELSINKI

Toimituksen puhelin:
(90) 567 4064

Painosmäärä:
13 000

Kirjapaino:
Helsingin Liikekirjapaino Oy

Julkaisija:
Nokia Data

ISSN 0782-8209

Sisällys:	
Täällä tänään	2
Käyttäjät resurssien tasalle	3
Enemmän kuin käyttöliittymä	4
Pankkijärjestelmän sydän	6
Lyhyt historia — pitkä tulevaisuus	8
Välkkyy, ei välky, välkkyy, ei välky	9
90-luku voi alkaa	12
Muistia millä mällätä	14
Ventura laajeni uusiin ympyröihin	16
Puhuu suomea ja ymmärtää hiirtä WP 5.0	18
Pureeko se	20
Learning Ada — Part 4	22
Tien päällä taas	26
Verkkotietokanta on kolmen kauppa	28
Excel oppoa järjestelmiin	30
SQL Windows-kurssi — Osa 2	32
Tuoteuutiset	37
Mikkonen	40

Täällä tänään

Vasta parin viime vuoden aikana markkinat ovat kypsyneet kehittyneille työryhmäverkkosovelluksille. Kauniita kaavioita on toki piirrelty jo pitkään, mutta tavalliselta käyttäjältä ovat puuttuneet ne työkalut, joilla hallita verkkoaan tehokkaasti. Lisäksi uusien järjestelmien markkinarynnäkkö on ollut niin kiivasta, että moni on jäänyt odottamaan seesteisempiä aikoja.

Järjestelmien lisääntyvä avoimuus ja standardointi ovat lopulta taanneet sen, että järjestelmän vanhentuminen ei alakaan heti kun eetteriverkon viimeinen piuha on pultattu kiinni. Eri toimittajien uudet laitteet ja sovellukset toimivat järjestelmässä joko sinällään tai pienin sovituksin. Jopa koko systeemin uudelleenjärjestely onnistuu ilman suuria mullistuksia.

Vuosia kuuroille korville saarnatut opit ovat alkaneet kantaa hedelmää, kun käyttäjän ei enää tarvitse välittää siitä, missä hänen tarvitsemansa tieto kulloinkin majaillee. Hajautetun verkkoarkkitehtuurin myötä tiedoston sisältävä looginen palvelin voi olla vaikka toisella puolen

valtakuntaa — tai oman työaseman sisuksissa.

Tie tähän ei ole suinkaan ollut lyhyt eikä helppo, vaan varsinaisen sovelluksen alta löytyy kerroksittain logiikkaa ja tietoliikenteestä vastaavia standardoituja koodia. Eri pulauttimien ja killuttimien nimilyhenteistä saisi koottua vaikka oman kirjansa.

• • •
Tarkkaavaisimmat lukijat ovat jo huomanneet Softarin takasivulla olevan palvelukortin muuttaneen muotoaan. Tästä edes moinen kysely löytyy jokaisesta Softarista siinä missä edellisen kyselyn tulostilastakin. Lehden tulevaisuuteen vaikuttamisen suomen ilon lisäksi kaikki ne, jotka hankkivat korttiinsa elokuun 20. päivää edeltävän postileiman, ovat mukana MikroMikko 4 -paitojen arvonnassa.

• • •
Näin parhaan marja- ja sieniajan lähestyessä kannattaa muuten muistaa, että allekirjoittaneen empiiristen tutkimusten mukaan metsässä samoiltu viikonloppu parantaa tietotyön tuottavuutta keskimäärin enemmän kuin viikon parin kurssi-putki...

Jyrki J. J. Kasvi



Jarkko Mäkinen

Käyttäjät resurssien tasalle

Tietoyhteiskunnassa yrityksen kilpailukyvn määrää sen tapa käsitellä tietoa — miten helposti ja nopeasti henkilöstö saa tarvitsemansa tiedot käsiinsä. Tämä henkilökoh- taisten mikrotietokoneiden läpimurron luoma ”tiedon uusi arkkitehtuuri” on sekä haaste että mahdollisuus koko elinkeino-elämälle.

Henkilökohtaiset tietokoneet ovat nostaneet yksilön tuottavuuden uusiin lukemiin ja tulleet erottamattomaksi osaksi kaikkia organisaatioita. Mutta suurinta osaa töistä eivät tee yksilöt vaan ihmiset, jotka työskentelevät yhdessä — työryhminä.

Suurkoneiden ja mikrojen edut yhteen

Työryhmäverkot ovat mahdollistaneet henkilökohtaisten tietokoneiden liittämisen yhteen ja lisänneet suuresti työryhmien tuottavuutta. Siitä huolimatta työryhmäverkkojen historia on käyttämättä jääneiden mahdollisuuksien historiaa.

Parhaimmillaan työryhmäverkot ovat työryhmätyöskentelyn viisastenkivi, joka yhdistää henkilökohtaisen tietokoneen parhaat puolet — vuorovaikutteisen graafisen käyttöliittymän, korkean suorituskyky/hinta-suhteen, nopean vasteajan ja edulliset helpokäyttöiset sovellukset — suur- ja minikoneiden perinteisiin vahvoihin puoliin eli tiedon ja käytön tehokkaaseen hallintaan, luotettavuuteen sekä tietoturvaan.

Yhteisiä projekteja ja julkisia kirjastoja

Graafisten käyttöliittymien, moniajon sekä tiedonvaihdon ja -yhdistelyn taustalla on myös laajempia näkymiä ihmisistä käyttämässä tietokoneita tulevaisuuden työpaikoilla.

Esimerkiksi joukko yhteistä projektia, kuten sopimusta tai raporttia, työstäviä ihmisiä hyötty suuresti työryhmäverkkojen tarjoamista mahdollisuuksista. Yksi henkilö voi laatia tekstistä luonnoksen, toinen täydentää ja kolmas korjaa — kaikki verkon kautta. Kommentit eri puolilta yritystä voidaan kerätä yhteen ennen päivitystä, josta saadaan myös merkinnät dokumentin uuteen versioon.

”Työryhmäverkkojen historia on käyttämättä jääneiden mahdollisuuksien historiaa.”

Työryhmä voi luoda oman grafiikka- ja tekstikirjaston, josta jokainen ryhmän jäsen voi noutaa tarvitsemansa kuvat ja kaaviot. Siinä sivussa ryhmä voi seurata, kuka on lukenut tietoja, ja voi jopa estää joitakin käyttäjiä tekemästä tiettyjä muutoksia.

Mikä tärkeintä, tämä kaikki onnistuu ilman, että ryhmän jäsenen täytyy tietää, ovatko he yhteydessä suurkoneeseen vai itsenäisessä tiedostopalvelimessa sijaitsevaan tietokantaan.

Standardoituja graafisia käyttöliittymiä hyödyntävät sovellukset muuntavat käyttäjän antamat hakukäskyt tietoliikennestandardien — käytännössä SQL:n (Structured Query Language) — mukaisiksi kyselyiksi. Tällöin esimerkiksi kehittyneitä MS OS/2 -palvelimia käyttävät työryhmäverkot voivat toimittaa tiedon sen tarvitsijalle ilman, että tämän tarvitsee tietää tiedon lähettä.



Bill Gates, 33, on monikansallisen Microsoft Corp. -ohjelmistotalon perustaja ja toimitusjohtaja. Hän on nykyaikaisen tietojenkäsittely- ja toimistoautomaatioteollisuuden kantavia voimia.

MS OS/2 ja Presentation Manager

Enemmän kuin käyttöliittymä

Presentation Manager muistuttaa MS-DOS-ympäristön Windowsia sekä asemaltaan, käyttöliittymältään että ohjelmointiympäristöltään. Siihen saatavien sovellusten määrä on vielä vähäinen, mutta tilanne muuttunee lähiaikoina.

Hallitseva asema

Presentation Manager on MS OS/2 -ytimen palveluita käyttävä sovellus, joka hallitsee yhtä järjestelmän tarjoamista rinnakkaisista näyttöruuduista. Presentation Manager saa ytimeltä näytön, jonka se jakaa tarvittaessa omissa ikkunoissaan ajettaville ohjelmille. Ikkunat voivat olla myös lomittain.

MS-DOS:ista puuttui joukko graafisen käyttöliittymän tarvitsemia käyttöjärjestelmäpalveluita, jotka Windowsin täytyi toteuttaa itse. MS OS/2:n ydin tarjoaa laajat ja monipuoliset muistinhallinta- ja moniajopalvelut, joten Presentation Manager voi olla puhtaammin sovellus. Toisaalta MS OS/2 -ytimen palvelut tekevät mahdolliseksi sen, että Presentation Manager vaikuttaa MS OS/2:een voimakkaasti integroidulta.

Vaikka Presentation Manager on sovellus, se ottaa hallitsevan aseman järjestelmässä. Se käynnistyy automaattisesti ensimmäiseksi istunnoksi ja tarjoaa välineet muiden istuntojen ja ohjelmien hallintaan. Lisäksi myöhemmät istunnot ja ohjelmat käynnistetään Presentation Managerin palveluilla. Samoin niitä ja koko järjestelmää ohjataan Presentation Managerin tarjoamilla apuohjelmilla.

Siinä missä Microsoft OS/2:n versio 1.0 oli pelkkä käyttöjärjestelmäydin komentotulkkeineen, muistinhallintoineen ja moniajotukineen, versio 1.1 sisältää myös graafisen käyttöliittymän ja ohjelmointiympäristön — Presentation Managerin.

Esikuvana Windows

Presentation Manager muistuttaa käyttöliittymältään suuresti Windowsia, ja pienehköt olemassaolevat erot tulevat vähenemään Windowsin tulevissa versioissa.

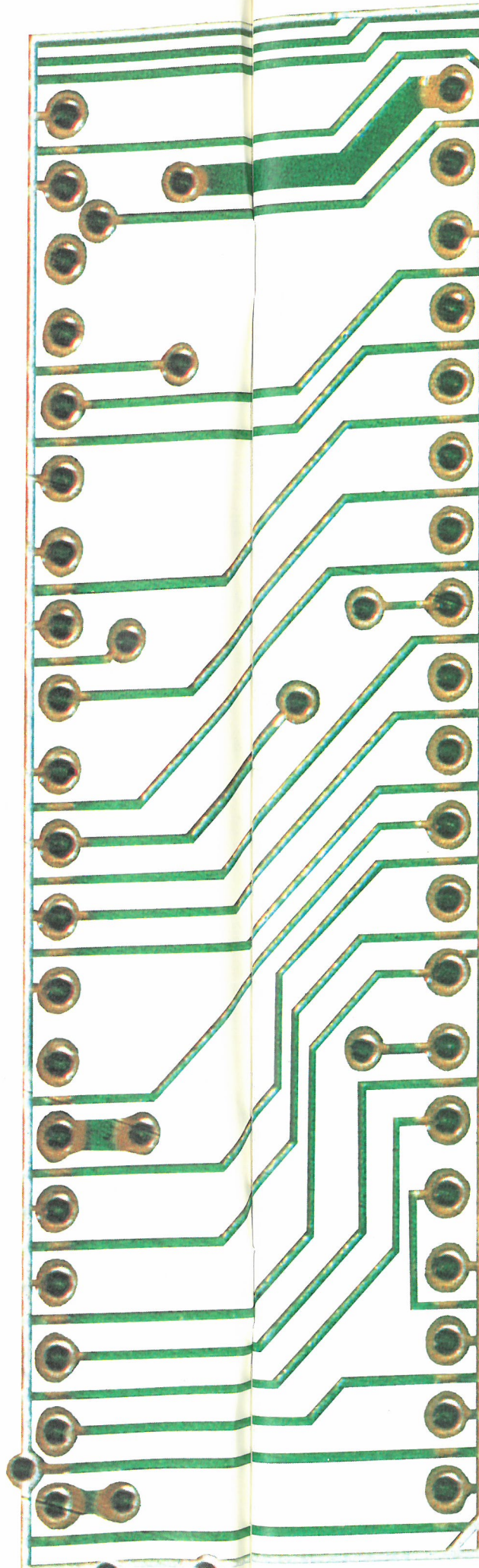
Ohjelmat Presentation Manager jakaa kolmeen luokkaan sen mukaan mitä palveluita ne tarvitsevat.

Aidot Presentation Manager -ohjelmat on tehty varta vasten Presentation Manageria varten ja ne käyttävät sen palveluita ikkunoiden ja valikoiden luomiseen, piirtämiseen jne.

Merkkipohjaisia ohjelmia voidaan ajaa suoraan Presentation Managerin ikkunassa. Ohjelman ei tarvitse tietää Presentation Managerista mitään, ja jos se käyttää pääasiassa tiedostoja ja standardikantavia syöttöön ja tulostukseen, se voi toimia jopa samalla kuvaruudulla Presentation Managerin kanssa.

Lopuksi tulevat vielä ne ohjelmat, jotka tarvitsevat oman kuvaruudun. Nämäkin ohjelmat voidaan käynnistää Presentation Managerista, mutta niille varataan oma koko kuvaruudun istunto.

Istunnosta toiseen päästään tietyllä näppäinyhdistelmällä. Presentation Managerin ikkunasta toiseen päästään samalla näppäinyhdistelmällä tai suoraan hiirellä osoittamalla. Haluttaessa aktiivinen sovellus voidaan valita myös Presentation Managerin valikosta.



Sovellukset lisääntyvät

MS OS/2:n yleistymistä on hidastanut sitä hyödyntävien sovellusten vähäisyys. Sovellusten tekoa taas on viivyttänyt Presentation Managerin odottelu.

Nyt kun Presentation Managerin julkaisemisesta on kulunut puolisen vuotta, tilanne on suurin piirtein ennallaan. Merkittäviä ohjelmia ei tuotantoversioina juurikaan ole. Sen sijaan useiden Windows-sovellusten esiversioita on viime aikoina voitu esitellä myös Presentation Managerin alaisuudessa. Onkin odotettavissa, että syksyn ja talven mittaan markkinoille alkaa tulla tuotantokelpoisia versioita taulukkolaskentaohjelmista, piirto-ohjelmista, tekstinkäsittelyohjelmista, julkaisujärjestelmistä, sovelluskehittimistä, jne.

Windows-ohjelman siirto Presentation Manageriin on varsin suoraviivaista.

Pois perinteisestä

Presentation Manager tarjoaa Windowsin tapaan oman ohjelmointiympäristön, joka eroaa ratkaisevasti perinteisestä ratkaisusta. Windows ja Presentation Manager ovat ohjelmoinnin kanalta samantapaiset, mutta eivät samanlaiset.

Windows-ohjelman siirto Presentation Manageriin on varsin suoraviivaista. Sen sijaan uuden ohjelman teko ympäristöön on melko työlästä, joskin alkuun pääsyä helpottavat olennaisesti valmiit esimerkkiohjelmat.

Ikkunat ja käsittelijät

Presentation Manager -ohjelmointi poikkeaa totutusta jo ajatustavaltaan. Järjestelmän keskeinen osa on viestien lähettäminen,

välitys ja käsittely. Kun jokin tapahtuu, järjestelmä päättää, mihin ikkunaan asia liittyy ja lähettää ikkunaan liittyvälle käsittelijälle viestin tapahtuneesta. Käsittelijä on aliohjelma, joka haarautuu viestin tyyppin perusteella ja joko käsittelee viestin tai välittää sen edelleen, tyypillisesti järjestelmän oletuskäsittelijään.

Presentation Manager -ohjelman perusosat ovat alustus, joka ilmoittautuu järjestelmälle ja luo ohjelmalle yhden tai useamman ikkunan, viestin luku viestijonosta sekä viestijonosta luetun viestin käsittely.

Viesti aiheutuu esimerkiksi kun käyttäjä on painanut hiirtä sovelluksen ikkunassa, jolloin sovellus saa tiedon painalluksesta ja kohdistimen paikasta. Vastavasti käyttäjä on voinut tehdä valinnan sovelluksen määrittelemistä valikoista, jolloin sovellus saa tiedon siitä mitä on valittu. Käyttäjä on voinut myös siirtää sovelluksen ikkunaa, jolloin sovellus saa käskyn piirtää se uudelleen, tai käyttäjä on siirtänyt muita ikkunoita niin, että sovelluksen ikkunasta on paljastunut niiden alla olleita osia, jolloin sovellus saa käskyn piirtää ikkunansa uudelleen.

On huomattava, että suuri osa järjestelmän toiminnoista tapahtuu ilman, että sovelluksen tarvitsee niistä välittää. Tällaisia ovat esimerkiksi kohdistimen liikuttelu hiiren liikkeiden mukaan, valikoiden hoito, ikkunoiden liikuttelu ja koon muuttaminen.

Ohjelmointityökalut

Presentation Manager -ohjelmien teko tapahtuu pääosin C-ohjelmointikielellä. C-kääntäjän lisäksi tarvitaan liittymät järjestelmän palveluihin ja eräitä erikoisohjelmia sovellusten käyttämien keskusteluikkunoiden, kirjaintyyppien ja graafisten symbolien määrittelyyn ja käsittelemiseen. Tällaisia symboleja ovat esimerkiksi kohdistimet ja ikonit, pienet kuvat jotka edustavat ohjelmaa silloin kun se itse ei ole näkyvissä. ■



Jukka Aspelund toimii Nokia Datan MS OS/2 ja LAN Manager -tuotteistusprojektin projektipäällikkönä.

Sarjafoto

Nokia Datalla on pitkä historia pankkijärjestelmien ja -päätteiden kehittäjänä ja toimittajana yhteistyössä lähes kaikkien pankkien kanssa. Suomen markkinoiden säilyttämisen lisäksi tavoitteena on nousta johtosemaan myös Ruotsissa sekä saada merkittävä osuus Euroopan markkinoista. Tavoitteiden saavuttamiseksi on kehitetty uusi sovellustukijärjestelmä — NBF.

Nokian uusi MS OS/2 -pohjainen Alfaskop Financial System (AFS) on pankkikonttoreille tarkoitettu tietojärjestelmä, jonka hajautettu rakenne mahdollistaa sopeutumisen eri maiden ja pankkien vaihteleviin järjestelmiin.

AFS perustuu Nokian Tiimityöryhmäverkkorajajärjestelmään, joka puolestaan hyödyntää Microsoftin OS/2 -käyttöjärjestelmää, TokenRing- tai Ethernet-verkkotekniikkaa, MS LAN Manageria yhteydenpitoon (katso Softari 1/89) sekä 286- ja 386-pohjaisia tietokoneita sekä monikäyttöisinä työseminä että palvelimina. Järjestelmään kuuluu myös pankkikohtaisia oheislaitteita kuten pankkikirjoittimia ja magneettikortinlukijoita.

Nippu standardeja yhdessä nytyissä

Nykyinen pyrkimys avoimiin, standardoituihin järjestelmiin pakottaa laitevalmistajat panostamaan kokonaisvaltaisiin sovelluksiin. AFS:n ytimenä toimiva Nokia Banking Frame (NBF) on yksi esimerkki, jossa niin viralliset kuin elävän elämän standarditkin kuten SQL, SNA, OSI ja OS/2 sekä 286- ja 386-pohjainen teitokonetekniikka toimivat kehitystyön perustana.

Valmiita rutiineja omaan kehitystyöhön

NBF on sovellustuki ja -kehitysjärjestelmä, joka on suunniteltu erityisesti pankkipalveluja silmällä pitäen. Se on yhteensopiva kaikenlaisien OS/2-sovellusten kanssa, on kyse sitten vakio-ohjelmistoista, räätälöidyistä oh-

Pankkijärjestelmän sydän

Nokia Banking Frame



Jarmo Hietaranta

jelmista tai asiakkaan omista sovelluksista.

NBF:n ajonaikainen osa yhdistää käyttöoikeusvalvonnan, joka huolehtii järjestelmän perusturvallisuudesta, joustavan sovellusvalikon ja sovellushallinnon, joka muiden toimien ohessa pitää huolta OS/2-työasemien yhtenäisyydestä. NBF:n palvelurutiinikirjasto tarjoaa pankkiohjelmuksen kehittäjälle valmiit funktiot yleisimpien pankkitapahtumien tekemiseen. Tapahtumanhallinta (transaction manager) huolehtii virhetiloista toipumisesta ja toimintojen peruuttamisesta parantaen järjestelmän toimintavarmuutta.

NBF-kehitysympäristö antaa käyttäjälle mahdollisuuden hyödyntää ja hallita järjestelmän kaikkia voimavaroja erityisen korkean tason ohjelmointikielen, Transaction Description Language (TDL) avulla. TDL yhdistää näytönhallinnan, tulokset ja tietoliikenteen järjestelmätasolla. Sovelluksen pääosat eli näytöt, paperitulosteet ja ohjelman rakenne voidaan määritellä interaktiivisesti. Sovellusten kehittäjät voivat hallita ohjelmien kulkua osittain tai kokonaan myös C-kielen avulla ilman, että valmiiden NBF-pankkipalvelurutiinien hyödyntäminen siitä mitenkään hankaloituu.

NBF:n lomakkeenkäsitely perustuu Nokian omaan ERGO-FORMS (aiemmin MPSFORMS) näytön- ja näppäimistönsäilytyjärjestelmään. Tietokantana käytetään SQLBase-relaatiotietokantaa.

Hajautetut resurssit ja niiden hyödyntäminen

Hajautettu arkkitehtuuri pohjautuu loogisiin työasemiin ja loogisiin palvelimiin, jolloin jokainen järjestelmä voidaan määritellä optimaalisesti. Samassa fyysisessä palvelimessa voi sijaita monta loogista palvelinta siinä missä yhdessä työasemassa voi parhaimmillaan majailla sekä looginen työasema että nippu loogisia palvelimia.

Työasema tarjoaa käyttöliittymän ja sisältää enemmän tietojenkäsittelykapasiteetin kun taas palvelimet tarjoavat tiedosto-, tulostus-, tietoliikenne- ja tietokantapalveluita. NBF:ää käyttävälle ohjelmiojalle tällainen hajautettu järjestelmä on täysin läpinäkyvä eli hänen ei tarvitse huomioida sitä, missä mikin tieto tai palvelu majoilee.

Omat sanomat omalla kielellä

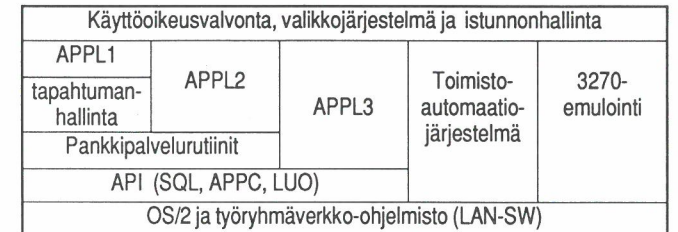
Liitettävyys, tai paremminkin viestintäkyky, on myös pankkipalvelujärjestelmän kohtalonkysymys. NBF sisältää oman toiminnallisen rajapinnan varsinaisen tietoliikennepalvelun yläpuolella peittäen yhteiskäytännön sovellukselta. Sanomat kuvataan sovellusten ulkopuolella MDLE-sanomankuvauskielen (Message Description Language) avulla.

Monta työtä kerralla selväksi

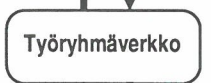
Monilla pankkialan sovelluksilla on yhtäläinen perusrakenne, ja NBF:n sovelluksenkuvauskieli TDL on suunniteltu vastaamaan juuri tätä rakennetta pankkipalvelurutiinikirjastoon kaikkineen. Näin päästään mahdollisimman tehokkaaseen tulokseen vähillä kehitys- ja ylläpitokustannuksilla. Myös järjestelmään sisällytetty voimavaroja hallitseva kehitysympäristö alentaa omalta osaltaan ylläpitokustannuksia, koska se helpottaa huolto- ja jatkokehitystoimenpiteitä.

Laaja laitetarjonta (työasemat ja työryhmäverkot) mahdollistaa jokaisen alajärjestelmän suunnitteleminen suorituskyky/hintasuhteeltaan kyseisissä olosuhteissa mahdollisimman edulliseksi.

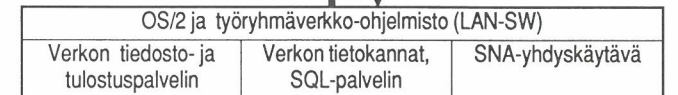
NBF:n toiminnallinen rakenne, esimerkiksi tietoliikennejärjestelmän erottaminen sovelluksesta, mahdollistaa asteittaisen kehityksen tekniikan kehittyessä ilman, että aiemmat laitehankinnat jäävät nurkkiin pölyttymään. Lisäksi NBF:n avoin arkkitehtuuri mahdollistaa muiden toimitta-



Työasema



Palvelin



jien ohjelmistotuotteiden lisäämisen järjestelmään edullisimman mahdollisen sovellusympäristön luomiseksi.

Hallittu rinnakkaisprosessointi tarjoaa käyttäjälle enemmän voimavaroja päivittäisten työtehtävien hoitamiseen. Käyttäjä voi ajaa useampaa kuin yhtä sovellusta yhtä aikaa eikä hänen tarvitse lopettaa yhtä ennen toisen aloittamista. Eli kun asiakas saapuu tiskille, pankkikassan voi jättää kesken olleen tehtävän palvelukseen asiakasta samalla työasemalla ja palata sen jälkeen takaisin alkuperäisen tehtävänsä pariin.

Australia tänään, mitä huomenna?

NBF:n ideat ja niiden toteutukset ovat johtaneet toimituksiin kaikkialla maailmassa. Pankit niin Suomessa, Espanjassa kuin Australiassakin ovat saaneet uudet NBF-järjestelmänsä eri valmiusasteille yhteistyössä Nokian tai Nokian paikallisen edustajan kanssa.

Tänä vuonna Ranskan ja Espanjan markkinat vaikuttavat kaikkein lupaavimmilta. Pitkällä tähtäimellä NBF pyrkii pankkipäätejärjestelmämarkkinoille kaikkialla, missä Nokia on edustettuna. ■

BF-ohjelmiston ajonaikainen arkkitehtuuri. Tyypillinen järjestelmä emuloi yleisesti käytössä olevaa 3270:aa hyödyntääkseen vanhoja ohjelmistoja uudessa ympäristössä. Yleensä se sisältää myös toimistoautomaatiojärjestelmän. Kuvan esimerkkitapauksessa nämä toiminnot käyttävät eri palvelintoimintoja MS LAN Managerin tasolla.

APPL3 on liittymä SQL-tietokantaan siinä missä APPL1 ja APPL2 ovat NBF:n avulla kehitettyjä sovelluksia. Molemmat käyttävät samoja pankkipalvelurutiineja, mutta APPL1 käyttää TDL-kielillä laadittua käyttöliittymää kun APPL2:n suunnittelussa on turvaututtu C-kielen.

Pankkipalvelurutiinit kattavat esimerkiksi pankkitietokantojen, rahankulun seurannan, tapahtumalokin, yhteyden katkaisemisen ja tiedon salauksen. Ne vastaavat myös kaavakeista ja tulostuksen muotoilusta sekä viestien muodostamisesta ja tulkitsemisesta.

Kaikki sovellukset on suojattu sekä käyttöoikeusvalvonnan, valikkojärjestelmän että istunnonhallinnan kautta.

Peter Gregfors toimii Nokia Data Systems Ab:n pankkijärjestelmien tuotepäällikkönä Tukholman Kistassa.

Lyhyt historia — pitkä tulevaisuus

Microsoft ja OS/2

Koulutukseltaan Magnus Andersson on diplomi-insinööri ja hän on valmistunut Tukholman kuninkaallisesta teknillisestä korkeakoulusta vuonna 1987. Pääaineenaan hän luki tietojenkäsittelyoppia kuten arvata saattaa.

Kimmokkeen ohjelmistotalalle antoi Yhdysvalloissa tietojenkäsittelyä opiskellut sukulainen, joka sai vuosikymmenen alussa lukiotaan päättävän Magnuksen vakuuttuneeksi alan kehityspotentiaalista.

Käyttöjärjestelmät kalkkiviivoilla

Magnus Andersson näkee jokaisella käyttöjärjestelmällä omat puolensa. Esimerkiksi vanha MS-DOS on muodostunut käytännön standardiksi, jolle suurin osa ohjelmistokehityksestä on pohjautunut — ja pohjautuu vielä pitkään.

Eri UNIX-järjestelmät ovat puolestaan lähdekooditasolla yhteensopivia siinä missä OS/2:n eri toteutukset ovat binääritasolla yhteensopivia.

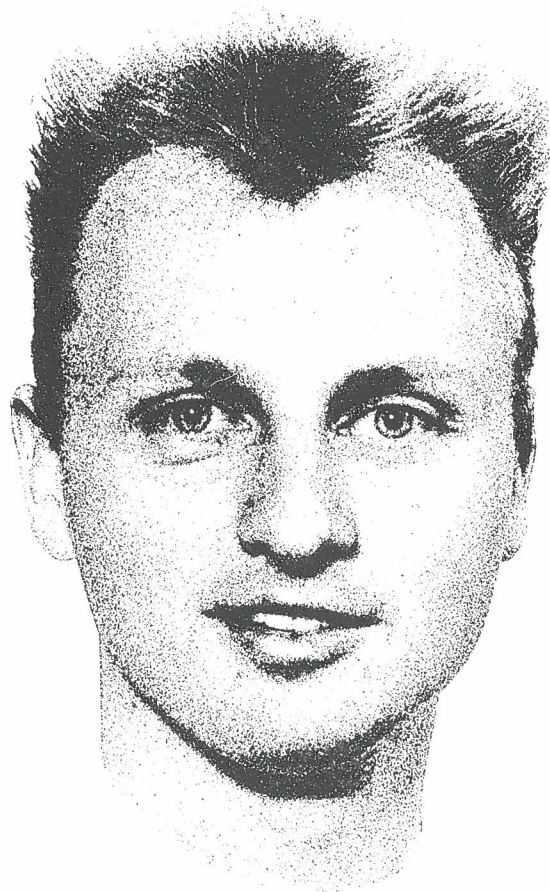
OS/2 tarjoaa hyvän pohjan myös työryhmyöskentelylle. Lisäksi se osaa hyödyntää 386-suorittimien suorituskykyä kun taas MS-DOS on alun perin suunniteltu huomattavasti heikkotehokkaammille suorittimille: ”On tärkeää ottaa uuden hyvän järjestelmän voimavarat käyttöön, mutta yhteensopivuutta ei saa koskaan unohtaa.”

Yhteistyönä IBM:n kanssa

Vihjaukset Microsoftin ja IBM:n OS/2:n eroavaisuuksista Andersson torjuu kertomalla koko käyttöjärjestelmän syntyneen Microsoftin ja IBM:n yhteistyönä. Kummankin OS/2 1.1:t ovat täysin identtisiä. Vain laajennussovelluksissa ja oheislaiteajureissa voi olla pieniä eroja.

OS/2:n historia on varsin lyhyt, sillä järjestelmän kehitys aloitettiin vasta vuonna 1985 ja se julkistettiin 1987. Ensimmäinen versio päästettiin markkinoille 1988 siinä missä graafisesta

Pohjoismaissa Microsoft Corporationin ohjelmatuotteita edustaa Tukholmassa majaansa pitävä Microsoft AB. Heidän OS/2-asiantuntijanaan toimii Magnus Andersson, ja pitihän miestä käydä näin OS/2:n läpimurto-vuonna jututtamassa.



käyttöliittymästä vastaava Presentation Managerkin.

Tänä vuonna julkaistava SQL Server vastaa tietokantojen ylläpidosta, ja Communications Server mahdollistaa myös MS-DOS-pohjaisten työasemien yhdistämisen muuten OS/2-asemia käyttävään verkkoon.

OS/2:n elämäankaari on kuitenkin vasta alullaan: ”OS/2:sta on muodostumassa uusi käyttöjärjestelmästandardi — uusi MS-DOS. Koska ihmiset arvostavat yhä enemmän yhteensopivuutta eri järjestelmien välillä, OS/2 tulee elämään vielä pitkään — ainakin seuraavat kymmenen vuotta.”

OS/2 soveltuu sekä 386- että 486-suorittimille eli laitteistokehityskään ei tule aivan heti ajamaan sen ohitse.

Painopiste työryhmäverkoissa

Aina siitä lähtien, kun MS LAN Managerin suorituskyky kävi 1987 ilmeiseksi, OS/2:ta on kehitetty verkkosovellusten suuntaan niin ohjelmisto- kuin järjestelmäkehityspuolellakin. Tuloksena on ollut joukko erilaisia uusia käsitteitä kuten dynaaminen palvelin.

Käytännössä OS/2 soveltuu niin työaseman kuin palvelimenkin käyttöjärjestelmäksi, ja molemmat voivat järjestelmän dynamiikan seurauksena sijaita fyysisesti vaikka saman mikron kuorien sisällä — ja pyörittää samaa käyttöjärjestelmää yhtä aikaa.

Järjestelmätason sovellusten kirjoittaminen on OS/2-ympäristössä helppoa kattavien standardien ja valmiiden kehitysohjelmistojen avulla. Lisäksi järjestelmä on dokumentoitu poikkeuksellisen hyvin jo etukäteen.

Binääritason yhteensopivuuden myötä myös käyttöliittymät ja tiedonsiirto yhtenevät. MS LAN Managerin avulla voidaan erilaisia kirjastoja ja palvelinjärjestelmiä yhdistellä toimiviksi kokonaisuuksiksi, joita jokainen verkon käyttäjä voi käsitellä kuin omiaan.

Presentation Manager piilottaa OS/2:n

OS/2:n graafisen käyttöliittymän eli Presentation Managerin tehtävänä on itse asiassa peittää koko OS/2 käyttäjän näkyvistä. Näin käyttäjä voi keskittyä itse sovelluksiin. ”Presentation Manageria voidaan käytännössä kuvata parhaiten UNIX-ympäristöstä tutulla ’shell’-käsitteellä, jonka kautta annetaan käskyjä varsinaiselle järjestelmälle.”

Presentation Manager on ilmestymässä myös UNIX-maailmaan, sillä Open Software Foundationin X-pohjainen OSF/Motif tuo PM-tyyppisen käyttöliittymän myös UNIX-ympäristöön. ”Motif-projekti yhtenäistää tulevaisuuden käyttöjärjestelmäkäyttöliittymät Presentation Managerin mallin mukaan, jolloin eri järjestelmästä toiseen siirtyminen onnistuu helposti.”



Useat aikaisemmat tutkimukset ovat osoittaneet, että tekstiä luetaan hitaammin näyttöruudulta kuin paperilta. Useimmat tutkimukset on kuitenkin tehty tavallisia näyttöjä käyttäen, jolloin polariteetti (tumma/vaalea tausta merkeillä), tekstityyppi (font) ja erottelukyky (resoluutio) poikkeavat hyvin voimakkaasti painetusta tekstistä.

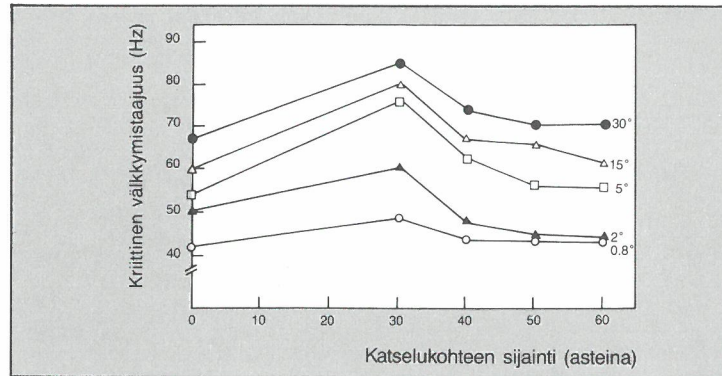
Parempaan lukunopeuteen

Gould on tullut tutkimuksissaan siihen tulokseen, että näyttöltä voidaan lukea tekstiä yhtä nopeasti kuin paperilta, jos näytön ja merkkien ominaisuuksia parannetaan riittävästi. Parannukset kohdistuivat merkkien täsmälleen samanlaiseen ulkoasuun painettujen merkkien kanssa,

erottelukykyyn, polariteettiin ja harmaan sävyjen käyttöön. Mikään edellämämainituista seikoista ei yksinään selittänyt parantunutta lukunopeutta, vaan kyseessä oli niiden yhteisvaikutus. Voitiin kuitenkin osoittaa, etteivät henkilökohtaiset ominaisuudet (ikä, sukupuoli, kokeneisuus jne.) eikä katodisädeputkitekniikassa itsessään olevat tekniset puutteet selittäneet tutkimustuloksia.

Positiivinäytöt ovat silmille ystävällisempiä kuin tavalliset tummapohjaiset näyttöruudut — silmien ei tarvitse mukautua erilaisiin valaistusolosuhteisiin, ja heijastukset ovat vähäisempiä. Positiivinäytöllä työskentelevät valittavat kuitenkin monesti näytön välkkymistä. Positiivinäytöjä arvioitaessa huomio kiinnittyy siten helposti pelkkään uudistustaajuuteen, vaikka välkkymisen aistimiseen vaikuttavat muutkin seikat kuten kuvaruudun kirkkaus, käytetty loisteaine sekä kuvaputken rakenne.

Kuva 1. Keskiarvot (N/6) kriittisestä välkymistajuuudesta. Parametrinä katselu-kohteen koko ja vaaka-akselina katselu-kohteen sijainti asteina (Bauer et al 1983, 31–33).



Suurin osa näytöistä on suunniteltu käytettäväksi niin, että merkit ovat vaaleita ja tausta tumma. Tämä järjestely aiheuttaa suuria luminanssikontrastivaihteluita näkökentässä ja saa helpommin myös aikaan kirkkaiden esineiden kuvastumisen päätteen kuvaputken pinnassa. Useimmissa tavallisissa graafisissa näytöissä olisi mahdollista käyttää myös muita taustavärejä kuin mustaa, mutta tämän seurauksena aistitaan usein kuvaruudun välkymistä.

Positiivinäytöiltä aistittavaa välkymistä on pääasiassa kahta eri tyyppiä: aluevälkymistä ja silmän liikkeestä aiheutuvaa reunavälkymistä. Aluevälkkyminen on koko pinta-alalta aistittavaa välkymistä ja reunavälkkyminen taas johtuu silmän liikkeiden ja uudistustaajuuden interferenssistä.

Aluevälkkyamisen monet syyt

Kriittinen välkymistajuuus (CFF / critical flicker frequency) on se tajuus, jolla valoa ei enää mielletä välkkyväksi, vaan tasaista valoa antavaksi valonlähteeksi. Kirjallisuudessa on esitetty

CFF:n arvoiksi 45–50 hertsiä (tässä välähdystä sekunnissa).

Laine on mitannut omassa koejärjestelyssään tajuudeksi 30–40 hertsiä. Nämä tutkimukset on usein tehty yksittäisellä valonlähteellä eikä näyttöruudulla (katodisädeputkella). Kuitenkin käytännössä on havaittu, että työntekijät ovat valittaneet välkymisestä vaikka heidän päätöksensä uudistustaajuus (virkistystajuus) on 50–60 hertsiä. Uudistustaajuus tarkoittaa sitä tajuutta, jolla elektronisuihku "virkistää" kuvaputken sisäpinnassa olevan loisteaineen.

Näköjärjestelmän herkyys välkymiselle kasvaa välkkyvän pinnan luminanssin eli "kirkkauden" kasvaessa. Lisäksi on todettu, että asiaan vaikuttaa myös pinnan koko ja luminanssin vaihtelun suuruus (modulaatio).

Positiivinäytöjen välkymistä voidaan vähentää kasvattamalla uudistustaajuutta tai valitsemalla loisteaineeksi sellainen aine, jonka loisteominaisuudet kestävät pitemmän aikaa.

Kriittinen tajuus riippuu fosforista

Bauer on tutkinut kuinka näkökentän suuruus ja sen sijainti vai-

uttaa välkkyamisen kokemukseen, ja mikä on kriittinen välkkymistajuus positiivinäytöisellä katodisädeputkella. Näiden tutkimusten perusteella voidaan valita sellainen uudistustaajuus, ettei välkymistä havaita. Tutkimuksessa käytettiin tavallista näyttöpäätettä, jonka loisteaineena käytettiin P4-fosforia. Sen luminanssi vähenee 10 %:iin maksimimäärästä 25 mikrosekunnissa. Näyttöä oli muunneltu niin, että sen uudistustaajuutta voitiin vaihdella 0–150 hertsin välillä. Kuvassa 1 on esitetty kuuden koehenkilön kriittisen välkkymistajuuden keskiarvot, kun parametrinä on näkökentän koko ja abskissa katselu-kohteen sijainti. Katselu-kohteen luminanssi oli 100 cd/mFD. Tuloksista voidaan todeta, että välkkyminen aistitaan suurimpana kun katselukohde on 30° sivussa ja sen koko suuri.

Kriittistä välkkymistajuutta tutkittaessa otettiin huomioon edellämainittu tulos koejärjestelyä laadittaessa. Koe järjestettiin 31 koehenkilölle siten, että taustan luminanssiksi oli valittavissa 80, 160 tai 320 cd/mFD. Katselukohde oli 30° sivussa ja yleisvalaistus oli normaali toimistovalistus (600 luksia horisontaalitasolla). Tulokseksi saatiin, että toimisto-olosuhteisiin sopivalla 80 cd/mFD taustaluminanssille tarvitaan 87 hertsin uudistustaajuus, jotta 95 % koehenkilöistä ei aistisi välkymistä.

Jopa erittäin hyvälaatuisille tekstinäytöille (DIN A4, 740 x 1008 kuvapistettä) on teknisesti mahdollista tehdä positiivinäytöjä, joiden uudistustaajuus on 100 hertsiä, jos videosaanalin kaistanleveys valitaan oikein. Jos kuvapisteiden lukumäärä joudutaan kasvattamaan ratkaisevasti, kuten graafisissa näyttöpäätteissä (CAD:n vaatima näyttö), ei 100 hertsin uudistustaajuudella voida enää saavuttaa riittävä tarkkuutta, koska videosaanalin kaistanleveyttä ei voida pienentää riittävästi.

Loisteaineen hitaus auttaa asiaa

Toinen keino vähentää välkymistä on käyttää "hitaampia" loisteaineita. Tällä tavalla ei loisteaineen luminanssi ehdi laskea kovin paljon ennen seuraavaa uudistuspyyhkäisyä. Loisteaineen kirkkaus pysyy tasaisempana eli modulaatio pienenee.

Bauer on myös tutkinut, miten loisteainevalinnoilla voidaan ehkäistä välkymistä. Hänen koejärjestelyssään oli käytössä kaksi näyttöpäätettä, jotka olivat loisteainetta lukuun ottamatta täysin samanlaisia. Toisessa näyttöpäätteessä käytettiin "nopeaa" P4-fosforia vastaavaa loisteainetta, jonka luminanssi vähenee 10 %:iin maksimimäärästä 0,065 millisekunnissa. Toisessa näyttöpäätteessä oli käytössä "hidas" loisteaine, jonka luminanssi vähenee 10 %:iin maksimimäärästä 0,2 millisekunnissa. Kokeessa tutkittiin 29 koehenkilön aistimaa välkymistä molemmilla näyttöpäätteillä 80 cd/mFD taustaluminanssilla ja tulosten perusteella piirrettiin jakaumat, jotka olivat normaalisia. Tulokset on esitetty kuvassa 2.

Jos vaatimukseksi asetetaan, ettei 95 % väestöstä aisti välkymistä, on uudistustaajuus nostettava nopeaa fosforia käyttävälle näytölle 91 hertsiin ja hidasta fosforia käyttävälle 72 hertsiin.

Hitaat fosforit jättävät kuvaa "vieritettäväksi" kuitenkin jälkikuvia, koska fosforin loiste sammutuu hitaasti. Jälkikuvista on mahdollista päästä eroon käyttämällä tekniikkaa, jossa näytetään vain ruutu kerrallaan. Tällaista ratkaisua ei kuitenkaan voida käyttää kaikkiin sovelluksiin.

Toinen negatiivinen piirre on hitaiden fosforien huonompi hyötysuhde. Samalla elektronisuihkuun energiamäärällä saadaan aikaiseksi matalampi luminanssitaso hitaassa fosforissa kuin nopeassa, ja näin kontrasti taustan ja merkkien välillä on pienempi kuin nopeilla fosforeilla varustetussa näytöissä. Se rajoittaa tällaisten näyttöjen käyttöä vaativissa valaistusolosuhteissa. Energian kasvattaminen puolestaan huonontaa merkkien terävyyttä, koska merkkien ja taustan rajapinta muuttuu epäselväksi.

Reunavälkkymistä on kahta eri tyyppiä

Toista välkkymistyyppiä, reunavälkkymistä, tutkittaessa on havaittu, että sitä esiintyy kahta eri tyyppiä: näyttöruudun reunan kohdistettu katse aiheuttaa reunaviivan aaltoilemisen ja kohdistusliikkeet näytöllä aiheuttavat tummia ja/tai vaaleita välähdyksiä päätteen reunalla. Näistä ongelmista päästään kasvattamalla näyttöruudun reunan (L_{su}) ja sen kuvapinnan (L_{sc}) luminanssi-contrastia siten, että L_{sc} :n suhde L_{su} :hen on 2:1. Toinen tapa ratkaista ongelma on järjestää kuvaruudun reunaa alueelle vähintään 5 mm leveä tumma vyöhyke. Kuvassa 3 on esitetty reunavälkkyamisen tyyppit ja ratkaisu ongelmaan.

Edellämainittu ongelma esiintyy vain sellaisissa näytöissä, joissa käytetään nopeaa fosforia. Kun käytettiin nopeaa fosforia 91 hertsin uudistustaajuudella, (95 %:n vaatimus) kaikki koehenkilöt (N/29) aistivat reunavälkkymistä. Hitaalla fosforilla, jonka uudistustaajuus oli 72 hertsiä (95 %:n vaatimus), ei kuitenkaan aistinut reunavälkkymistä.

Myös muut tutkimusryhmät ovat tutkineet välkkymisilmiötä. Ne ovat kuitenkin enimmäkseen keskittyneet aluevälkkymiseen ja sen vaikutuksiin. Laubli on tutkimuksissaan todennut, että mitä alhaisempi uudistustaajuus on, sitä enemmän koehenkilöt raportoivat silmäoireista. Sama tutkimusryhmä on antanut toisessa tutkimuksessa raja-arvoja uudistustaajuuksille: tummapohjaisille näytöille 60 hertsiä ja vaaleapohjaisille 100 hertsiä. Näihin suositusarvoihin on kuitenkin syytä suhtautua varauksella, sillä koejärjestely ei vastaa kaikilta osiltaan normaalia näyttöä, sen katseluolosuhteita eikä teknisiä eroavuuksia.

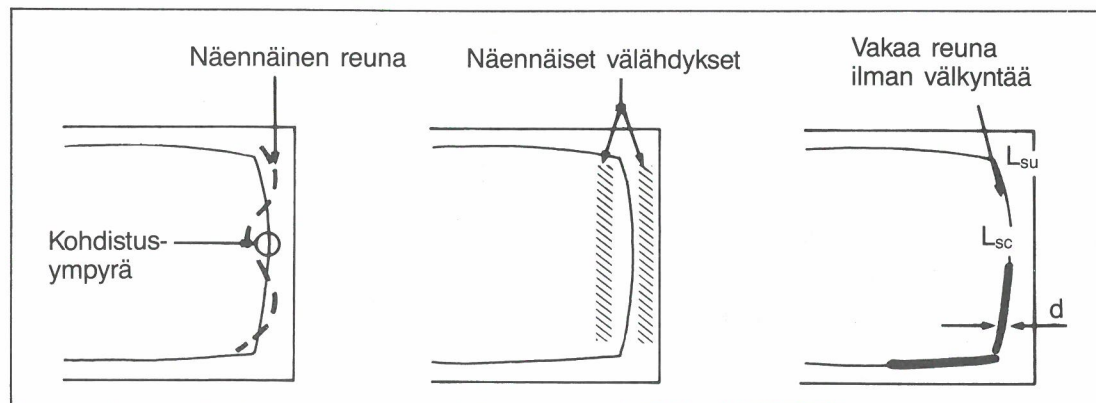
Valkomusta vs. mustavalko

Bauer ja Cavonius ovat tutkineet tekstin luettavuutta sekä positiivinäytöisellä että kahdella negatiivinäytöisellä näyttöpäätteellä. Positiivinäytön uudistustaajuus tutkimuksessa oli 100 hertsiä, taustan luminanssin kirkkauden pystyi säätämään 50–70 cd/mFD, ja valaistusvoimakkuus oli 550 luksia. Uudistustaajuus takaa välkkyttömän näytön nopeillakin fosforeilla. Negatiivinäytön vastaavat arvot olivat: uudistustaajuus 50 hertsiä, tausta joko pienempi kuin 10 cd/mFD tai 80 cd/mFD ja valaistusvoimakkuus 279 luksia. Koejärjestelyssä yritettiin tunnistaa neljä satunnaista merkkiä näytöltä ja kirjoittaa ne näppäimistöllä.

Eniten virheitä esiintyi negatiivinäytöllä, jonka taustan luminanssi oli korkeampi ja vähiten positiivinäytöllä. Positiivinäyttöä piti parempana 22/23:sta koehenkilöstä, ja ainoa eri mieltäkin ollut selvisi tehtävästä paremmin positiivinäytöllä.

Positiivinäytöisen näyttöruudun parhaita ominaisuuksia on sen sijoitettavuus tavalliseen toimistoympäristöön. Bauerin koejärjestelyssä todettiin, että näyttö, joka oli varustettu sopivalla suodattimella, ei menettänyt ratkaisevasti merkkien luettavuutta olosuhteissa, joissa kuvaruudulle heijastui 2000 cd/mFD valonlähde ja horisontaalinen valaistusvoimakkuus oli 5700 luksia.

Positiivinäytöt ovat tulevaisuudessa yksi askel kohti paperitonta toimistoa. Teknisten ratkaisujen täytyy kuitenkin kehittyä edelleen, jotta tekstiä olisi yhtä helppo ja mukava lukea näyttöruudulta kuin paperilta.



Kirjallisuutta:

Bauer, D. (1984a) Causes of flicker at VDUs with bright background and ways of eliminating interference. Teoksessa Grandjean, E. (toim.) Ergonomics and health in modern offices, 364–370. London and Philadelphia: Taylor & Francis.

Bauer, D. (1984b) Improving VDT workplaces in offices by use of physiologically optimized screen with black symbols on a light background: basic considerations. Behaviour and Information Technology, 3(4), 363–369.

Bauer, D. (1987) Use of slow phosphors to eliminate flicker in VDUs with bright background. Displays, 9, 29–32.

Bauer, D., Bonacker, M. ja Cavonius, C. R. (1983) Frame repetition rate for flicker-free viewing of bright VDU screens. Displays, 4 (1), 31–33.

Bauer, D., Cavonius, C. R. (1982) Improving the legibility of visual display units through contrast reversal. Teoksessa Grandjean, E., Vigliani, E. (toim.): Ergonomic aspects of visual display terminals, 137–142. London: Taylor & Francis.

Creed, A., Dennis, I., ja Newstead, S. (1987). Proof-reading on VDUs. Behaviour and Information Technology, 6(1), 3–13.

Gould, J., Alfaro, L., Finn, R., Haupt, B. ja Minuto, A. (1987) Reading from CRT displays can be as fast as reading from paper. Human Factors, 29(5), 497–517.

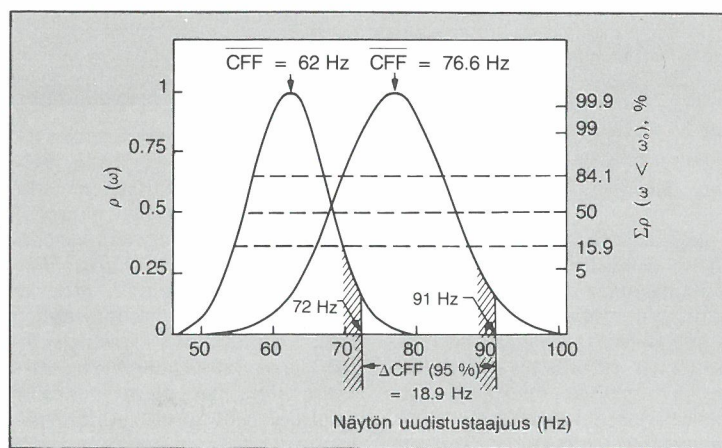
Laine, H. (1972) Ihmisen näköaistin tajuusominaisuudet ja niiden käyttö väsymykseen mittaamiseen. Diplomityö, Teknillinen korkeakoulu, Teknillisen fysiikan osasto, 48–50.

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

Laubli, T., Gyr, S., Nishiyama, K., Gierer, R. ja Grandjean, E. (1985) Effects of refresh rates of a simulated CRT display with bright characters on a dark screen. Tutkimusraportti. Department of Hygiene and Ergonomics, Swiss Federal Institute of Technology, Zürich, Switzerland.

Kuva 3. Reunaviivan aaltoilu kohdistetussa katselussa (vasemmalla), kohdistusliikkeiden aiheuttamat välähdykset (keskellä) ja ratkaisu ongelmiin (oikealla) (L_{su} / kuvaruudun reunaa alueen luminanssi, L_{sc} / kuvaruudun luminanssi ja d / tumman reunaa alueen leveys) (Bauer 1984a, 364–370).

Kuva 2. Kriittisen välkkymistajuuden jakaumat hitaalle (vasemmalla) ja nopealle (oikealla) fosforille 80 cd/mFD luminanssille (Bauer 1987, 29–32).



Ensimmäistä kertaa Nokia Datan historiassa tuotevalikoima on täydentynyt kokonaisella laitesarjalla. Uuteen MikroMikko 4 -sarjaan kuuluu kaksi MikroMikko 4TT-työasemamallia ja kaksi MikroMikko 4-palvelinmallia.

Muotojen takana voimaa

Vaikka uusien laitteiden silmiinpistävin ominaisuus on niiden muotoilu, suunnittelun perustana on ollut raaka suorituskky. Kaikki MikroMikko 4 -laitteet pohjautuvat Intelin 80386SX- tai 80386-teknologiaan, nopeisiin kiintolevyihin, kiintolevyjen ohjaimiin sekä nopeaan muistitekhnologiaan.

Korttipaikkojen säästämiseksi VGA-ohjain on kaikissa malleissa sijoitettu väyläkortille. Se on yhteensopiva myös kaikkien näyttövaihtoehtojen eli uuden positiivinäytön, värinäytön ja LCD-lattanaytön kanssa.

Käyttöjärjestelmäksi voi kaikkiin MikroMikko 4 ja 4TT -malleihin valita joko MS-DOS- (3.3 tai 4.01) tai Microsoft OS/2-käyttöjärjestelmän. MS OS/2 versio 1.1 vaatii työasemalta vähintään neljän megatavun muistia eli ilman muistinlaajennusta se ei kaikissa sarjan laitteissa suostu pyörimään.

MikroMikko 4 -sarjan suunnittelussa on helppokäyttöisyydelle pantu erityistä painoa. Esimerkiksi kortin lisääminen tai poistaminen onnistuu kaapeleita irrottamatta. Mikron takaosa on myös suojattu pölyltä, ja kaapeleille on ylä- ja alaosassa omat aukot. Kiintolevyt, 5,25 tuuman lisälevykeasema ja nauhavarmistimet ovat siististi oven takana. Tuoksena on siisti ja toimiva kokonaisuus.

Grafiikka pyörii — 4TT m326

Esimerkiksi Windows-ohjelma-perhe perustuu graafiseen käyttöliittymään, jonka sujuva toiminta vaatii mikrolta tavallista enemmän suorituskkyä. Niin numermurskausvoiman kuin massa-

90-luku voi alkaa

MikroMikko 4 on valmis

Katujen varsia ja lehtien sivuja ovat jo jonkin aikaa koristaneet uusien MikroMikkojen muhkeat muodot. Monelle on kuitenkin jäänyt epäselväksi, mitä on modernin muotoilun takana. Ovatko kyseessä keisarin uudet vaatteet vai ulottuuko uutuus myös pintaa syvemmälle?



Jarmo Hietaranta

ja käyttömuistinkin on riitettävä monimutkaisten graafisten hahmojen käsittelemiseen.

MikroMikko 4TT m326:n keskusyksikkönä pyörii 16 megahertsin taajuudella Intelin 80386SX-suoritin. Se vastaa sisäisesti 386-suorintaa, mutta ulospäin, esimerkiksi muistille ja oheislaitteille, se näyttää 286:lta. 80386SX pystyy siis täysin hyödyntämään 386-mikroille tehtyjä ohjelmistoja. 80386SX-suorittimeen pohjautuvia mikroja toimitetaan ennusteiden mukaan ensi vuonna jo yli kolmannes kaikista 386-mikroista.

MikroMikko 4TT m326 on tarkoitettu tavalliseen toimistokäyttöön henkilökohtaiseksi mikrosiksi tai lähiverkon työasemaksi. Vakiomallin työmuisti on kaksi megatavua, ja sitä voidaan laajentaa ilman lisäkorttia aina neljään megatavuun asti. Oheislaitteita varten laitteesta löytyy vakiona kaksi sarja- ja yksi rinnakkaisliitäntä. Sarjaliitännöjen säästämiseksi hiirelle on oma liitäntänsä. Vapaiksi jäävät neljä korttipaikkaa voi käyttää esimerkiksi muistin laajentamiseen 16 megatavuun sekä verkko- ja tietoliikenneliitännöihin.

4TT-työasemien kiintolevyjen hakuajat ovat keskimäärin 28 millisekuntia. Tallennuskapasiteettivaihtoehtoja on kaksi — 40 tai 70 megatavua.

Ammattilaisen työkalu — 4TT m336

MikroMikko 4TT m336 on työasema käyttäjälle, joka arvostaa laitteen pientä kokoa, mutta vaatii suorituskkyä vaativien sovellusten ajamiseen. Pienuudesta huolimatta MikroMikko 4TT m336 on täysin 32-bittinen kone, jonka suorittimena on 20 megahertsin taajuudella kelloitettu Intel 80386. Samalla nopeudella toimivalle aritmetikkasuorittimelle löytyy peruskortilta valmis kanta.

Työmuistia voi olla joko 2, 4, 8 tai 16 megatavua. Sarjaliitännöjä on neljä ja rinnakkaisliitäntä yksi. Kolme vapaata korttipaikkaa voidaan käyttää muihin laajennustarpeisiin.

Työryhmäverkon työjuhta — 4 m338

MikroMikko 4 -mallit on suunniteltu nimenomaan verkon palvelimiksi. Ne ovat ratkaisu myös sellaiselle käyttäjälle, joka haluaa ajaa mikrollaan suorituskkyä ja laajennettavuutta vaativia ohjelmistoja kuten erilaisia julkaisu-, grafiikka- ja CAD-sovelluksia.

Myös MikroMikko 4 m338 pohjautuu 20 megahertsin kello- taajuudella toimivaan Intelin 80386-suorittimeen. Sen työmuisti voi olla kooltaan 4, 8 tai 16 megatavua. Sarjaliitännöjä on neljä ja rinnakkaisliitäntä yksi. Korttipaikkoja on vapaana kaikkiaan viisi.

Kiintolevyvaihtoehtoja on kaksi: Kapasiteeteiltaan 150 ja 300 megatavua sekä hakuajoiltaan 23 ja 18 millisekuntia.

Laiteperheen lippulaiva — 4 m348

MikroMikko 4 m348 pohjautuu 25 megahertsin kello- taajuudella toimivaan Intelin 80386-suorittimeen. Samalla kello- taajuudella toimivan aritmetikkasuorittimen lisääminen on mahdollista, mutta lisäpotkua saa myös Weitekin 1167- ja 3167-aritmetiikkasuorittimilla.

Työmuistia voi olla 4, 8, 16 tai 24 megatavua. Toimintoja nopeuttaa 64 kilotavun välimuisti. Kuudesta vapaasta korttipaikasta on kaksi varattu muistin kasvattamiseen kahdeksasta 24 megatavuun ja neljä näytönohjaimelle, tietoliikenteelle, verkkoliitännälle tai muihin laajennuksiin.

Uudet uljaat näytöt — 11, 14 ja 15 tuumaa

Nokia Datan vanha positiivinäyttö on edelleen yksi markkinoiden parhaista yksivärinäytöistä, vaikka se on julkistettu jo vuonna 1984. MikroMikko 4 -laitesarjaa varten Nokia Data on kehittänyt uuden 15-tuumaisen yksivärinäytön. Se on toimintaperiaatteen laajennettu VGA-näyttö eli erottelukyky on parhaimmillaan 800x600-pistettä.



Ove Tammelin

Eri ohjelmistojen kuten Windows/286:n, Windows/386:n ja Venturan uuden näytön yhteydessä tarvitsemat laiteajurit toimitetaan laitteiston mukana.

Uudistustaajuus on laajennetussa VGA-tilassa jopa 99 hertsiä. Tavallisissa VGA-tiloissa saadaan täyskuva eli surureunoja ei näy. Myös uudessa 14 tuuman värinäytössä on samat ominaisuudet kuin mustavalkonäytössä. Molemmat näytöt ovat heijastamattomia, välkkymättömiä ja antistaattisia. Magneettinen ja elektrostaattinen säteily on alennettu minimiinsä.

Kolmas uusi näyttö on erikoiskäyttöön tarkoitettu 11 tuuman VGA-lattanayttö. Se on 6,4 cm syvä taustavalaistu nestekidenäyttö. Pienenä ja kevenä se on parhaimmillaan ahtaissa paikoissa. Lattanayttö soveltuu erityisesti teksti- ja tietokantasovelluksiin sekä kotikäyttöön.

Putkitongit ovat tarpeen

Henkilökohtaisen tietojenkäsittelyn lisääntyessä ja massamuistien yleistessä laitteilta vaaditaan täydellistä tietoturva. Myös verkon palvelimissa säilytetään korvaamattomia tietoja.

Jo MikroMikko 3TT m215:sta tutuksi tullut kolminkertainen salasanasuojaus on mukana koko MikroMikko 4 -laitesarjassa. Turvallisuutta on parannettu myös avainsuojauksella. Sekä työasemalla palvelinmalleissa voi koko kotelon lukita avaimella. Lisäksi työasemamalleissa on turvaketjurenkas. Millään putkitonkeja kevyemillä aseilla eivät asiattomat pääse tietoihin käsiksi. ■

MikroMikko 4:sten perhepotretti eri näyttövaihtoehtoineen.



SariFoto

Anneli Martonen toimii Nokia Datan tuotemerkkinäytössä mikrotietokoneista vastaavana tuotepäällikkönä.

Muistia millä mällätä

MikroMikko 4 ja isot muistit

Ajanuloon mikron käyttäjien nost on ilmestynyt useita erilaisia 640 kilotavun tuottoa puoleen sijoittavia apuohjelmia kuten työmuistilevykkeitä ja muistipuskureita. Vain se, joka on käyttänyt extended- ja expanded-muistinsa näitä ohjelmia, puoleen, kunnes 80286- ja MS-DOS:n osto-avaruuden katto on tullut vastaan. MikroMikko 4 ei ole ainoa ohjelma, joka tunnustaa

Työmuisti, jonka koko ylittää yhden megatavun, ei ole ollut kovinkaan käyttökelpoinen perinteisessä 80286-suorittimeen pohjautuvassa MS-DOS-ympäristössä. 80386-suorittimeen perustuvassa MS-DOS:issa tilanne on aivan toinen. Tapauksesta riippuen voidaan esimerkiksi neljän megatavun työmuistia hyödyntää kokonaan MS-DOS-sovellusten hyväksi, siitä huolimatta että perinteiset MS-DOS-sovellukset voivat osoittaa korkeintaan 640 kilotavun kokoista perusmuistia (conventional memory).

Seuraavassa käsitellään expanded-muistin käyttöä 80386-suorittimeen perustuvissa MikroMikko 4 -malleissa, joissa käyttöjärjestelmänä on MS-DOS 4.0 tai uudempi versio.

Expanded-muisti käyttöön

LIM/EMS 4.0 -määrittelyn mukaista expanded-muistin hallintaa voidaan toteuttaa eri tavoin, koska kyseinen määrittely koskee vain MS-DOS-sovelluksen ja EMM-laiteohjelman (EMM / Expanded Memory Manager) välistä ohjelmistotason liittymää, eikä lainkaan laitteistovaatimuksia.

Expanded-muistin käyttö edellyttää sekä laitteistoa (expanded-muistikortti) että ohjelmistoa 80286-suorittimeen perustuvissa mikroissa. Vastaava elektronikka sisältyy 80386-suorittimeen ja expanded-muistin käyttö vaatii 80386-ympäristössä vain tarkoitukseen sopivan ohjelman. 80386-suorittimen ansiosta voidaan samaa työmuistia käyttää eri tavoin muistinhallintaa ohjaavasta ohjelmasta riippuen.

MikroMikko 3TT m215 muodostaa poikkeuksen 80286-suorittimeen perustuvista mikroista siten, että expanded-muistin

käyttö edellyttää vain tarkoitukseen sopivaa ohjelmaa, kuten MS-DOS:iin sisältyvää EMM412.SYS:iä, eikä välttämättä erillistä expanded-muistikorttia.

Työmuistilevykkeitä ja muistipuskureita

Perusmuistia käyttävien varsin naisten MS-DOS-sovellusten lisäksi löytyy muitakin extended- ja/tai expanded-muistia käyttäviä ohjelmia, joiden käytöstä MS-DOS-sovellukset voivat hyötyä.

RAMDISK- ja RAMDRIVE-toiminnon avulla osaa työmuistista voidaan osoittaa ja käsitellä kuten kiintolevy-yksikköä. Työmuistilevyn sisältö katoaa kuitenkin automaattisesti kun virta katkaistaan koneesta tai käyttöjärjestelmä ladataan uudelleen. Työmuistilevyn käyttöä hallitseva laiteohjain on määriteltävä CONFIG.SYS-tiedostoon. Työmuistilevy voidaan sijoittaa perus- tai extended-muistiin. Sijainti määrittellään laiteohjelman määrittelyissä. Työmuistilevyn sijoitus perusmuistiin on tuskin toivottavaa, koska se pienentäisi MS-DOS sovelluksen käytettävissä olevaa muistitilaa.

HIMEM on ominaisuus, jota tukevat MS-DOS (versio 4.01:stä lähtien), Windows (versio 2.10) sekä LAN Manager. HIMEM:iä kutsutaan myös nimellä XMS tai HMA. HIMEM.SYS-laiteohjelma hallitsee ensimmäisiä 64:ää ensimmäisen megatavun ylittävää extended-muistin kilotavua. Tämän tuloksena HIMEM:in käyttö antaa MS-DOS-sovelluksen käytettäväksi ylimääräiset 64 kilotavua 640 kilotavun perusmuistin lisäksi. Tosin itse laiteohjelmakin vaatii hiukan muistitilaa. HIMEM-toiminnon käyttö voi olla ratkaisu muistipulmiin sellaisessa ympäristössä, jossa perusmuistin mää-

rä on toimivuuden kannalta kriittinen kuten verkkokäytössä.

SMARTDRIVE-laiteohjain (SMARTDRV.SYS) tukee kätkö-muistiominaisuutta (disk caching feature). MS-DOS-sovellukset, jotka toistuvasti suorittavat levyhakuja, voivat käyttää tätä laiteohjelmaa lukuaikojen lyhentämiseksi. SMARTDRV.SYS laiteohjelma on määriteltävä CONFIG.SYS-tiedostoon ja käytettävä muistikoko on käyttäjän määriteltävissä. SMARTDRIVE voidaan sijoittaa joko extended- tai expanded-muistiin laiteohjelman määrittelyistä riippuen.

Jos perusmuistin määrä on kriittinen, voi Windows/286 sekä jokin EMS-emulointiohjelmisto olla muistia säästävämpi vaihtoehto kuin Windows/386.

EMS-emulointiohjelmalla tarkoitetaan ohjelmaa, joka mahdollistaa extended-muistin käytön expanded-muistina LIM/EMS 4.0 määrittelyn mukaisesti. Markkinoilla on saatavilla useita erilaisia EMS-emulointiohjelmaa kuten 386Max.

FASTOPEN on MS-DOS-komento, joka nopeuttaa usein käytettävien levytiedostojen ja hakemistojen avauksia. Komento käyttää muistitilaa kyseisten toimintojen nopeuttamiseksi.

BUFFERS määrittelee levypuskureiden lukumäärän, jonka MS-DOS varaa muistista käyttöjärjestelmän latauksen yhteydessä. Puskureiden määrittely sisältyy CONFIG.SYS-tiedostoon. Levypuskurit voidaan kyseisen määrittelyn avulla sijoittaa joko extended- tai expanded-muistiin.

Lisäksi saattaa käytössä olla varsinaisessa perusmuistissa sijaitsevia memory-resident-ohjelmia, jotka kuluttavat samaa muistitilaa kuin sovellus ja käyttöjärjestelmä. Memory-resident-ohjelmat pysyvät muistissa MS-DOS-sovelluksen alla.

Expanded-muisti ja 386-koneet

Edeltävästä selostuksesta huomataan, että MS-DOS-sovellusten käyttöön on mahdollista varata muistitilaa myös ensimmäisen megatavun ylittävästä muistialueesta. MikroMikko 4 -malleissa expanded-muistin hallinta voidaan toteuttaa useiden vaihtoehtojen laiteohjelmien avulla.

Käytettävän laiteohjelman nimi on pääsääntöisesti määriteltävä CONFIG.SYS-tiedostossa. Expanded-muisti on käytettävissä kun kyseisen muistin hallintaa toteutettava laiteohjain on ladattu muistiin. CONFIG.SYS-tiedostossa määritellyt laiteohjelmat latautuvat muistiin MS-DOS-käyttöjärjestelmän yhteydessä.

Tässä yhteydessä tarkoitetaan Windows-sovelluksilla sellaisia sovelluksia jotka käynnistyvät ainoastaan Windows-ohjelman alaisuudessa. Vastaavasti muut kuin Windows-sovellukset voidaan käynnistää ilman Windowsia.

Windows/386 tarjoaa erinomaista expanded-muistin tukea 386-suorittimeen perustuvissa MikroMikko 4 -malleissa. Expanded-muistin koko määritellään WIN.INI tiedostossa. Tässä ympäristössä on kaikki expanded-muistia käyttävät sovellukset (myös muut kuin Windows-sovellukset) käynnistettävä Windowsista, koska expanded-muistin määrittelyt astuvat voimaan vasta Windows-ohjelman latauksen jälkeen. Windows/386

hyödyntää automaattisesti myös HIMEM-ominaisuutta, vaikka sitä ei ole määriteltävä CONFIG.SYS-tiedostossa. SMARTDRIVE- ja RAMDRIVE-laiteohjelmat on sen sijaan määriteltävä CONFIG.SYS-tiedostoon, mikäli niitä käytetään.

WDRV.SYS-laiteohjain on määriteltävä CONFIG.SYS-tiedostoon, jos expanded-muistia halutaan käyttää kokoonpanossa, joka sisältää SCSI-tyyppisen massamuistiohjelman.

Jos perusmuistin määrä on kriittinen toimivuuden kannalta esimerkiksi verkkoympäristössä, voi Windows/286 sekä jokin EMS-emulointiohjelmisto kuten 386Max olla muistia säästävämpi vaihtoehto kuin Windows/386. Käytettävä EMS-emulointiohjelma on määriteltävä CONFIG.SYS-tiedostossa. 386Max osaa hyödyntää HIMEM-piirrettä ilman vastaavaa määrittelyä CONFIG.SYS-tiedostossa. Tilanne voi olla toinen jossakin muussa EMS-emulointiohjelmassa.

Mikäli käytetään MS-DOS:iin sisältyvää expanded-muistin laiteohjelmaa muiden laiteohjelmien sijasta, on vastaava laiteohjelma (EMM386.SYS) määriteltävä CONFIG.SYS-tiedostoon.

Kokemusta ja näkemystä

Muistin käyttö- ja määrittelymahdollisuudet MS-DOS-ympäristössä ovat monipuolistuneet. Tämä helpottaa muistin käytön optimointia. Koska käyttöympäristö muodostaa kokonaisuuden, on huomattava, että muistin käytön hallinta vaatii enemmän kuin yksittäisen sovelluksen tuntemusta. Muistin käytön optimointiin on olemassa hyviä työkaluja, mutta niiden käyttö edellyttää kokemusta ja kokonaisnäkemystä. ■

Xerox Ventura Publisher 2.0

Ventura laajeni uusiin ympyröihin

Venturasta on nyt kolme eri versiota: Perusversio, Professional Extension, jossa on joitakin lisäominaisuuksia sekä Network Server, jossa on verkkokäytössä tarvittavat tiedostojen suojaukset.

Ohjelmien laajentuessa myös niiden laitteistovaatimukset kasvavat. Venturan perusversiota voi käyttää vaikka lisämuistia ei olisi. Tällöin CONFIG.SYS-tiedostossa ei voi olla DEVICE = -määrittäjiä, paitsi DEVICE = HIMEM.SYS, jos HIMEM.SYS on käytettävissä.

Ventura osaa hyödyntää LIM/EMS 4.0:n mukaista lisämuistia, joka on todella tarpeen työskentelyn sujuvuuden takia. Perusversio osaa sijoittaa osan ohjelmakoodista lisämuistiin, ja Professional Extension tallentaa sinne myös käsiteltävän julkaisun tietoja. Suuren kokonsa ja monipuolisten ominaisuuksiensa takia Ventura on parhaimmillaan MikroMikko 4:n kaltaisissa tehokkaissa mikroissa.

Kehittynyt käyttöliittymä

Vaikka käyttöliittymän perusteet ovat pysyneet ennallaan, käyttö-mukavuutta on pystytty parantamaan pienillä muutoksilla. Nyt Venturaan kuuluu yli 300 avustusta. Apua saa osoittamalla avustuskon dialogi-ikkunassa. Tällöin saa tietoja kyseisestä toiminnosta ja sen käytöstä.

Venturan kakkos-versioon on lisätty uusia ominaisuuksia, joita kaivattiin jo versiossa 1.1, kuten viritetty käyttöliittymä, 300 apuruutua ja kyky hyödyntää lisämuistia.

Muutoksista huolimatta Ventura 1.1:n käyttäjä osaa käyttää Ventura 2.0:aa ilman opetusta.

Kaikki eivät pitäneet Ventura ykkösen itseksesi esiin putkahtavista valikoista. Nyt käyttäjä voi valita joko Windows-tyyliset alas vedettävät valikot, jotka avautuvat, kun valikon nimen päällä painetaan hiiren nappia, tai valikot, jotka avautuvat, kun hiiri osoitin osuu valikon nimen päälle. Näin käyttäjä voi toimia totuttamalla tavalla.

Keskustelulaatikoihin on lisätty valikoita, joista valitaan sallitut arvot. Tällainen valikko on esimerkiksi paperin koon valinta helpottamassa. Samalla tavalla voidaan valita käytettävät mittayksiköt.

Kuvat mukaan väreissä tai ilman

Julkaisuohjelmissa kuvien liittäminen tekstiin on keskeisellä sijalla. Versio 2.0:n myötä Ventura osaa käsitellä useita uusia kuvatiestotyyppiejä. Lisäksi Ventura osaa käsitellä värejä ja harmaansävyjä.

Ventura 2.0:n dokumenteissa voidaan käyttää Microsoft Windows -ympäristössä toimivien ohjelmien kuten Designerin ja Microsoft Excelin kuvia, kun kuvat tallennetaan Windows Metafile -muodossa (WMF). Jos ohjelma ei osaa tallentaa kuvaa WMF-formaatissa, voidaan käyttää Venturan mukana tulevaa CLIP 2VP.EXE-ohjelmaa, joka osaa ottaa Windowsin leikepöydältä (clipboard) kuvat ja tallentaa ne WMF-muodossa, minkä jälkeen ne voidaan lukea Venturaan.

Harmaansävyjen käsittelyn ansiosta kuvanlukijalla luetut valokuvat saadaan tulostettua tavallisella 300 dpi:n (dots per inch — pistettä tuumalle) laserkirjoittimella kohtuullisen hyvin. Erottelutarkkuudeltaan parempaa tulostinta käytettäessä sävykuvat saadaan tulostettua todella hyvän näköisinä, eikä kuvia tarvitse liittää painossa erikseen julkaisuun.

Sävykuvat siirtyvät Venturaan TIFF-muodossa (Tagged Image File Format), joka pitää sisällään harmaansävyinformaation. Harmaansävyt saadaan näkymään näytöllä kun käytetään VGA- tai

EGA-näytönohjainta, ja järjestelmästä löytyy lisämuistia. Lisäksi näyttöjen valmistajat ovat tehneet näyttöjensä varten Venturalle erityisiä näytönohjaimia, joilla harmaansävyt saadaan käyttöön.

Lisää erilaisia kirjasintyyppiejä

Kun käytetään Nokia 1000 tai muuta HP LaserJet II -yhteensopivaa laserkirjoitinta, Ventura sisältää kaksi kirjasinleikkaukseriä (Swiss ja Dutch). Nämä toteuttavat sen julkaisujärjestelmän vähimmäisvaatimuksen, että käytössä on eri kokoja sekä päätteellisistä että päättömistä kirjasimista. (Esimerkiksi Softarin leipätekstin tekstityyppinä on English Times, jossa on päätteet, ja väliotsikot ovat HELIOS Extra Bold II:Sta, jossa ei ole päätteitä.)

HP LaserJet II:n peruskirjasintyypeistä puuttuvat kuitenkin isot merkkikoot (24 pistettä). Näitä voidaan tehdä pakettiin kuuluvalla Bitstream fontware -ohjelmalla. Venturan toimituspaketissa olevien Dutch-, Swiss-, ja Courier 10 roman -kirjasintyyppien lisäksi on mahdollista hankkia muita tyyppiejä, joita on saatavilla useita kymmeniä.

Kun käytetään PostScript-kirjoitinta, valittavissa on automaattisesti useita kirjasintyyppiejä, jotka riittävät hyvin normaaliin julkaisuutoimintaan. Post-

Scriptin kirjasintyytit ovat luonnollisesti yhteensopivia PostScript-kieltä ymmärtävien latomakoneiden tyyppien kanssa, joten julkaisun muokkauksen voi tehdä mikrolla, vaikka lopullinen tulostus tehtäisiin kirjapainossa.

Kaikki PostScript-kirjoittimet eivät välttämättä osaa tulostaa kaikkia kirjasintyyppiejä, mutta

Times-kirjasintyyppin anatomiaa.



Ventura tulostaa korvaavan kirjasintyyppin merkit oikean kirjasintyyppin leveydestä mukaisesti, jolloin teksti vie paperilla oikean tilan. Taitto voidaan siis edelleen tehdä mikrolla, vaikka tulos niin näytöllä kuin laserilla-kin näyttää liian tiheältä tai väljältä, sillä latomakoneen kirjasintyyppiä on huomattavasti paljon suurempi.

Ventura 2.0:n PostScript-kieli on edelleen parantunut versiosta 1.1 eli se pystyy kuvaamaan asioita, jotka eivät ennen välttämättä onnistuneet.

Ammattilaisen laajennus

Harvoin tarvittavia mutta silloin välttämättömiä ominaisuuksia on koottu Professional Extensioniin. Näitä ovat muiden muassa matemaattisten yhtälöiden kirjoittaminen ja taulukoiden laatiminen. Lisäksi Professional Extensionissa on, kuten alussa mainittiin, laajennettu lisämuistin tuki.

Matemaattiset lausekkeet, joissa voi käyttää erikoismerkkejä kuten neliöjuurta, summausmerkkiä ja integraalia, kirjoitetaan erikseen lausekkeen muokkaajalla, jossa erikoismerkeillä on omat nimensä. Kun tekstiin lisätään lauseke, Ventura lisää automaattisesti rivinvälä etteivät tekstit mene päällekkäin.

Taittoa suomeksi tai englanniksi

Ventura 2.0 on saatavilla suomen- ja englanninkielisenä. Suomenkielinen paketti sisältää englanninkielisen Venturan, suomenkielisen käsikirjan ja suomenmennusohjelman, jolla asennettu Ventura käännetään suomenkieliseksi.

Tällä hetkellä Professional Extension on saatavilla vain englanninkielisenä, mutta senkin suomenmennus valmistuu lähiaikoina. Network Server -tuotetta ei Suomesta vielä saa.



Jari Luukkala on perehtynyt julkaisuohjelmiin Nokia Datan tuotekehityksen ohjelmistotryhmässä.

Puhuu suomea ja ymmärtää hiirtä

Uusi WordPerfect 5.0 on entistä kehittyneempi tekstinkäsittelyohjelmia, joka avaa käyttäjälleen aivan uusia mahdollisuuksia tuottaa ulkoasultaan viimeisteltyjä tekstejä.

Uusi suomenkielinen WP 5.0 esiteltiin ensimmäisen kerran julkisesti Helsingin Messukeskuksessa ELTEK 89 -näyttelyn yhteydessä huhtikuussa. Softarin ilmestyessä toimituksien olisi jo pitänyt alkaa ja mukana pitäisi olla myös Suomessa WordPerfectille ohjelmoitu hiiri-ohjaus.

Uuden WP:n käyttöliittymä on samankaltainen kuin WP 4.2:ssa. Komennot annetaan toimintonäppäimillä ja valikoista kuten ennenkin. Uudessa WP:ssä ovat mukana myös kaikki WP 4.2:n ominaisuudet, joista osaa on laajennettu ja parannettu. Uusi WordPerfect lukee tietenkin myös WP 4.2:n tiedostoja ja tallennuksen voi tehdä WP 4.2 -muodossa. Mukana on myös joukko uusia ominaisuuksia.

Grafiikkaa kehään

WordPerfectin avulla voi nyt liittää myös grafiikkaa tekstin lomaan. WP 5.0 osaa lukea yleisimpiä grafiikkatiedostoja, esimerkiksi PIC- (Lotus 123 ja Symphony), PCX- (PC Paintbrush), HPGL-, EPS- ja MSP-tiedostotyyppisiä. Lisäksi WP:n mukana tulee ohjelma, jolla näytöltä voi siepata grafiikkaa (screen capture) eli tarvittaessa kuvat siirtyvät lähes ohjelmasta kuin ohjelmasta.



Uudessa WP:ssä grafiikkaa voi myös käsitellä eli esimerkiksi pyörittää tai venytellä (suurentaa ja pienentää erikseen sekä vaaka- että pystyasossa) ja siihen voi liittää kuvatekstin. Kuvan paikan tekstin suhteen voi tietenkin määrätä erikseen siinä missä tekstin saa ohjattua kiertämään kuva-alueen.

Älyä tulostukseen

WP 5.0 hallitsee omaa tulostustaan. Asiakirjaa kirjoitettaessa ei käytettävän kirjoittimen ominaisuuksia tarvitse enää ottaa huomioon. Tekstin koko voidaan valita kuudesta eri vaihtoehdosta: hieno, pieni, normaali, suuri, hyvin suuri ja erittäin suuri. Tekstin osia voidaan myös korostaa tulostusasua muuttamalla. Vaihtoehtoja on seitsemän: puolilihavoitu, alleviivattu, kaksoisalleviivattu, kursivoitu, ylivuivattu, varjostettu ja kapiteelit (pienaakkosten korkuiset versaalimerkit).

Lisäksi saadaan vielä määriteltä jokin perusmerkkilaji, jolloin voidaan hyödyntää laserkirjoittimissa olevia lukuisia kirjaisintyyppisiä (fontit) kuten Courier, Times Roman, Helvetica jne.

Tulostaessaan WP valitsee aina merkkilajin vaihdon tai tehosteen vaihdon yhteydessä kirjoittimen merkkilajeista sopivimman ja tulostaa sillä. Eri kirjoittimilla lop-

putulos on tietysti erilainen, sillä esimerkiksi laserkirjoittimet tulostavat lähes poikkeuksetta kaikki koot ja ulkoasut, kun taas kiekkokirjoittimet tulostavat vain yhtä kokoa vahvennettuna, alleviivattuna tai ylivuivattuna.

Älykkäästä tulostamisesta on sen selkeyden lisäksi vielä se hyöty, että asiakirjoja ei tarvitse muokata uudelleen jos halutaan tulostaa eri kirjoittimilla, sillä lopputulos on aina niin hyvä mihin kulloinkin kirjoitin pystyy.

Merkit vs. pisteet

Edellisessä WP:ssä sivun ulkoasu määriteltiin perinteisillä konekirjoittajan mitoilla eli merkeillä ja riveillä. Tämä on kuitenkin hankalaa kun merkit ja rivit saattavat eri tilanteissa olla eri kokoisia. Nyt kaikki mitat voidaan vaihtoehtoisesti määrätä usealla eri tavalla — myös absoluuttisesti. Vaihtoehdot ovat: WP 4.2 -mitat (rivit/merkit), tuumat, senttimetrit, pisteet (1/72 tuumaa) ja WP 5.0 -mitat (1/1200 tuumaa).

Absoluuttisten mittojen käytöstä on se hyöty, että enää esimerkiksi marginaalit ja sarkaimet eivät vaihda paikkaa, kun merkkien koko tai laji vaihtuu.

Sivun koko voidaan valita suoraan eri standardeista kuten A4, A4 Landscape, Legal, US Government jne.

WordPerfect 5.0 avaa uusia uria

Nyt WP:ssäkin on mahdollista käyttää niin kutsuttuja tyyli-tiedostoja. Tyylit automatisoivat ja helpottavat muotoilua. Ne myös vähentävät muotoiluun kuluva aikaa ja yhtenäistävät asiakirjojen ulkoasua. Kun tyyli on kerran saatu kohdalleen, voidaan sitä käyttää toistuvasti muotoilematta jokaista yksityiskohtaa erikseen.

WP 5.0:n lisäksi voi halutessaan tilata kohtuullista korvausta vastaan erillisen BITSTREAM softfont -paketin HP Laserjet -yhteensopiville ja PostScript-kirjoittimille. Paketti sisältää Times Romanin ja Helvetican näköiset (look alike) kirjaisintyytit. Tyyppien koot ovat käyttäjän itse määriteltävissä kuudesta pisteestä ylöspäin ja mahdollistavat suurienkin merkkien käsittelyn.

Sen saat mitä näet

Jo WP 4.2:ssa oli mukana esikat-selu, mutta vasta uudessa WP 5.0:ssa esikatselu on todella hyödyllinen. Asiakirjaa voi katsella näytöllä grafiikkoinen, eri kirjaisintyyppineen ja tehosteineen. Asiakirja näkyy näytöllä juuri sellaisena kuin se tulostuisi kirjoittimellekin eli jos kirjoittimesta puuttuu jokin ominaisuus, sitä ei näytetä näytölläkään. Tämä vähentää huomattavasti uudelleen tulostamisen tarvetta.

Muuta mukavaa

Uudessa WP:ssä asetukset ja oletusarvot voidaan asettaa suoraan ohjelmasta eikä käynnistämällä ohjelma määrättyllä valitsimella kuten edellisissä versioissa. Kaikki oletusarvot kuten sivun koko, marginaalit, sarkainasetukset jne. tallentuvat aina automaattisesti asiakirjan mukana eli asiakirjojen siirto käyttäjältä toiselle tai oletusarvojen muutos on vaivatonta.

WP 5.0:n makrojen teko on helppoa. Tavanomaisen näppäinten painalluksien tallentamisen lisäksi makroja voi tehdä erityisellä makrokielellä. Makrokielen tehokkudesta saa hyvän kuvan tarkastelemalla uuden WP:n mukana tulevaa taskulaskimen toteuttavaa makroa.

Lisäksi joihinkin WP:n yleisimmin käytettyihin toimintoihin kuten alueiden siirtämiseen, poistamiseen ja kopioimiseen, tiedostonhallintaan sekä etsi/korvaa-toimintoon on tehty hiirelle omia oikopolkuja.

Kun WP käynnistetään hiirituella, WPHIIRI-ohjelma lataa itsensä taustalle ja käynnistää itse WP-ohjelman. Kun WP lopetetaan, WPHIIRI-vapauttaa varaimansa muistitilan eikä jää roikkumaan keskusmuistiin kuten jotkut huonosti suunnitellut apuohjelmat.

Lisämausteena WPHIIRISSÄ on oma HIIRIHELPPI.

Hiiren oma kieli

Koko hiiriohjausohjelmiston kehitystyö alkoi jo viime vuoden kesällä, ja koska markkinoilla ei ollut sopivaa yleiskäyttöistä ohjelmistoa hiiriohjauksen tekemiseen, se päätettiin tehdä itse.

Hiiriohjausohjelmistoa varten suunniteltiin oma korkean tason kieli Nokia Mouse Menu ja sille tehtiin oma kääntäjä (omaa käyttöä varten, ei myytäväksi). Nokia Mouse Menu on täysin yleiskäyttöinen, ja sillä voidaan tehdä hiiriohjaus suureen osaan sellaisia ohjelmia jotka eivät osaa itse hyödyntää hiiriohjausta. WordPerfect on ensimmäinen Nokia Mouse Menun avulla viritetty ohjelma, mutta lisää hiiriohjaustukea ainakin Perfect-perheelle on odotettavissa.

Idea hiiriohjauksen tekemiseen syntyi Nokia Datan tuotekehityksen ohjelmistot -ryhmässä. Nokia Mouse Menun suunnittelu ja toteutus sekä WordPerfect-tekstinkäsittelyn hiirituki ovat alikirjoittaneen käsialaa, mutta myös ohjelmistoasiantuntija Vesa Laatikainen on antanut arvokkaan työpanoksen erityisesti hiiriohjauskehittämisen kehitysvaiheissa. ■

C++-ohjelmointikieli

Pureeko se?

Ei ole kauan siitä, kun C-kielestä povattiin seuraavaa johtavaa yleiskäyttöistä ohjelmointikieltä. C:tä mainostettiin monipuoliseksi, siirrettäväksi ja tehokkaaksi kieleksi, jonka käyttöalue ulottuu mikroista suurimpiin keskustietokoneisiin asti.

C-kielellä voi 'kääntää bittejä' yhtä hyvin kuin harrastaa strukturoitua ja modulaarista ohjelmankehitystä. Miksi siis tarvitaan jälleen uusi ohjelmointikieli, vai tarvitaanko sitä lainkaan?

Oikeastaan C++-kielestä ei pidä puhua kokonaan uutena ohjelmointikielenä, vaan C-kielen laajennuksena. Muutamaa poikkeusta lukuunottamatta hyvää ohjelmointityyliä käyttäen kirjoitetut C-ohjelmat kääntyvät C++-kääntäjällä selaisenaan tai pienin muutoksin.

C-kielen piirteiden lisäksi C++ tarjoaa joukon lisäominaisuuksia, joita voi opetella hallitsemaan yhden kerrallaan ja näin helpottaa siirtymistä C++-kieleen.

Tärkeimmät uudistukset koskevat määrittelyjä. C-kielen syntaksi tukee strukturoitua ohjelmointia ja abstrakteja tietotyyppiä, mutta jää puolitiehen. C++ tarjoaa joukon uusia abstrahointityökaluja ja vahvan tyyppitarkistusmekanismin, joiden avulla ohjelmakoodin selkeyttä, oikeellisuutta ja uudelleenkäytettävyyttä voidaan parantaa. Nämä ovat samoja ominaisuuksia, joita on pidetty muiden uuden sukupolven kielten, kuten Modula-2:n ja Adan, vahvoina puolina.

C++ poikkeaa kuitenkin ratkaisevasti edukseen sinä suhteessa, että se on ylöspäin yhteensopiva paljon käytetyn C-kielen kanssa eikä tyydy pelkästään muistuttamaan jotain vahempaa kieltä, kuten Modula-2 tai ole kokonaan uusi kieli, kuten Ada.

Näiden ominaisuuksien lisäksi C++ tukee voimakkaasti oliokeskeistä ohjelmointia (object oriented programming), joka on nousemassa strukturoidun ohjelmoinnin ohi tärkeimmäksi ohjelmointityyliksi. Oliokeskeisiä kieliä on toki olemassa muitakin, mutta näillä näkymin todella vakavasti otettavaa haastajaa C++:lle ei ole näköpiirissä.

Tärkeimpiä haastajia ovat Smalltalk ja Objective-C. Smalltalkin suurin ongelma on samalla sen vahvin puoli: se on erittäin suuri ja raskas, täysin suljettu tulkaava ohjelmointiympäristö. Sillä on helppo kokeilla oliokeskeisiä tekniikoita, mutta se soveltuu em. seikoista johtuen huonosti varsinaiseen tuotantokäyttöön. Objective-C on C:stä ja Smalltalkista kehitetty hybridi, jossa oliokeskeinen syntaksi on lainattu Smalltalkista ja muu ohjelmointi tapahtuu C:n tapaan. Objective-C:n syntaksia ei voi pitää täysin onnistuneena, eikä se

ole saavuttanut sellaista suosiota kuin C++.

Data ja funktiot

Oliokeskeisessä ohjelmoinnissa on olennaista tiedon esitysmuoto, jossa data ja sen käsittelemistä varten oleva ohjelmakoodi yhdistetään yhdeksi paketiksi, josta käytetään nimitystä olio. C++-kielessä olion käsitettä vastaa luokka. Luokkamäärittely on uusi, ohjelmoijan määrittelemä tietotyyppi, johon liittyy sekä data että sen käsittelyfunktio.

Luokat mahdollistavat tiedon piilottamisen ulkopuolisilta, olioiden automaattisen alustamisen, implisiittiset tyyppimuunnokset, operaattoreiden määrittelyt, muistinhallinnan jne. Oleellista on se, että luokan määrittelyn jälkeen sen käyttö ei mitenkään poikkea sisään rakennettujen tyyppien käytöstä, olipa määritelty tyyppi miten monimutkainen tahansa.

Luokka on itse asiassa suora laajennus C-kielen tietuekäsitteeseen (struct). C:n tietueissa on ainoastaan datakenttiä. Sen sijaan luokkiin saa (ja pitää) määritellä myös funktiot, joiden avulla datakenttiä käsitellään. Kukin näistä määrittelyistä voidaan piilottaa muilta kuin luokkaan kuuluvilta olioilta. Hyvin suositeltavaa on piilottaa kaikki datakentät ja tarjota ulospäin rajapinta, joka muodostuu joukosta funktioita.

Esimerkissä 1 määritellään uusi tietotyyppi complex ja em. tyyppiä oleva muuttuja c. Tyyppiin complex kuuluu kaksi piilotettua datakenttää re ja im, jotka molemmat ovat double-tyypisiä. Avainsanaa public seuraavat määrittelyt ovat julkisia eli myös ulkopuolisten käytettävissä.

Sisäisiä funktioita

Funktio complex, real ja setimag ovat nk. jäsenfunktioita. Tässä esimerkissä funktiolla real saadaan selville reaali- ja imaginaarikomponentin arvo ja funktiolla setimag voidaan asettaa imaginaarikentälle uusi arvo. Jäsenfunktioita ovat ainoita funktioita, joilla on lupa käsitellä luokan piilotettuja kenttiä. Ne on aina esiteltävä luokkamäärittelyn yhteydessä. Niiden

varsinainen määrittely eli runko voidaan antaa joko esittelyn yhteydessä (complex ja setimag) tai erikseen (real). Ensiksi mainitussa tapauksessa funtioista tulee inline-funktioita eli niiden koodi korvautuu jokaisen kutsun tilalle eikä varsinaista funktiokutsua suoriteta lainkaan. Jälkimmäisessä tapauksessa funktioista tulee todellisia, ellei niiden määrittelyä edellä avainsana inline.

Funktio complex on erikois- asemassa: kaikki funktiot, joilla on sama nimi kuin luokalla, johon ne kuuluvat, ovat konstruktoreita, joiden avulla kyseistä tyyppiä olevat muuttujat alustetaan. Esimerkissä muuttujan c reaali- ja imaginaarivälikomponentit asetetaan 2.0 ja 1.5.

Jäsenfunktioiden kutsu tapahtuu C-kielestä tuttua tietueen kentänviitauksella eli pistettä käyttäen. Lauseke c.real() tarkoittaa muuttujan c reaali- ja imaginaarivälikomponentin kutsua. Tämä funktio palauttaa kutsujan rekentän arvon. Jäsenfunktioilla voi olla myös parametreja, kuten esimerkimmme setimag-funktiolla. Lauseella c.setimag(-2.0); asetetaan muuttujan c imaginaarikomponentille uusi arvo -2.0.

Valta ja muunnokset

C++-Kielessä funktion esittelyyn kuuluu aina sekä paluuarvon että parametrien tyyppilista. Tällä varmistetaan, että funktioita kutsutaan vain oikean tyyppisillä parametreilla. Toinen käyttö parametrilistalla on implisiittisen eli kääntäjän automaattisesti suorittamien tyyppimuunnosten yhteydessä.

Esimerkin 2 tapauksessa funktio abs olettaa saavansa tyyppiä long olevan parametrin, mutta sitä kutsutaankin int-parametrilla. Tässä tapauksessa kääntäjä ei anna virheilmoitusta, vaan muuntaa annetun int-parametrin ennen funktiokutsua long-tyypiksi. Tästä käytetään nimitystä implisiittinen tyyppimuunnos. Toinen muunnos suoritetaan funktion paluuarvolle: funktio abs palauttaa long-arvon, mutta se sijoitetaan int-muuttujaan, joten se on ensin muunnettava.

Kääntäjä ei muunna mitä tahansa mielivaltaisia tyypejä toi-

siksi. Pääsääntö on, että sisäänrakennetut numeeriset tyytit voidaan aina muuntaa toisiksi. Tiettyissä tapauksissa muunnoksen tuloksena menetetään tarkkuutta tai saadaan virheilmoitus. Funktiokutsut, joissa parametreja on annettu joko liikaa tai liian vähän, kääntäjä luonnollisesti hylkää aina.

Ylimääräistä kuormaa

Koska funktion tyyppi määräytyy sekä paluuarvon että parametrien tyyppien perusteella, voidaan funktiot ylikuormittaa (overloading) eli määritellä useita funktioita, joilla on sama nimi, mutta erilaiset parametrit. Niiden paluuarvot saavat olla joko samaa tai eri tyyppiä. Itse asiassa näillä funktioilla ei kielen määrittelyn kannalta ole mitään muuta yhteistä kuin nimi. Käytännössä ei tietenkään kannata määritellä täysin eri funktioita samannimisiksi, vaan tervien järjen käyttö on paitsi sallittua myös suotavaa.

Esimerkissä 4 esitellään kolme abs-nimistä funktiota, jotka todennäköisesti kaikki palauttavat parametrinsa itseisarvon. Ylikuormittaminen on sallittua ainoastaan, mikäli nimi on ensin määritelty ylikuormitettavaksi overload-määrittelyllä. Tällä pyritään estämään tahattomat nimien uudelleen käyttämiset, jotka mahdollisesti johtaisivat ohjelmointivirheisiin. Seuraavassa C++-kielen versiossa (2.0) overload-avainsanaa ei enää tarvitse käyttää.

ANSI-C-kielessä, johon C++-tyyliset esittelyt on otettu mukaan, tyyppilistaa käytetään varmistamaan, että funktion kutsussa käytetään oikeita parametreja. Tyyppilistan avulla kääntäjä valitsee samannimisista funktioista oikean, mikäli sellainen on olemassa tai antaa virheilmoituksen, jos sopivaa funktiota ei ole määritelty.

Edellisessä kappaleessa esitetyt tyyppimuunnossäännöt tuovat tähän omat vaikeutensa: esim. int-tyyppinen parametri voidaan muuntaa joko long- tai double-tyypiseksi, joten kääntäjällä ei ole mahdollisuutta tietää, kumpaa funktioista abs(long) ja abs(double) ohjelmoija haluaa kutsua, jos hän kirjoittaa esimerkiksi abs(-2). Näissä tapauksissa ohjelmoijan on suoritettava tyyppimuunnos eksplisiittisesti.

Oletetaan, että ...

On mahdollista määritellä funktioita myös siten, että funktiokut-

sussa kaikkia parametreja ei tarvitse antaa, vaan puuttuvien tilalla käytetään oletusparametreja.

Esimerkissä 4 muutettiin luokan complex konstruktorifunktiota siten, että se ottaa 0, 1 tai 2 double-tyyppistä parametria. Mikäli parametreja ei anneta, tulee sekä re- että im-kentän arvoksi 0.0. Jos annetaan vain yksi parametri, tulee se re-kentän arvoksi ja im-kentän tule 0.0. Jos halutaan käyttää oletusarvoja, on funktioiden parametrilista suunniteltava huolella, sillä oletusparametrien on aina oletava viimeisenä listassa.

Operaattoreita tai sitten ei

Operaattorit ovat C++-kielessä funktioita. Näin ollen ne voidaan myös määritellä itse luoduille tyypeille ja tarvittaessa ylikuormittaa.

Esimerkissä 5 lisättiin luokan complex määrittelyyn kaksi miinus-operaattoria. Ensimmäinen on binäärioperaattori eli se tarvitsee kaksi operandia ja toinen on unäärioperaattori eli se ottaa ainoastaan yhden operandin. Binääriminus palauttaa operandinsa erotuksen, unääriminus puolestaan operandinsa vastaluvun. Lähes kaikki operaattorit on mahdollista määritellä ja ylikuormittaa itse määrittelyille tyypeille.

Sisäänrakennettujen tyyppien operaattoreita ei saa määritellä uudelleen eikä uusia operaattorisymboleita voi luoda. Esimerkin pääohjelmassa luodaan kolme complex-muuttujaa ja käytetään näitä operaattoreita. Kaksinkertaisen kauttaviivan (//), joka C++:ssa toimii kommenttien alkumerkkeinä, jäljessä on vastaava funktiokutsu muotoon: c.setimag(-2.0);

Perunkirjoitus

Oliokeskeisessä ohjelmoinnissa hyvin keskeisellä sijalla on myös perintä. Se tarkoittaa tyyppimäärittelyä, jonka osana on jokin olemassa oleva tyyppi, jolta uusi olio perii ominaisuuksia. C++-kielen nykyisessä versiossa on toteutettu ainoastaan yksiperintä, jossa olio voi periä ominaisuuksia ainoastaan yhdeltä toiselta oliolta. Tällöin syntyvästä nk. oliohierarkiasta tulee puumainen. Seuraaviin C++-versioihin on tulossa myös moniperintä. Moniperintä on teoreettisesti vaikea ja jopa epäsuotava piirre, mutta käytännössä se on osoittautunut hyödylliseksi. ■

Esim 1. Määritellään luokka complex, jonka avulla käsitellään kompleksilukuja:

```
class complex {
    double re, im;
public:
    complex(double r, double i) {re = r; im = i;}
    double real();
    void setimag(double i) {im = i;}
};

double complex::real()
{
    return re;
}

main()
{
    complex c(2.0, 1.5);
    double rea;
    rea = c.real();
    c.setimag(-2.0);
}
```

Esim 2. Tyyppimuunnos

```
long abs(long);
int i = abs(-3);
```

Esim 3. Funtioiden ylikuormittuminen

```
overload abs;
long abs(long);
double abs(double);
double abs(complex);
```

Esim 4. Oletusparametrit

```
class complex {
    double re, im;
public:
    complex(double r = 0.0, double i = 0.0)
    {re = r; im = i;}
    ...
};
```

Esim 5. Operaattorien määrittely

```
class complex {
    ...
    complex operator-(complex);
    complex operator-();
    ...
};

inline complex complex::operator-(complex c)
{
    return complex(re - c.re, im - c.im);
}

inline complex complex::operator-()
{
    return complex(re, -im);
}

main()
{
    complex c1(-1.0, 2.4), c2(3.5), c3();
    c3 = c1 - c2; // c3 = c1.operator-(c2);
    c3 = -c3; // c3 = c3.operator-();
}
```


Learning Ada

Part 4

Part 3 dealt with the basic data types of Ada. Here we will see how data types are derived from existing types, and look at the differences between types and subtypes. We will also go a little deeper into array types, record types, and access types.

In Part 3 we said that the programmer does not generally use predefined data types, but defines his or her own types basing on these. This is done by type derivation, in which the programmer specifies a new type whose values are to be kept separate from other types.

Derived Types

A derived data type resembles an existing type in most respects but it is nevertheless a distinct type. Data types are derived from existing ones using the reserved word **new**:

```
type TEMPERATURE__TYPE is new INTEGER;
type WEIGHT__TYPE is new FLOAT range 0.0 .. 1000.0;
```

A derived type inherits its values and operations from the parent type. The range of values can be limited by a **constraint** as in the definition of **WEIGHT__TYPE**. New types can be derived either from predefined or user-defined types.

The values of a derived type are not compatible with the parent type, but a **conversion** between the two types is always possible in both directions:

```
I : INTEGER := 10; -- I is initialized to 10
L : TEMPERATURE__TYPE := 20;
...
I := L; -- ERROR!
-- Illegal assignment; types are not the same!
I := INTEGER(L);
-- Type conversion is possible between the types.
```

Type derivation is an easy way to define new types basing on other types. We can separate our own types from the predefined types, and thus enable the compiler to discover errors. It is impossible to mix values of two different types accidentally, because the conversion has to be made explicitly. But whenever you find a lot of type conversions in your program, you should consider revising your whole type system.

Types and Subtypes

We have repeatedly stressed the strict boundaries between separate types. Strict boundaries are necessary, but we often need to constrain an existing type without creating a new type. For example, we can define the number of floors in a building as a normal integer whose value lies between 1 and 12. **FLOOR__TYPE** is an example of a **subtype**:

```
subtype FLOOR__TYPE is INTEGER range 1 .. 12;
```

FLOOR__TYPE is not a new type. It is only a new name for the type **INTEGER** with its values limited by a **range constraint** to 1 .. 12. **FLOOR__TYPE** is obviously a better name for a type than **INTEGER**. Note that the constraint is optional; we can omit it and just create a new name for a type. The old name is not affected in any way.

The values of the subtype **FLOOR__TYPE** also belong to the type **INTEGER**, and no conversions are needed in assignments or other operations:

```
T : FLOOR__TYPE := 5;
...
T := T + 1; -- I is of type INTEGER
```

An important point to note is that constraints can only be checked during run time; in most cases they cannot be checked during compilation. Even if the compiler finds out that a constraint is going to be violated, it may only give you a warning, not an error message. If say, the variable **I** in the above assignment has a value of 90, a run time error will occur when this statement is executed. This means raising the predefined exception **CONSTRAINTERROR**, and ending the execution of the program if the programmer has not been prepared for such an error.

Here is an example of how to use a subtype:

```
type DAY__TYPE is (MON, TUE, WED, THU,
                  FRI, SAT, SUN);
subtype WORKING__DAY__TYPE is
  DAY__TYPE range MON .. FRI;
WORKING_HOURS : array (WORKING__DAY__TYPE)
  of FLOAT;
...
for D in WORKING__DAY__TYPE loop
  WORKING_HOURS(D) := 0.0;
end loop;
```

In this example, there is no need to check the index boundaries in the assignment during run time.

To complete the list of predefined types, we must now mention the predefined subtypes **NATURAL** and **POSITIVE** of the type **INTEGER**:

Note the similarities in the syntax:

```
type INT__TYPE is range 0..50;
type NEW__INT__TYPE is new INTEGER
                      range 0..20;
subtype SUB__INT__TYPE is INTEGER
                      range 0..100;
```

The first defines an integer type, the second a derived type, and the third a subtype.

```
subtype NATURAL is INTEGER range
  0 .. INTEGER'LAST;
subtype POSITIVE is INTEGER range
  1 .. INTEGER'LAST;
```

where **INTEGER'LAST** is the biggest integer in the implementation.

One final note associated with general data typing in Ada: even if two types are structurally equivalent they are not the same if their names are not the same:

```
type A__TYPE is array (1 .. 100) of BOOLEAN;
type B__TYPE is array (1 .. 100) of BOOLEAN;
```

These two types are different, and they can have different (user-defined) operations. According to a special rule, however, an array of the type **A__TYPE** can be converted to a value of **B__TYPE**, and vice versa.

Unconstrained Array Types

In Part 3 we described the basic forms of array types, record types, and access types. Now we can take a closer look at them.

The array types in the examples up to now have all been **constrained**, which means that the size of the array is fixed in the type definition. All variables of **A__TYPE** above are arrays with 100 components. But it is also possible to leave the size of the arrays open in the type definition and fix the size later, in a variable declaration. In the following example, **BIT__VECTOR__TYPE** is **unconstrained**:

```
type BIT__VECTOR__TYPE is
  array (INTEGER range <>) of BOOLEAN;
```

We now know that arrays of this type have indexes of the type **INTEGER** and components of the type **BOOLEAN**, but the number of components is not yet known. This is indicated by the **box <>** in the definition. When we declare a variable, the size must be fixed:

```
WORD : BIT__VECTOR__TYPE (0 .. 31);
-- 32 components
```

Here we have added an **index constraint** to the declaration. Similar constraints can also be added to subtype declarations:

```
subtype HALF__WORD__TYPE is
  BIT__VECTOR__TYPE (0..15);
```

In this case the variable declarations cannot contain a constraint. Note that a **box does not** mean that the size of an array can be changed during run time. If you need structures whose size can be changed dynamically, you must use access types.

The predefined type **STRING** is defined as an unconstrained array type:

```
type STRING is array (POSITIVE range <>) of
  CHARACTER;
```

A string is an array of characters. Here again, the length of a string must be fixed at the latest when declaring a string variable:

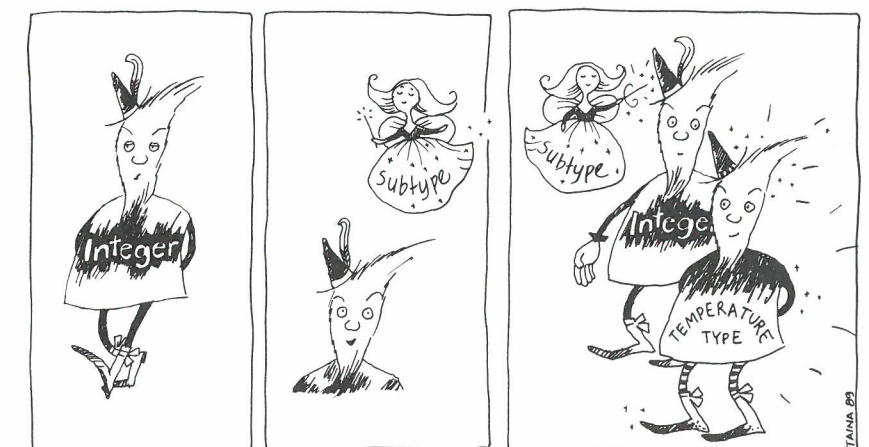
```
SHORT__STRING : STRING (1 .. 10);
LONG__STRING : STRING (1 .. 100);
```

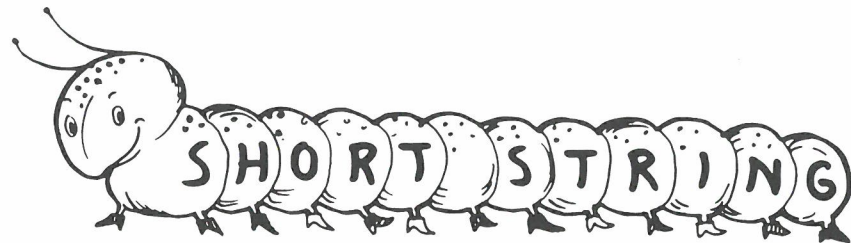
The lower boundary should preferably be kept at 1 when a string is defined.

Now here's an interesting point: if we try to assign **SHORT__STRING** to **LONG__STRING**, what do we get? The compiler tells the programmer that the exception **CONSTRAINTERROR** will be raised during run time, because the strings are of different length. We have now violated the index constraint associated with **LONG__STRING** in the same way as we violated the range constraint when assigning 90 to **T** of the subtype **FLOOR__TYPE**.

There must be a way to do the assignment. What we need is a new concept, a slice to state where we want to put **SHORT__STRING**:

```
LONG__STRING (1 .. 10) := SHORT__STRING;
-- Or (28 .. 37), or whatever
```





We now have strings of equal length on both sides of the assignment operator. A slice is a part of a one-dimensional array (not necessarily a string).

Note, that strings can be indexed in the normal way because they are arrays:

```
C: CHARACTER := LONG_STRING (22);
...
LONG_STRING (33) := 'X';
  --- Apostrophes (character)
LONG_STRING (3.6) := "TEXT"; --- Quotes (string)
```

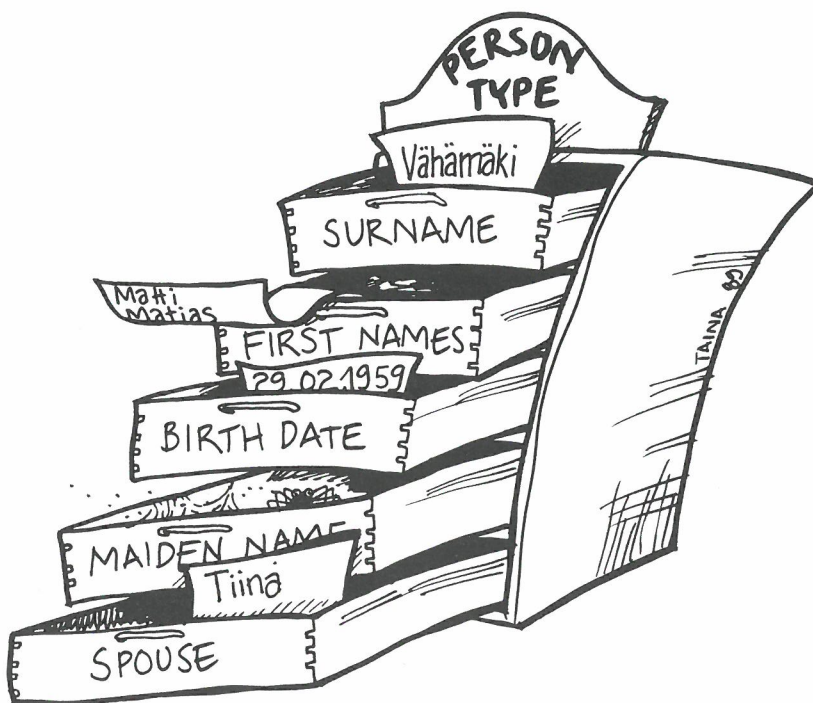
Variant Records

Ada, like Pascal, contains **variant records**. Such records can be used to implement structured real-world objects that are logically connected but differ slightly. Here we will take just one example and skip over the rules concerning variant records. A definition of a typical variant record is shown in Figure 1.

To separate the variants we must declare a **discriminant** (like S in Fig. 1) in the record type definition. All variables of PERSON_TYPE have the components SURNAME, FIRSTNAMES, and BIRTH_DATE, but only women have the component MAIDEN_NAME. The discriminant can have a default value as in Figure 1, but if we want to use some other value, or there is no default, it is given as a **discriminant** constraint in later declarations.



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



Discriminants can be used in other kinds of records, too:

```
type BUFFER_TYPE (SIZE : NATURAL := 100) is
  record
    POS : NATURAL := 0;
    VALUE : STRING (1 .. SIZE);
  end record;
```

Here the discriminant SIZE controls the length of a text buffer. If no discriminant constraint is given, the length is 100.

Access Types

Access types are used to implement objects whose size or form changes during program execution. Linked lists, trees, and graphs are familiar examples.

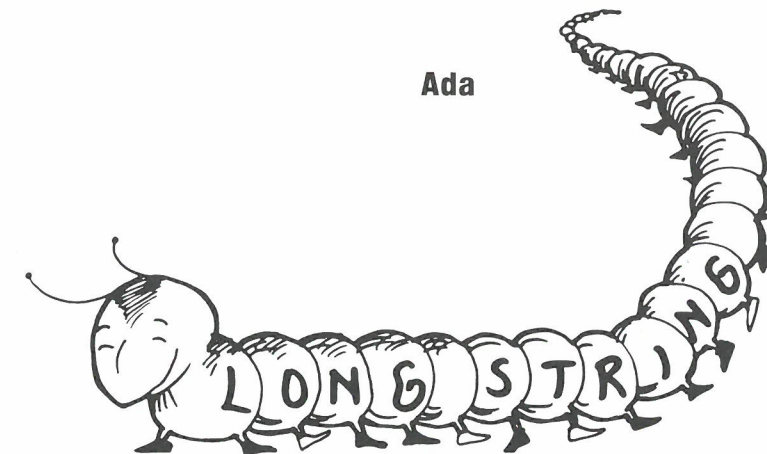
```
type SEX_TYPE is (FEMALE, MALE);
subtype DATE_TYPE is STRING (1..10);
subtype SURNAME_TYPE is STRING (1..20);
subtype FIRSTNAME_TYPE is STRING (1..20);

type PERSON_TYPE (S: SEX_TYPE := FEMALE) is
  record
    SURNAME : SURNAME_TYPE;
    FIRST_NAMES : FIRST_NAME_TYPE;
    BIRTH_DATE : DATE_TYPE;
    case S is
      when FEMALE =>
        MAIDEN_NAME : SURNAME_TYPE;
      when MALE =>
        SPOUSE : FIRST_NAME_TYPE;
    end case;
  end record;

EVE : PERSON_TYPE;
  --- EVE is a woman
ADAM : PERSON_TYPE (MALE);
  --- ADAM is a man
  --- (constraint)

subtype GUY is PERSON_TYPE (MALE);
```

Figure 1: A variant record type definition



Let us define an access type, whose values are pointers to persons, and declare a couple of access type variables:

```
type PERSON_LINK_TYPE is access PERSON_TYPE;
PERSON_1,
PERSON_2 : PERSON_LINK_TYPE;
```

Access type variables are the only variables that are automatically initialized in Ada; they get the value **null**, a pointer to nowhere. During run time person records are dynamically created with the **new allocator**:

```
PERSON_1 := new PERSON_TYPE;
PERSON_2 :=
  new PERSON_TYPE' (MALE,
    "Wilson",
    "John Edward",
    "1952-08-07",
    "Ann");
```

Note, that this **new** has no connection with the new we use in the derived type definitions. The variable PERSON_2 is initialized with a record aggregate. Note, too, that if there is a the discriminant, it is the first component of the record. No explicit dereferencing is needed for selecting a component of a person record:

```
PERSON_1.BIRTH_DATE := "1953-06-25";
```

Testing the equality of access variables (PERSON_1 = PERSON_2) means testing whether the pointers **denote the same dynamic variable**. If we want to test the equality of the **contents** of the variables denoted by two pointers, we must use the selector **all**:

```
if PERSON_1.all = PERSON_2.all then ...
```

Attributes

For a program to be maintainable and portable a knowledge of **attributes** is needed. Some 40 attributes are defined in the Ada standard. Most of them are associated with data types, so this is a suitable point to mention a few of them.

We have already seen one attribute, LAST, denoting the biggest possible integer, in the definitions of the subtypes NATURAL and POSITIVE above. LAST is an attribute associated with all scalar types or array types. We can loop through the LONGSTRING, without needing to know its length, by writing:

```
for I in 1 .. LONG_STRING'LAST loop
```

or perhaps better:

```
for I in LONG_STRING'FIRST ..
LONG_STRING'LAST loop
```

or shorter:

```
for I in LONG_STRING'RANGE loop
```

using the attributes FIRST and RANGE that are also available for array types. If the length of the LONG_STRING is changed, the loop statement need not be changed (assuming we still want to go through the whole array).

Another useful pair of attributes is PRED and SUCC, applied to the predecessor and successor of a discrete type value. For example, DAY_TYPE' SUCC (WED) / THU.

The attribute POS gives the position number of a discrete type value:

```
DAY_TYPE'POS (TUE) = 1
```

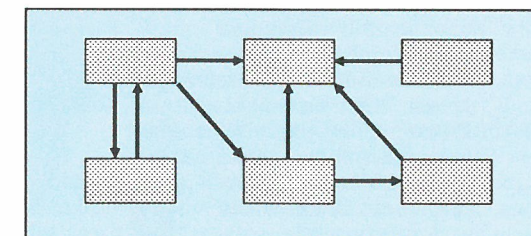
and VAL denotes the inverse:

```
DAY_TYPE'VAL (0) = MON
```

There are also lower-level attributes ADDRESS, which gives the machine address of any object, and SIZE, which yields the number of bits allocated to hold an object.

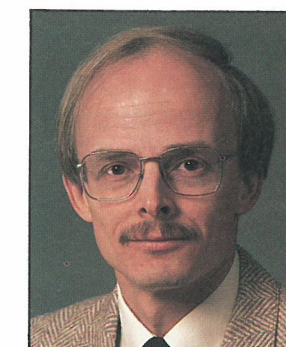
Exercise

Construct the data type definitions needed to implement a graph of nodes in which the nodes are connected to other nodes with varying numbers of edges. Between two arbitrary nodes x and y there may be 0..2 edges.



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, August 1989.
- 4/ Amoroso, S., Ingargiola G.: Ada. An Introduction to Program Design and Coding, Pitman, 1985.



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

Tien päällä taas

Saarisen Tapsan muistelmat — osa V

Tapsa keräili kaikessa rauhassa itseään myöhäisen aamukahvin ääressä. Leena oli jo aikaa vienyt pojan hoitoon ja kadonnut määrätunneiksi kirjanpidon kassaraksi.

Tapsa oli kotiutunut Rauta-Yhtiön systeemiestistä vasta aamutunneilla ja päätti tavoitella työpaikalleen aikaisin-
taan iltapäiväksi hyvin muistaen Tiukka-Tasasen määräyksen kaikkien ehdottomasta läsnäolovelvollisuudesta aamun laskutuspalaverissa. Ovikello soi kesken aamun hiljaisuuden ja myyntitykki Mäkinen kohelsi vauhdilla sisään matkalaukku kainalos-
saan.

— Moro Tapsa! Nyt tuli lähtö, kaappaa kassallinen kalsareita mukaan, taksi odottaa alhaalla.

— Mihin nyt ollaan menossa? Missä kaupantekokravattisi on?

— Ei nyt ehdi selitellä, pistetään menoksi vaan!

— Hyvä on, kirjoitan vain Leenalle lapun pöydälle, mitä pistän siihen tuloajaksi?

— Lättää siihen, että palaamme, kun siltä tuntuu. Nyt ovat suuremmat voimat pelissä.

— Selvä!

Vihdoin lentokoneessa saatuaan pullon shamppanjaa, pari olutta ja konjakit alkoi Mäkinen kertoa, mistä on kysymys.

— Sain aamulla kuulla, että hävisin Masuunin suuren konekaupan. Kolme vuotta sitä hie-
rottiin, syötiin herroiksi klubilla, hiottiin alennuksia ja yksityiskohtia. Käytiin ulkomailla referenssikäynneillä, pidettiin esitelmiä ja juhlapuheita, minäkin jouduin kolme uutta pukua ostamaan projektin aikana, joulunpyhinäkin väänsin kuusen alla sopimustekstiä. Vielä viime viikolla homman piti Masuunin atk-väen mukaan olla kypsää tavaraa. Ja tänään ovat tehneet sopimuksen sen Aaltosen nakki-
kioskin kanssa.

— Sen saman Aaltosen, joka oli meillä vuosia sitten harjoittelijana, eikä saanut ikinä ainoaa ohjelmaansa toimimaan?

— Sama mies. Kai tämän muuten vielä kestäisi, mutta kun menivät siltä sliipatulta kaljupää-
kekkeruusilta ostamaan. Äijä vaan liittää kansainvälisillä lentokentillä valkoisessa kesäpuvussa ja junailee minun selkäni takana valmiin kaupan itselleen te-

kemättä oikeastaan ollenkaan hommia sen eteen. Allekirjoitus-
tilaisuuskin on kuulemma jos-
sain erähotellissa tennisottelun ja paratiisihoidon välisenä aikana. Kyllä mua riepoo!

— Eikö sen kaiken järjen mukaan pitänyt olla meidän keikka, meillä oli halvin ratkaisu ja sovel-
lus, joka kävi Masuuniin räätä-
löimättä. Mistä se oikein kii-
kasti?

— Olen melko varma, että kaikki jäi jälleen kiinni siitä, kun Tiukka-Tasanen nimitti Masuunin johtajaa vuosia sitten saunaillassa pikkusieluksi. Ei siinä auta atk-asiantuntijoiden perustelut, ihmiset on ihmisiä. Kyllä Haapalainen sen muistaa ikänsä.

Tapsa kuunteli ja katseli samalla ympärillä tarjoutuvia erivärisiä, -kokoisia ja -hintaisia paljaita pyllyjä.

— Sehän on Tiukan moka, eikä sinun.

— Älä usko. Aamulla oli jo hätäpalaveri, missä Tasanen kaatoi jupakan niskaani, kun olen kuulemma liian pehmo ja aina ottamassa yksityishenkilöitä huomioon. Business ei ole kuulemma mitään sosiaalista toimintaa vaan tiukkaa sotaa. Jos minulla olisi ollut samaa apinan rai-
voa kuin hänellä, kauppa olisi ollut meidän.

— Paskapuhetta. Mutta nyt mennään eikä meinata, mihin muuten?

— Kohta näet. Luota Lufthansa-

Düsseldorfissa heitä odotti vuokra-auto. Tapsa suhteellisen selvänä ryhtyi ajamaan. Mäkinen ilmoitti halunneensa aina nähdä Ajaxin pelaavan. Niillä oli illalla Amsterdamissa matsi, siis nokka sinne. Kentän löytäminen Amsterdamista ei ollut helppoa, Tapsa yritti seurata iloisesti pulloa ja karttaa heiluttavan Mäkinen oh-

jeita, mutta stadionia ei näkynyt. Pitkän alikulkutunnelin päässä heidän eteensä ajoi poliisiauto, Tapsa heitti viimeiset kurkkupas-
tillit suuhunsa ja valmistautui pitkään oleskeluun Hollannissa, putkasta käsin tosin. Korsto poliisi kävellä lampsi muikea virne huulillaan ajurin luo ja ilmoitti jo pidempään ajelleensa heidän perässään. Oli varmasti kivaa kiertää samaa ympyrää keskellä suurkaupunkia. Korsto tarjoutui viemään heidät poliisisaattuessa toivottuun paikkaan. Huumorintajuinen flaami pisti vielä pillit ulisemaan, jonon hänille liittyi pari maijaa ja niin Suomen urhoja vietiin turvallisesti Ajaxin stadionille.

Pojat koukkasivat tietysti kotijoukkueen kannattajien pubin kautta. Siellä selvisi, ettei peli ollutkaan täällä, vaan suuremman katsojajoukon toivossa vanhalla Olympiastadionilla. Fanit lupasivat viedä eksyneet turistit mukanaan oikeaan paikkaan. Tapsa ja Mäkinen suostuivat mielellään ja ryhtyivät halulla särpimään pohjoisen pojille tarjottuja olusia. Palkaksi he pitivät Ajaxin väelle pikakurssin siitä, miten Petri Tiainen lausutaan oikein flaamin kielellä. Siitä meinas jo vähän riitaakin tulla, onneksi lähtöhetki oli käsillä.

Jo ennen ottelun alkua Feyenoordin kannattajat pistivät hulinaksi, ja lihamassan painolla sääntäisivät kotijoukkueen kannattajien kimppuun. Viisisataa poliisia ja satakunta koiraa hiljensivät kahinan, tuloksena 80 putkaan, 40 sairaalaan ja Mäkinen sormet mustaksi, kun hän kiihkeimmillä hetkillä yritti kiivetä pakoon rauta-aidan yli ja mellakapoliisin toimesta hänet kannatettiin tylsty alas. Tapsa kaipasi salaa nuoruutensa pelejä rauhallisella pallokentällä, missä pahinta huliganismia oli fanaattisten äitien parjaukset tuomarille liiallisen likapyykin sallimisesta. Kai tämä on sitä suurta maailmaa, pohti Tapsa puhdistessaan nahkatakistaan siihen pystyyn lätkäistä jäätelötötteröä.

Pelin jälkeen eivät edes punaisen lamppujen kadut saaneet Mä-
kistä irtomaan masennukses-



taan. Mäkinen muisteli vuosiaan busineksen teossa, vuodesta toiseen kasvavia tavoitteita, yhä pienempiä katteita, älyttömällä vauhdilla kehittyvää tekniikkaa, alati julmenevaa Tiukka-Tasasta ja vapaa-aikaa, jota sentään alan kultaisina vuosina vielä oli. Nyt hän kirjoitti jo unissaankin kuolettavan yksitoikkaisia sopimus-
tekestejä. Ennen oli edustukseen reilusti varoja, nyt pitää jo nekin laskea etukäteen, perkeleen perkele. Tapsa kuunteli ja katseli samalla ympärillä tarjoutuvia erivärisiä, -kokoisia ja -hintaisia paljaita pyllyjä. Nuo ne sentään pysyvät vuodesta toiseen samalaisina, huomasi Mäkinenkin.

Kaverukset päättivät pitää suunnittelupalaverin läheisessä pubissa. Vasta perälle päästyään he huomasiivat, että paikka oli omistettu miesten vaatteisiin pukeutuneille neitokaisille. Murhaavien silmäysten jälkeen parivaljakko heitettiin muutaman amatsonin toimesta kadulle, missä he huomasiivat kuppilan paljastavan nimen Madame George välikkyyvän valomainoksessa. Mäkinen toipui ensin.

— Nyt tiedän, haluan Prahaan. Tahdon nähdä Kafkan kotitalon. Se ehkä auttaa tähän masennukseen.

— Jumalauta, oletko hullu, sinnehän on yli tuhat kilometriä.

— Se on sinun ongelmasi, minä en voi ajaa, minulla on sormet mustina.

Prahan kevät loisti aurinkoisena. Vaatimattomassa hotellissa nukutun yön jälkeen kaappasivat länsimaiset tietokonemiehet konjakkilitran muovikassiin ja suunnistivat keskustaan. Mäkinen tallasi onnellisena edestakaisin vihdoin löytämänsä Kafkan asuintalon edustalla. K:n korkokuva talon seinällä oli kiinni ruuveilla, se oli helppo irroittaa yleisen mielipiteen taas joskus vaihtuessa. Ajattele nyt, tässä se muuttui suunnatto-

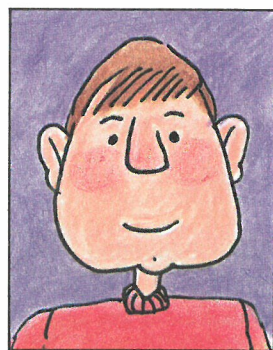
maksi syöpäläiseksi, hoki Mäkinen useaan kertaan, kunnes Tapsa lähti vetämään häntä totaalisen muodonmuutoksen pelossa kaupungin kattojen ylle, luostarinmäelle.

Kipuamisen jälkeen oli pakko poiketa pilsulle. Kun heidät tunnistettiin suomalaisiksi, lyöttäytyi seuraan Josef, joka pubin kantajoukoissa tunnettiin lempinimellä Marjamäki, minkä hän oli ansainnut Prahan MM-keväänä kehuttuaan maalivahti Dzurilalle juovuspäissään tämän Mansen hasardipakin mahtavaa kääntölaukausta. Vlado oli lyönyt Josefin hampaat sisään ja siellä ne olivat vieläkin. Marjamäki kertoi partisaanivuosiensa, kuuden vuoden kakusta posessa, kanto kassissaan Hitlerin valokuvaa ja läjää suomalaisia kuulakärkikyniä. Marja pisti viimeisetkin lantinsa peliin tarjotessaan aitoa tummaa pilsneriä pohjoisen sankarikansan edustajille, jotka sammuivat baarituoleilleen sylikäin. Josef jatkoi tarinointiaan, kunnes huomasi kuulijajoukon hiljentyneen ja kuljetti parivaljakon taksilla hotelliin nukkumaan. Onneksi Mäkinen taskun pohjat olivat täynnä monenlaista valuuttaa, niistä riitti vähän vielä Josefin Karhu-urheilulaukun sivutaskuun.

Viikkoa myöhemmin pojat seisoivat kotimaan lentokentän vastaanottohallissa, Mäkinen ryhdykkään uudessa saksalaispuvussaan, jolla hän lupasi hankkia vuosikymmenen kaupan kotiin ensi tilassa. Mäkinen nykäisi Tapsaa olkapäästä.

— Yhtä asiaa olen tällä matkalla monasti ihmetellyt. Miksi kaikki hyväsydämiset ihmiset ovat enemmän tai vähemmän renttuja?

Samalla Tapsakin huomasi odotuslasiin hengittävän Tiukka-Tasasen. Tämä oli jo monta päivää kytännyt kaikki Keski-Euroopan koneet. Aaltonen oli nähnyt pojat siellä ohimennen messumatkallaan, juovuksissa olivat kuulemma olleet. Tasanen raahasi Tapsan ja Mäkinen suoraan taksiin ja siltä paikalta Rauta-Yhtiön ennalta tiukkaan kauppaneuvotteluun. Miljoonia oli taas tulossa, jos vaan oltaisiin kovina, ei tuumaakaan periksi. Takapenkillä Mäkinen huomasi kevään tulleen jo Suomeenkin.



Nimimerkki Kevin Puskas on liki neljäkymppinen pitkän linjan atk-ammattilainen, jonka harastuksiin kuuluvat niin jalkapallo, kirjallisuus kuin olutkulttuurikin.

SQL-System yhdistää sovelluskehityksen, tiedonhallinnan ja

Verkkotietokanta on kolmen kauppa

Toimivien kokonaisvaltaisten sovellusten luominen on osoittautunut erityisesti verkkoympäristössä ongelmalliseksi. Guptan SQL-System on yksi ratkaisu, tarjoaahan se käyttäjälleen paitsi tietokantaohjelmiston myös sovelluskehittimen ja tietoliikennepalvelut.

DC (International Data Corporation) julkaisi viime syksynä tutkimuksen, joka osoitti suomalaisten yhtiöiden yhä kasvavan kiinnostuksen siirtää sovelluksiaan nykyisistä mini- ja suurympäristöistä mikrotietokoneverkkoihin.

Kehitystä on haitannut ammattimaisten, raskaiden tietokantasovellusten tekoon tarkoitettujen työvälineiden puute. Ensimmäisenä markkinoille ehti Kalifornialainen Gupta Technologies International, joka jo toissa keväänä julkaisi käyttäjä-palvelin-arkkitehtuuriin perustuvan SQL-System-sovelluskehitysympäristön.

Paineet palvelimelle

Käyttäjä-palvelin-arkkitehtuurissa sovellukset jaetaan kahtia: Työasemassa pyörivä käyttäjä-sovellus hoitaa käyttöliittymän hallitsemisen ja sovelluslogiikan sekä luo palvelupyyntöjä, jotka se lähettää edelleen palvelimessa sijaitsevalle palvelin-sovellukselle.

Tietokantapalvelimen tapauksessa palvelin työaseman kyselyn saatuaan kääntää sen suorittamaan muotoon, optimoi suoritusjärjestyksen, suorittaa varsinaisen haun ja palauttaa tulokset työasemalle. Näin sama tieto voidaan helposti jakaa eri työasemissa sijaitsevien sovellusten vä-

verkkopalvelut

lillä. Myös käyttöoikeuksien tarkistukset, virheistätoipumisrutiinit ja muut yleiset toimenpiteet voidaan suorittaa keskitetysti.

Käyttäjä-palvelin-arkkitehtuurin avulla koneiden tietojenkäsittelyvoima voidaan jakaa sinne missä sitä eniten tarvitaan. Palvelin voi olla esimerkiksi tehokas 386-pohjainen mikro, kun työasemiksi saattavat riittää 8086-suorittimella varustetut koneet.

Kehitystä kehittimellä

SQL-System koostuu kolmesta osasta: SQLWindows-sovelluskehittäjästä, SQLBase-tietokantapalvelimesta sekä SQLNet-yhdyskäytävästä, joka yhdistää järjestelmän IBM-suurkoneissa pyörivään DB2-tietokantaan.

SQLWindows on sovelluskehittäjä, jonka avulla voidaan rakentaa sovelluksia, jotka yhdistävät Microsoft-Windowsin graafisen käyttöliittymän sekä SQL-tietokannan tietojenkäsittelykapasiteetin.

Ohjelmointityötä on pyritty helpottamaan mahdollisimman paljon. Käyttöliittymä piirretään asettamalla hiirellä erilaisia näytöolioita (syöttökenttiä, taustatekstejä, painonappeja, listauksia, ...) paikoilleen, jonka jälkeen varsinainen ohjelmakoodi kirjoitetaan helpolla neljännen sukupolven Sal-ohjelmointikielillä.

SQLWindows toimii toistaiseksi vain MS-Windowsin alaisuudessa, mutta se tultaneen vielä tämän vuoden aikana siirtämään OS/2 Presentation Manager -ympäristöön.

Tietokanta verkossa

SQLBase on työryhmäverkossa toimiva tietokantapalvelin, jonka kyselykieli noudattaa täysin SQL-standardia. SQL (Structured Query Language) on jälleen yksi IBM-ympäristöstä polveutuva standardi, jonka sekä ANSI- että ISO-organisatio on sittemmin omaksunut. SQLBase tarjoaa myös muita, ennen vain suurkoneympäristöissä toteutet-

tuja palveluja kuten tapahtumatuon (commit/rollback), viastoitumispalvelut (journaling) sekä keskitetyn käyttöoikeuksien hallinnan.

Työasemasovellukset voidaan rakentaa joko SQLWindowsin avulla tai perinteisiä ohjelmointikieliä käyttäen (MSC, Microfocus, Cobol). Kommunikointi työasemasovelluksen ja tietokantapalvelimen välillä tapahtuu Netbios-tasolla, joten SQLBase sopii moniin eri valmistajien verkkoympäristöihin.

Verkkoja on mahdollista myös yhdistellä erilaisten Netbios-siltojen avulla siten, että kaikkien verkkojen työasemat voivat käyttää yhdessä verkossa olevaa tietokantaa. Järjestelmässä voi toimia yhtä aikaa myös useampia tietokantapalvelimia, joista jokainen voi edelleen sisältää useita tietokantoja.

DB2 kuvassa mukana

SQLNet-yhdyskäytäväohjelmisto koostuu kahdesta osasta, DB-Gatewaystä ja SQLHostista. DB-Gateway on verkossa toimiva ohjelma, joka välittää sovellusten SQL-komennot APPC-yhteyden kautta IBM-suurkoneeseen. Siellä ne vastaanottaa SQLHost, joka välittää komennot edelleen DB2-tietokantaan. Kyselyjen tulokset reititetään samaa kautta takaisin sovellusohjelmalle.

Yhdyskäytäväohjelmisto näyttää verkossa aivan samalta kuin paikallinen SQLBase-tietokanta. Sovellusten ei tarvitse tietää, sijaitseeko käytettävä tietokanta fyysisesti paikallisessa verkossa vai onko kyseessä DB2-tietokanta. Ainoa ero toiminnassa on reitityksen aiheuttama lisäviive vastauksissa.

Kehitystä Suomessa

Nokia, sen lisäksi että edustaa yksinoikeudella näitä Guptan tuotteita Pohjoimaissa, pyrkii myös kehittämään niitä edelleen. Tätä työtä edustavat esimerkiksi sovelluskehitystyön avuksi tarjotut opaskirjat. 'Nokia Data

Guidelines for User Interface Creation' paneutuu graafisten käyttöliittymien suunnittelussa vastaantuleviin ongelmiin ja pyrkii luomaan yhtenäisyyttä Nokian graafisiin sovelluksiin. 'SQLWindows Developers' Guide' antaa puolestaan ohjeita ja neuvoja sekä edellisen oppaan ajatuksien soveltamiseksi että SQLWindows ohjelmien suorituskäytön parantamiseksi. Molemmista edellä mainituista oppaista on ilmestynyt ensimmäiset, sisäiseen käyttöön tarkoitetut versiot.

Nokiassa on toteutettu myös liittymä Excel-taulukkolaskennan ja SQLBasen välille. Liittymän avulla Excelin käyttäjät voivat hakea taulukoihinsa tietoa SQL-tietokannasta tai vastaavasti päivittää tietokannan tietoja. Lisää tietoa tästä ohjelmatuotteesta löytyy 'Excel uppoaa järjestelmiin' -artikkelista (s. 30-31).

Kilpailu kiristyy

SQL-System oli siis ensimmäinen lähiverkkokäyttöön tarkoitettu raskaan sarjan tietokantajärjestelmä. Nyt on syntymässä jo muitakin vastaavia järjestelmiä, joista ehkä kaikkein mielenkiintoisin on Microsoftin, Ashton-Taten ja Sybasen yhteistyötä oleva SQL Server.

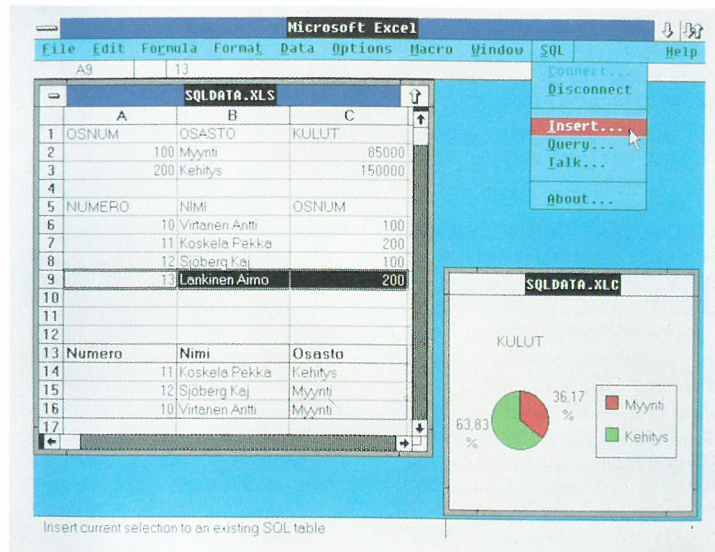
SQL Server perustuu aivan samanlaiseen käyttäjä-palvelin-arkkitehtuuriin kuin SQLBase. Toistaiseksi ei ole kuitenkaan tarkkaa tietoa esimerkiksi siitä, minkälaisia mahdollisuuksia SQL Server tarjoaa liitettävyydessä muihin tietokantoihin, tai milloin järjestelmään saadaan mukaan graafinen Windows-sovelluskehittäjä.

Myös muilla valmistajilla on kehitteillä tai jo valmiinaakin vastaavia tietokantajärjestelmiä. Tällaisia ovat esimerkiksi Novell NetWare SQL, Oracle Server sekä XDB-server. Unohtaa ei sovi myöskään IBM:ää, jonka tietokantajärjestelmästä on kuitenkin toistaiseksi olemassa vain yksittäisessä mikrossa pyörivä versio (OS/2 1.1 EE/ Database manager).

Mikko Röntynen toimii Nokia Datan Tuotekehityksen sovelluskehitysympäristöt -ryhmässä. Hän työskentelee tuoteasiantuntijana työasemajärjestelmien tietokanta- ja sovelluskehitysvälineiden parissa.

Excel ja SQLBase ne yhteen soppii, huomenna...

Excel uppoaa järjestelmiin



Excelin monipuoliset liittymä-mahdollisuudet tekevät siitä työvälineajattelua tukevan ohjelmiston. Excelin voi sulauttaa osaksi järjestelmää, jossa se avustaa analyysien tai esitysgraafikan laadinnassa. Pelkkä leikkaa-liimaa ei enää riitä, kun ulkopuolinen sovellus haluaa tehdä lennosta analyysin taulukolaskimella ja käyttää tulosta välittömästi hyväkseen, tai kun pitää esittää graafisesti työryhmän SQL-palvelimessa olevat varastotiedot. Näistä tehtävistä suoriutuakseen Excel hyödyntää Windows-ympäristön DDE-yhteyskäytäntöä ja dynaamista linkitystä.

Makrokieli automatisoi toimintoja

Excelin makrokieli on Basicin kaltainen kieli, joka soveltuu parhaiten Excelin toimintojen automatisointiin. Makroilla on helppo käyttää kaikkia Excelin piirteitä ja valikkorakenteen sekä valintalaatikon voi muokata mieleksikseen. Lisäksi makrokieleen on liitetty monia Windows-ympäristöä hyödyntäviä toimintoja.

Excelillä tehtävien makrosovellusten on syytä pysyä termin "Excelin toimintojen automatisointi" alueella. Raskaampi sovelluskehitys kannattaa tehdä varsinaisilla kehittimillä kuten SQLWindowsilla. Kun Excel siirretään Presentation Manageriin, toimivat makrot sellaisinaan uudessa ympäristössä.

Excel liittyy ympäristöön

Excel voi keskustella muiden sovellusten kanssa monin eri tavoin: tiedoston, leiketaulun, DDE-yhteyskäytännön, näppäinsarjojen simuloimalla tai dynaamisen kirjaston avulla. Tiedostosiirto, leiketäulu ja DDE eivät välttämättä vaadi käyttöä makroista käsin, mutta niiden soveltamismahdollisuudet laajenevat makrokielen myötä.

Tiedostosiirto on usein ainoa vaihtoehto siirtää tietoa Excelin ja muiden kuin kuin Windows-sovellusten välillä. Excelin kanalta on miellyttävintä, jos tiedosto on sen omassa BIFF-esitysmuodossa, mutta Excel osaa lukea suoraan myös muiden suosituimpien ohjelmistojen datatiedostojen. Makroilla voidaan ohjelmoida lisää eksoottisempia muunnoksia.

Leikkaa-liimaa-toimintoa leiketäulun kautta hyödyntävät lähes kaikki Windows-sovellukset. Toiminnon käyttö on eri sovelluksissa yhtenäistä ja helppoa. Se ei kuitenkaan sovellu suurten tietomäärien siirtoon.

DDE-yhteyskäytännöstä ja Excelistä puhuttiin enemmän Softarin numerossa 4/88. DDE soveltuu parhaiten ajantasaiseen tiedonsiirtoon. Kommunikoivien sovellusten on oltava Windows-sovelluksia ja niiden on mahduttava pyörimään samanaikaisesti.

Excel-makro pystyy käynnistämään toisen Windows-sovelluk-

sen ja simuloimaan näppäinten painalluksia siinä.

Excel kykenee käyttämään Windowsin dynaamisia kirjastoja dynaamisen linkityksen avulla. Linkitys tapahtuu ajonaikaisesti, joten itse Exceliä ei tarvitse (on-neksi) linkata uudestaan. Tämä tapa laajentaa makrojen käyttöä- luetta on saanut turhan vähän huomiota osakseen. Sen avulla voi suorittaa käyttöjärjestelmä-läheisiä tai tehokkuutta vaativia tehtäviä. Sovelluskehittäjä voi myös itse laatia dynaamisia kirjastoja Windows Software Development Kitin avulla. Tavallisen kirjaston muuntaminen dynaamiseksi on kertaluokkaa helpompaa kuin tavallisen sovelluksen muuntaminen Windows-sovellukseksi. Excel käyttää kirjastorutiineja kahden makrofunktion, REGISTERin ja CALLin avulla.

Excelin ja SQLBasen välillä on periaatteessa ollut yhteys heti tuotteiden ilmestyttyä.

Sovellukset työryhmäverkoissa

Excel voidaan laajentaa todelliseksi työryhmäsovellukseksi, jonka arkistopalvelut on (lähes) täydellisesti piilotettu käyttäjältä. Liittymä arkistosovellukseen kannattaa toteuttaa dynaamisella kirjastolla tai DDE-yhteyskäytännöllä. Tämä täytyy luonnollisesti huomioida arkistosovelluksen rakenteessa.

Excelistä voidaan toteuttaa suora yhteys työryhmän SQL-tietokantapalvelimeen. Excelin ja SQL-palvelimen väliin tarvitaan ohjelmamoduuli, joka liittymätavasta riippuen on joko dynaaminen kirjasto tai DDE-yhteyskäytäntöä tukeva Windows-sovellus. Jos tietokantapalvelin tukee automaattista päivityspyyntöä, on yhteys siihen järkevintä toteuttaa DDE-yhteyskäytännön avulla.

Excelin ja SQLBasen yhteys

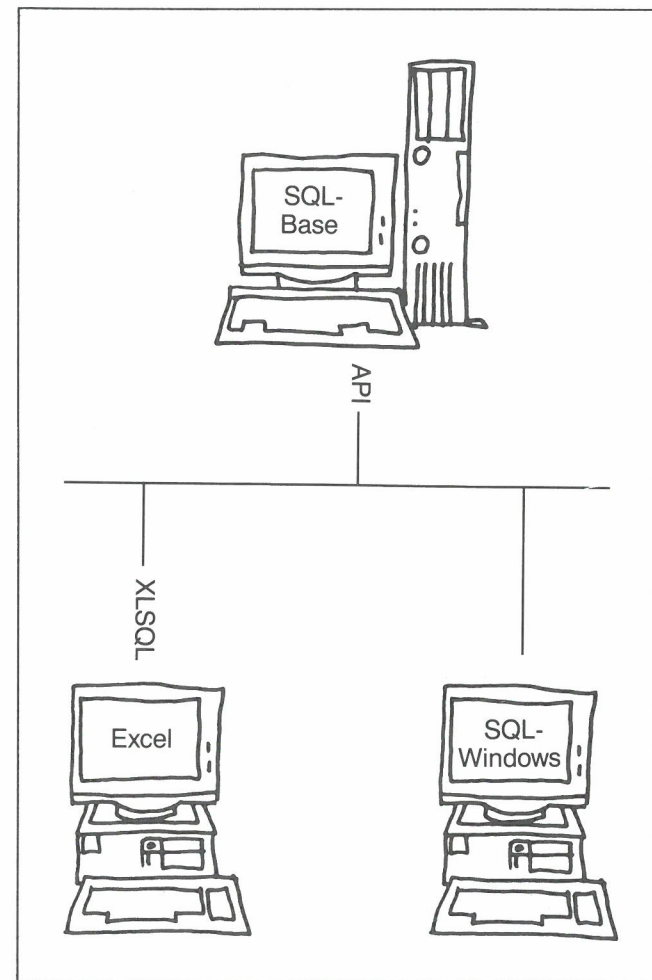
Sovellukset käyttävät SQLBase-tietokantapalvelinta standardin sovellusrajapinnan (Application Program Interface, API) kautta. Windows-sovelluksia varten API löytyy myös dynaamisena kirjastona, jota SQLWindows ja nyt myös Excel käyttää. Koska API dynaamisena kirjastona on juuri se kaipaamamme välittävä ohjelmamoduuli, on Excelin ja SQLBasen välillä periaatteessa ollut yhteys heti tuotteiden ilmestyttyä.

Ennen kuin yhteys toimii, on ratkaistava joukko ongelmia. Excelin kirjastorutiinien kutsumekanismeissa on rajoituksia, minkä vuoksi se ei pysty vastaanottamaan API-funktioilta muuta palautetta kuin virhekoodin. Lisäksi SQLBasen käyttö vaatii sovelluksen puolella käyttöjärjestelmä-läheisiä toimenpiteitä, kuten dynaamista muistinvarausta, mikä ei onnistu Excel-makrojen tasolla.

Edellä mainitut ongelmat ratkaisee Nokia Datan XLSQL-liittymäkirjasto, joka toimii Excelin ja SQLBasen API:n välissä. XLSQL välittää vain ongelmalliset kutsut, koska suurinta osaa API-funktioista Excel voi kutsua suoraan.

XLSQL versio 1 mahdollistaa Excel-makroissa seuraavat kolme perustoimenpidettä: 1) valitun alueen kopiointi laskenta-alustalta SQL-tietokannan taulukoon, 2) tietojen haku SQL-kyse-lyllä laskenta-alustalle ja 3) SQL-lauseen suoritus. Kolmannen kohdan SQL-lause ei voi olla SELECT eikä se saa sisältää sidottuja muuttujia. SQL-lauseet voivat olla enintään 255 merkkiä pitkiä, mutta rajoituksen voi usein kiertää näkemysten (view) avulla.

On mahdollista rakentaa työryhmäsovellus, jossa ohjelmiston runko on kehitetty SQLWindowsilla ja tiedon analyysi ja grafiikka tehdään SQL-tietokannasta suoraan Excelillä. Ilman Exceliä jouduttaisiin analyysi- ja grafiikkaosin ohjelmoimaan sovelluskehittäjällä, eikä kuitenkaan päästäisi Excelin tasolle ominaisuuksissa ja käytön helppoudessa. ■



Kuva 1: Nokia Datan XLSQL-moduuli poistaa viimeiset esteet Excelin ja SQLBasen suoralta yhteydeltä.



Tero Koskivirta työskentelee Nokia Datan tuotekehityksessä ohjelmistot-ryhmässä ohjelmistoasiantuntijana.

SQLWindows-kurssi

Softarin edellisessä numerossa aloitimme SQLWindows-kurssin, jolle seuraava tässä jatkoa. Ensimmäisessä osassa tutustuttiin SQLWindowsiin ohjelmointiympäristönä, Windows-ohjelmoinnin olio-käsitteeseen sekä SAL-kielisen ohjelman vakiorakenteeseen.

Kurssin tässä osassa kertaamme ensin edellisen osan pääkohdat ja aloitamme siten pienen esimerkkisovelluksen rakentamisen. Vastaavan esimerkkisovelluksen rakentamiseen artikkelisarjan kirjoittajalta kului Windows SDK:n ja C-kielen avulla aikaa noin kolme kuukautta, kun lähtötiedot Windows-ohjelmoinnista olivat olemattomat, mutta C-kielstä vanhat. SQLWindowsin avulla saman sovelluksen luomiseen meni noin yksi viikko, kun lähtötiedot SQLWindowsista olivat olemattomat, mutta Windows-käsitteet jo selvillä.

Esimerkkisovelluksena toimii yksinkertaisen jäsenrekisterin hoito. Tällä kertaa teemme sovellukseen sopivan valikon ja siitä käynnistettävään keskusteluikkunan, jonka avulla voidaan syöttää uuden jäsenen tiedot. Seuraavassa numerossa kehitämme sovellusta edelleen siten, että tiedot viedään tietokantaan ja ne voidaan tulostaa jäsenluetteloksi list boxiin. Sarjan viimeisessä osassa tutustumme table window -rakenteeseen, jonka avulla tietokantaa voidaan kätevästi lukea, muuttaa jne.

Viime numerossa ...

Sarjan ensimmäisessä osassa tarkastelimme SAL-kielisen ohjelman perusrakennetta, joka oli kuvan 1 mukainen. Kuvassa on NEWAPP.APP-sovellusrun- gon kohdat Global Declarations, Form Window (pääikkuna) ja Dialog Box avattu. Koska nämä kolme kohtaa alkavat samalta sarakkeelta (vrt. Options Box: Same level), ne ovat keskenään samantasoisia.

Tutkitaan ensin ajamalla NEWAPP.APP ja lukemalla SAL-ohjelmaa, miten Dialog Box käynnistetään ja miten se suljetaan. Rivi, jolla on seuraavan tason rivejä, voidaan avata näpyttämällä hiiren vasenta näppäintä nopeasti kaksi kertaa, kun kohdistin on rivin alun täytetyn mantelin kohdalla. Vastaavasti sulkeminen käy kaksoisnäpytyksellä otsikkorivin kohdalla. Nämä ovat vaihtoehtoisia tapoja Outline-ikkunan Outline-valikosta löytyville tavoille.

Viimeksi piirsimme kokeeksi painonappi-olion Pääikkunaan. Muistatko vielä, mihin kohtaan kuvassa 1 esitetty ohjelmarun-

koa SQLWindows teki siitä SAL-ohjelmanpätkän?

Palautetaan vielä mieleen, mitä nämä nykyisin muodossa olevat oliot oikein ovat. Kullekin oliolle on siis ominaista sen rakenne ja toiminta. Esimerkiksi painonappiolion sijainti näytöllä ja sen sisältämä teksti määrittelevät sen rakenteen. Toisaalta painonapin SAL-ohjelmakoodin Message Actions -kohta määrittelee sen toiminnan. Toiminnan laukaisee jokin sanoma, esimerkiksi tieto hiiren napin painalluksesta kohdistimen ollessa olion kohdalla.

SQLWindowsin ohjelmatiedostot

Viime kerralla käynnistimme SQLWindowsin kaksoisklikkaamalla tiedoston NEWAPP.APP nimen kohdalla. Tällöin käynnistimme ohjelmankehitysympäristön, ja näyttöön ilmestyi sekä Outline eli ohjelmaikkuna että Form Window eli pääikkuna. Tällöin saatoimme työskennellä Design-tilassa eli muokata ohjelmaa. Lopuksi testasimme ohjelman toimivuutta käynnistämällä sen User-tilaan ohjelmankehitysympäristön roikkuessa mukana.

Lopullinen käyttäjä ei näe ohjelmankehitysympäristöä, vaan pelkän valmiin sovelluksen. Sovellus saadaan tallennettua .RUN-päätteellä varustettuun tiedostoon valitsemalla File-valikosta Make Runtime. Kun tämän jälkeen poistuu SQLWindowsista ja kaksoisklikkaa .RUN-tiedoston nimen kohdalla, käynnistyy SQLRUN.EXE suoritamaan valmista sovellusta edellyttäen, että WIN.INI-tiedoston asetukset ovat ohjeiden mukaiset.

Valmiin sovelluksen toimintaa voi vielä nopeuttaa noudattamalla tarkoin version 1.1.0 mukana tulleita ohjeita ennen Make Runtime -valinnan tekemistä.

File-valikon Save As -kohdan avulla ohjelmaa voidaan tallentaa joko teksti-, binääri- tai käännetyssä muodossa. Tekstimuotoisen ohjelmaa voidaan käsitellä INDENT-apuohjelmalla, jonka jälkeen sitä voidaan käsitellä esimerkiksi jollain tekstintomitti-

Osa 2 — Rättäriklubin jäsenrekisteri

mella. Usein myös omituiset virhetilanteet saattavat selvittää tallettamalla ohjelmaa välillä tekstimuotoon ja lukemalla se uudestaan SQLWindowsiin. Tekstimuotoisen ohjelmatiedoston laajennusosana tulisi käyttää .APT:tä.

Binäärimuotoinen ohjelma on SQLWindowsin sisäisessä esitysmuodossa oleva ohjelma. Laajennusosana tulisi olla .APP.

Käännettyä muodossa ohjelma tallentuu automaattisesti, jos sitä on kerrankin istunnon aikana ajettu valitsemalla Run-valikosta User Mode. Kun User Mode valitaan, SQLWindows kääntää ohjelman keskusmuistissa tarkistaen samalla muun muassa sen, onko kaikki muututtu jätettyä määrittelyä. Jo pelkkä ikkunoiden siirtyä aiheuttaa muutoksia ohjelman Design Time Settings -kohtaan, ja tällöin SQLWindows kysyy, tallennetaanko ohjelma ennen kääntämistä. Muistin tilaongelmien välttämiseksi ja käännöksen nopeuttamiseksi kannattaa kaikki mahdolliset ikkunat sulkea tai tehdä pieniksi ennen käännöksen käynnistämistä.

Jos käännetyn ohjelman laajennusosana on .APP, käynnistyy SQLWIN.EXE-ohjelmankehitysympäristö, jos laajennusosana on .RUN, käynnistyy pelkkä sovellusta suorittamaan SQLRUN.EXE.

SQLWIN.EXE pystyy lukemaan Save As -valikossa määritellyn tyyppisiä tiedostoja ja päättää tiedostosta, mikä edellämämainitun tyyppinen tiedosto on kyseessä. Sen sijaan Make Runtime -valinnan tuottama .RUN-tiedosto ei enää pysty lukemaan SQLWIN.EXE:llä muokatavaksi.

Rättäsitikalla kulkevien kerho

Artikkelisarjan loppuosan vietämme jäsenrekisterisovelluksen parissa. Tarkoituksenamme on saada aikaan kuvassa 2 näkyvä sovellus. Kuvitteelliseksi toimintaympäristöksi on valittu rättäsitikalla kulkevien kerho, jonka jäsentiedot saadaan näin hieman

nykyaikaisemmin välinein varmaan talteen.

Kuvassa 2 näkyy, että tulemme muuttamaan Pääikkunan valikon yksinkertaiseksi Jäsenistö-valikoksi. Sen alta löytyvät vaihtoehdot uuden jäsenen tietojen syöttämiseen ja toisaalta jäsenluettelon tulostamiseen, kuten kuvassa 3 näkyy. Artikkelisarjan tässä osassa teemme vielä keskusteluikkunan, jonka avulla syötetään uuden jäsenen tiedot. Keskusteluikkuna käynnistetään Jäsenistö-valikon vaihtoehdosta Uusia Jäseniä ja sen muoto on nähtävissä kuvassa 5.

Kuvan 2 Pääikkunassa olevat kaksi tärkeää oliota, list box "Valitse Kuljin" ja table window "Jäsenluettelo", lisätään sovelluksemme artikkelisarjan myöhemmissä osissa.

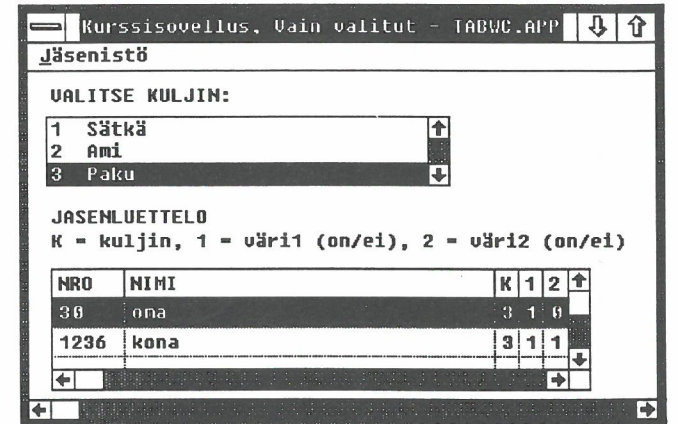
Opetusteknisistä syistä tulemme tekemään myös joitakin lisävaihtoehtoja, jotka eivät näy kuvan 2 lopullisessa sovelluksessa.

Ota nyt siis työn alle joko NEWAPP.APP (muista tallentaa eri nimelle) tai viimekertainen harjoittelupohja. Voit poistaa siinä mahdollisesti olevat ylimääräiset oliot valitsemalla ne joko Outline:sta (näpytys otsikkorivin mantelin kohdalla) tai Pääikkunasta (näpytys, kun kohdistin on nelipäinen nuoli) ja sen jälkeen painamalla poistonäppäintä (del).

Valikko — erilainen olio

Aloitamme esimerkkisovelluksemme rakentamisen muuttamalla NEWAPP.APP:ssa olevan valmiin valikko-ehdotuksen kuvan 3 mukaisesti tarkoitukseemme sopivaksi.

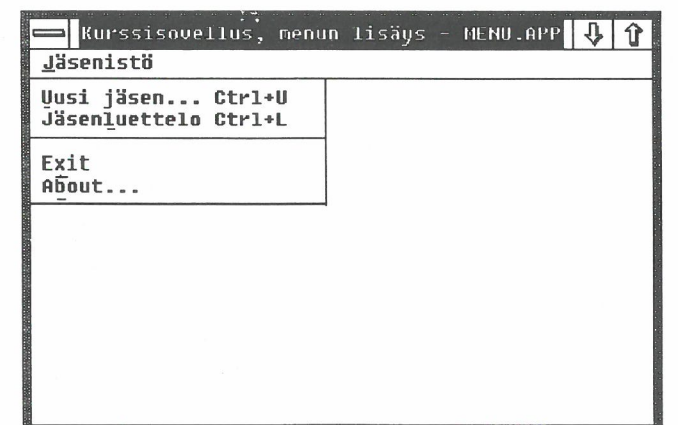
Avaa ensin NEWAPP.APP:ssa oleva Pääikkunan Menu-kohta kokonaan. Sen voit tehdä valitsemalla Menu-rivin ja sitten joko Outline-ikkunan Outline-valikon Expand All Levels-vaihtoehtolla tai sen pikavalinnalla *, joka tarkoittaa numeronäppäimistön yllä olevaa kertomerkkiä. Nämä valikon pikavalinnat, jotka näkyvät rivin loppuosassa, toimivat, kun valikko ei ole näytölle avattuna. Rivien alkuun kirjoitetuissa sanoissa on yleensä jokin kirjain



alleiviivattuna tai erivärisenä. Sitä kirjainta painamalla voi tehdä pikavalinnan, kun valikko on avattuna näytölle. Vastaavasti näytöllä koko ajan näkyvän päävalikon vaihtoehdon voi valita painamalla alt-näppäintä ja valikon nimesä alleiviivattuna tai eriväristä kirjainta.

Jos tarkastellaan yhtä Menu Item -oliota esimerkiksi valitsemalla rivin Menu Item:&Form, huomataan, että valikon vaihtoehdot ovat olioita. Niillä on rakenne- ja toimintaosat. Nämä oliot ovat erilaisen näköisiä verrattuna viime kerran painonappiolioon, sillä niiden toimintaosan otsikkona ei lue Message Actions, vaan pelkkä Actions. Tämä johtuu siitä, että valikosta voidaan vain valita jokin vaihtoehto, eikä se voi saada muunlaisia sanomia kuin "sinut valittiin". Niinpä valikon Actions-otsakkeen alla seuraavalla sisennystasolla EI ole On-lauseita, joilla haaraannuttaisiin saadun sano-

Kuva 2: Esimerkkisovellus, jonka rakennamme.



Kuva 3: Esimerkkisovelluksen valikkorakenne.

Kuva 1: NEWAPP.APP-sovellusrun- gon rakenne.



SarjaFoto

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

man perusteella, vaan toiminnat kirjoitetaan suoraan Actions-otsakkeen jälkeen sisennettynä. Tämän huomaa helposti, jos ottaa View-valikosta vaihtoehdon Options Box ja valitsee jonkin Actions-rivin.

Tulkoon valikko — Ja valikko tuli

Kuva 4: Valikkorakenteen aiheuttamat muutokset ohjelmakoodiin.

Seuraavaksi voit kokeilla NEWAPP.APP:n valikkorakenteen muokkaamista. Yritä esi-

Application Description: SQLWindows-kurssin sovellus
Design-time Settings
Global Declarations
Form Window: MainForm
Title: Kurssisovellus, menun lisäys - MENU.APP
Icon File: (none)
Display Settings
Menu
Popup Menu: &Jäsenistö
Menu Item: &Uusi jäsen...
Keyboard Accelerator: Ctrl+U
Menu Settings
Enabled when:
Checked when:
Actions
Call SalMessageBox('uusi jäsen', 'uusi jäsen valittu', MB_Ok)
Menu Item: Jäsen&luettelo
Keyboard Accelerator: Ctrl+L
Menu Settings
Enabled When:
Checked when:
Actions
Call SalMessageBox('luettelo', 'jäsenluettelo valittu', MB_Ok)
Menu Separator
Menu Item: E&xit
Keyboard Accelerator: (none)
Menu Settings
Enabled when:
Checked when:
Actions
Call SalQuit()
Menu Item: A&bout...
Keyboard Accelerator (none)
Menu Settings
Enabled when:
Checked when:
Actions
Call SalModalDialog(AboutBox, hWndForm)
Contents
Variables
Message Actions
Dialog Box: AboutBox

merkiksi siirtää päävalikon vaihtoehtoja toiseen järjestykseen. Siihen on kaksi tapaa. Voit merkitä jonkin rivin alariveineen ja siirtää sitä samantasoisten olioiden joukossa painamalla alt- ja nuoli ylös -näppäintä. Näin voit myös muuttaa rivien sisennyksiä. Jos haluat siirtää useampia oliota, voit merkitä ensimmäisen ja sen jälkeen vaihtonäppäintä alhaalla pitäen merkitä toisia.

Toinen rivien tai niiden osien siirtämiseen sopiva menetelmä on kopiointi. Ensinnä siirrettävät kohdat merkitään ja sitten valitaan Edit-valikon Copy, merkitään kohta, johon siirretään, ja lopuksi valitaan Edit-valikon Paste. Huomaa, että Paste on harmautettuna, kunnes löydät siirrettävälle tavaralle SQLWindowsin mielestä sopivan paikan. Kokeile siirtää Edit-in avulla jokin Menu Item toiseen paikkaan toisaalta Menu-otsakkeen alle, toisaalta esim. Pääikkunan Contents-kohtaan.

Nyt voit aloittaa kuvan 3 mukaisen valikon tekemisen. Poista valikoista ylimääräiset jättäen vain Form-valikko muokattavaksi. Olion poisto tapahtuu merkitsemällä se (klikkaus Outline:ssa olion ensimmäisen rivin alussa) ja poistonäppäimellä (del). Pääset helposti muokkaamaan uutta otsaketta Menu Itemille näpyttämällä mantelin kohdalla hiiren oikeata näppäintä.

Liitä muokkaamaasi Jäsenistö-valikkoon Uusi jäsen — ja Jäsenluettelo-valintoihin toiminnoksi MessageBox:it. Huomaa, että valinnan perässä tulevat kolme pistettä tarkoittavat, että siitä käynnistyy jokin lisäikkuna, eikä toiminta suoraan. Toistaiseksi korvaamme molempien valintojen toiminnat näillä MessageBoxeilla vain, jotta pystyisimme näkemään jotain toimintaa — esimerkiksi:

Call SalMessageBox('boxin teksti', 'boxin otsake', MB_Ok)

Funktioille kuvaavat nimet

Kaikkia aliohjelmaa kutsutaan SQLWindowsissa funktioiksi C-kielen tapaan, vaikka niiden toiminta olisikin Pascal-kielen proseduurien tapaan eli ne eivät palauta mitään arvoa, vaan vain suorittavat jonkin tehtävän.

Funktioiden kirjoittamista helpottaa tieto siitä, että niiden, kuten muuttujienkin nimet, kirjoitetaan pääsääntöisesti pienillä kirjaimilla, mutta nimen sisällä sanat alkavat isoilla kirjaimilla. SAL-ohjelmassa esiintyvät isokirjaimiset sanat ovat siis yleensä vakioiden nimiä.

SQLWindowsissa on funktioilla aina alkuosana joko Sal (Sql Application Language) tai Sql (vain tietokannan käsittelyyn liittyvä funktio). Näiden funktioiden esittelyrivit löytyvät Global Declarations -kohdasta otsikon External Functions alta. Funktiot, joiden alku on jokin muuta, ovat joko ohjelmioijan itsensä kirjoittamia (Internal Functions) tai sitten dynaamiseen kirjastoon vietyjä funktioita (External Functions).

Etsi External Functions -kohdasta SalMessageBox-funktion esittelyrivit ja avaa ne kokonaan näkyviin. Kohta Parameters kertoo, minkä tyyppisiä ja montako parametria tulee funktion kutsussa antaa. Se siis ottaa vastaan string-tyyppisen tekstin, sitten string-tyyppisen otsakkeen ja vielä numeron, joka kertoo esimerkiksi, että halutaan painonappi OK mukaan. Tämä MB_Ok siis on oikeastaan jokin vakionumero, ja sen määrittely löytyy kohdasta Constants: System. Vilkaise myös sen määrittelyä.

SalMessageBox-funktion esittelyriveissä, kohdassa Description, on malliesimerkki kutsusta. Yleensä voit suoraan kopioida sen Edit-valikon avulla ohjelmaasi. Toisella rivillä olevan esimerkkikutsun merkitseminen käy pai-

namalla hiiren näppäin alas nResult-sanan alussa ja pitämällä näppäintä alhaalla vetäen sitä samalla esimerkkikutsun loppuun asti. Näppäimen päästäessä kyseinen kutsu on merkitty. Sitten vain Edit-valikon kimppuun.

Paste-vaiheessa saattaa ilmetä ongelmia, sillä SQLWindows vaatii kopioitavalle kutsulle juuri sopivan paikan. Tässä tapauksessa tulisi kirjoittaa valmiiksi (Outline Options:in avulla) toimintakohtaan rivin alkuun Set-lause ja saada kohdistin pystyviivaksi Set-sanan ja välilyönnin jälkeen.

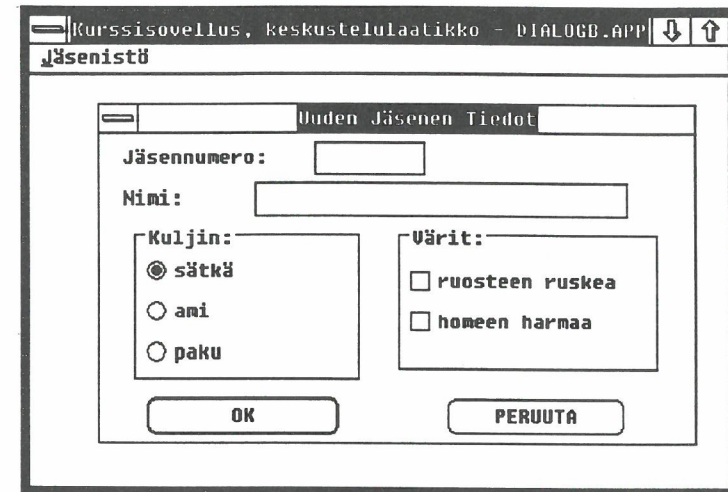
Funktioita voidaan kutsua kolmella eri tavalla. Ensinnäkin Call-lauseella, kuten olemme tehneet. Toiseksi, jos funktion esittelyriveillä kohdassa Returns on mainittu, minkätyyppistä tavaraa funktio palauttaa arvonaan, voidaan sitä vaihtoehtoisesti Call-tavan sijaan kutsua vielä kahdella eri tavalla. Tällöin funktiota voidaan kutsua vaikkapa Set-lauseella, jolloin funktion palauttama arvo asetetaan jonkin samantyyppisen muuttujan arvoksi myöhemmin testattavaksi. Tai funktion kutsu voidaan kirjoittaa suoraan esimerkiksi If-lauseen testin osaksi.

Tallenna nyt ohjelmasi binäärimuodossa tiedostoksi MENU.APP (File: Save As). Testaa sitten ohjelmasi toimintaa: Run-valikosta ensin Slow Animate päälle, sitten käynnistys: User Mode.

Valikoissa on myös lisäpiirteitä, kuten niiden harmauttaminen tai ruksittaminen, joihin voit tutustua toimituksen mukana tulevien esimerkkiohjelmien avulla. Kuvassa 4 näet tämänkertaisen tehtävämme esimerkkiratkaisun.

Keskusteluikkuna välittää tietoa

Nyt olemme saaneet esimerkkisovelluksemme valikkorakenteen kuntoon. Seuraavaksi lähdemme rakentamaan toimintaa valikon vaihtoehtoon Uusia Jäseniä.



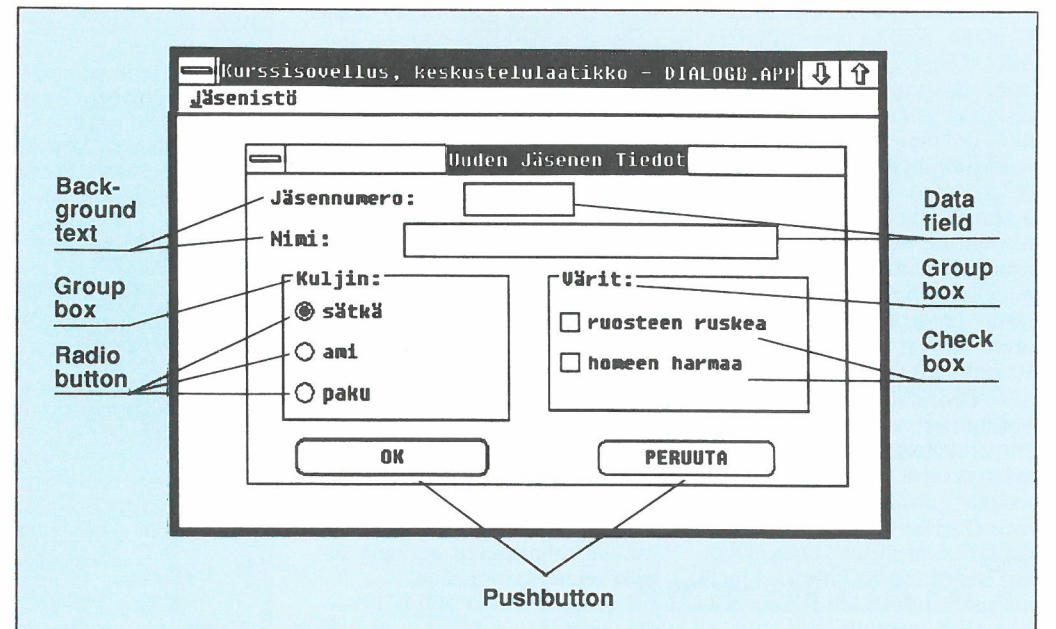
Kuva 5: Rättärikubisovelluksen keskusteluikkuna.

Uuden jäsenen tietojen syöttöön sopii hyvin keskusteluikkuna, joka tullaan käynnistämään edellämainitusta valikon vaihtoehtosta. Piirrämme ensin keskusteluikkunan, minkä jälkeen liitämme sen osaksi muuta ohjelmaa. Seuraavaksi kerraksi jää keskusteluikkunan kenttien tarkempi käsittely.

Huomaa, että keskusteluikkunan määrittelyrivit alkavat samalta sarakkeelta kuin Pääikku-

nan eli ne ovat samantasoisia oliota. Voit tarkastella keskusteluikkunan liittämistapaa muuhun ohjelmaan vaikkapa avaamalla NEWAPP.APP:sta kohdat Menu Item:About ja Dialog Box: AboutBox. Missään kohdassa ei näy koodia, jolla käsiteltäisiin keskusteluikkunan piirtämistä, kenttien syöttövaihetta tai ikkunan poistamisvaihetta näytöltä, vaan Windows hoitaa nämä tehtävät pääosin itse.

Kuva 6: Keskusteluikkunan kenttien tyypit ja nimiehdotukset.



Keskusteluikkunalle myös täytettä

Kuvassa 5 näet keskusteluikkunan, jollaisen pitäisi ilmestyä näytölle, kun käyttäjä valitsee Jäsenistö-valikosta vaihtoehdon Uusi jäsen...

Muuta sovelluksen otsikko ja tallenna sovellus nimelle DIA-LOG.APP.

Määrittele kuvassa näkyvä keskusteluikkuna: Ota ensin View-valikosta Outline Options näkymiin. Kokeile, minkälainen olio Outline:ssa tulee olla valittuna, jotta Outline Options antaisi vaihtoehdoissaan keskusteluikkunan.

Muokkaa keskustelulaatikko sopivan kokoiseksi. Myöhemmin sen kokoa voi muuttaa vain antamalla Outlineen uudet arvot sen window location & size -kohtaan. Tarvittaessa saat keskusteluikkunan näkyviin valitsemalla sen määrittelyrivin (Dialog Box: UusiJasen) Outline:sta ja sitten View-valikosta Show Window.

Piirrä keskusteluikkunan sisälle olioita valitsemalla niitä Draw-valikosta ja piirtämällä tällä kertaa keskusteluikkunaan. Voit käyttää olioille kuvassa 6 esitettyjä nimiä, vaikka parempi tapa olisi nimetä ne unkarilaisittain (esim. check box "satka" nimitäisiin cbSatka, data field "nimi" fldNimi jne.). Kuvasta 6 ilmenevät myös kenttien tyypit. Tarkkaile samalla, mihin kohtaan Outline:n ohjelmaa olioiden koodi sijoittuu, valitsemalla olio keskusteluikkunasta, kun kohdistin on nelipäisen nuolen muotoinen. Täytä tarvittavat kohdat Outlineen jättäen kuitenkin Message Actions -kohdat tyhjiksi.

HUOM. Olioiden sijoittelu ja syöttöjärjestys kannattaa suunnitella etukäteen ja piirtää syöttöjärjestyksessä. Piirtämisjärjestys määrää olioiden järjestyksen myös Outline:ssa, joka taas määrää syöttämisjärjestyksen. Oliot ovat siirreltävissä Outlinessa merkitsemällä olio ja sen jälkeen Alt- ja nuolinäppäimillä.

Liitetään keskusteluikkuna valikkoon lisäämällä sen Uusia Jäseniä -vaihtoehtoon funktion SalModalDialog kutsu. Toisaalta keskusteluikkuna pyydetään poistamaan näytöltä, kun käyttäjä painaa OK- tai Peruuta-painonappia. Poistaminen tapahtuu kutsumalla funktiota SalEndDialog. Kutsujen muodot ovat seuraavat:

Call SalModalDialog
(UusiJasen, hWndForm)
Call SalEndDialog (hWndForm, 0)

joissa UusiJasen on käynnistettävän keskusteluikkunan nimi eli template (riviltä Dialog Box: UusiJasen). hWndForm on ensimmäisessä kutsussa käynnistettävän keskusteluikkunan nimi eli kahva (hWndForm on määritetty valmiiksi Global Section: Variables -kohdassa) ja toisessa kutsussa Windows on hoitanut sille arvoksi keskusteluikkunan kahvan, sillä SalEndDialog-kutsu voi esiintyä vain lopetettavassa keskusteluikkunassa (suoritettava siinä viimeisenä toimenpiteenä). Jälkimmäisen kutsun toinen parametri eli 0 tai jokin muu numero välittyy käynnistäjälle funktion SalModalDialog paluuarvona. Sitä voitaisiin tutkia siellä, jos kutsu alkaisi sanoilla If tai Set. Näin saadaan selville esimerkiksi painoiko käyttäjä OK- vai Peruuta-nappia.

Testaa ohjelmasi toimivuutta! Jos sinulla on ongelmia sen kanssa, voit tutkia kuvassa 7 olevaa malliratkaisua!

Jatkuu lokakuussa

Seuraavassa osassa opimme lisää keskusteluikkunan kenttien käsittelystä, SQL-lauseiden upottamisesta SAL-kieliseen ohjelmaan ja list box:ista. Tällä välillä voit jo tutustua esimerkiksi keskusteluikkunoiden käyttöön tarkastelemalla toimituksen mukana tulevia esimerkkiohjelmaa.

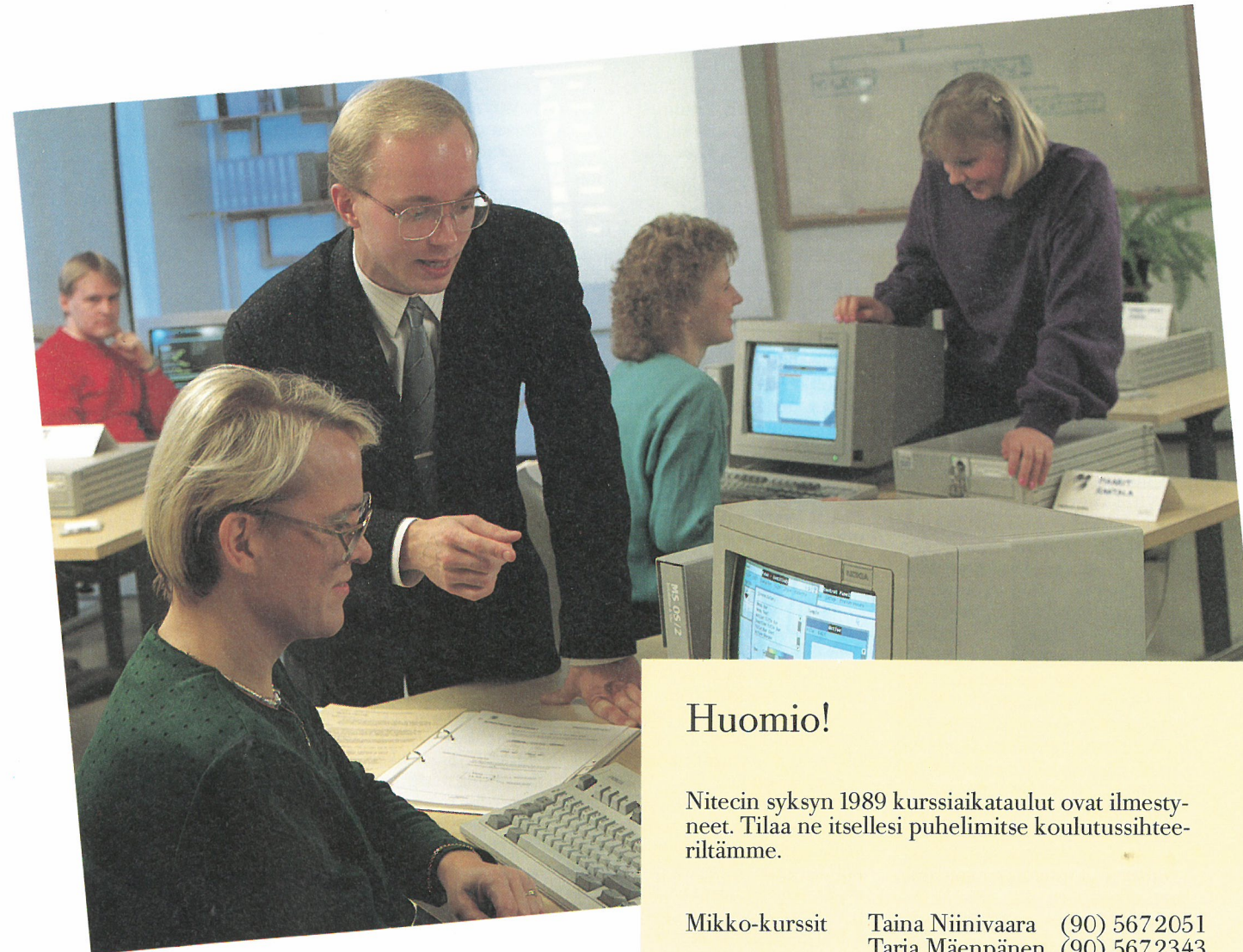
Hauskoja hetkiä SQLWindow-sin parissa!

```
Application Description: SQLWindows-kurssin sovellus
Design-time Settings
Global Declarations
Form Window: MainForm
Title: Kurssisovellus, keskustelulaatikko - DIALOG.APP
Icon File: (none)
Display Settings
Menu
  Popup Menu: &Jäsenistö
    Menu Item: &Uusi jäsen...
      Keyboard Accelerator: (none)
      Menu Settings
        Enabled when:
        Checked when:
      Actions
        Call SalModalDialog(UusiJasen, hWndForm)
    Menu Item: Jäsen&luettelo
      Keyboard Accelerator: (none)
      Menu Settings
        Enabled when:
        Checked when:
      Actions
        Call SalMessageBox('luettelo', 'Jäsenluettelo valittu', MB_Ok)
    Menu Separator
    Menu Item: E&xit
    Menu Item: A&bout
Contents
Variables
Message Actions
Dialog Box: AboutBox
Dialog Box: UusiJasen
Title: Uuden Jäsenen Tiedot
Display Settings
Contents
  Background Text: Jäsennumero:
  Window Location and Size
  Justify: Left
  Background Text: Nimi:
  ...
  Data Field: nro
  Data
  Display Settings
  Message Actions
  Data Field: nimi
  ...
  Group Box: Kuljin:
  Window Location and Size
  Radio Button: satka
  Title: satka
  Window Location and Size
  Message Actions
  Radio Button: ami
  ...
  Radio Button: paku
  ...
  Group Box: Värät:
  Window Location and Size
  Check Box: ruoste
  Title: ruosteen ruskea
  Window Location and Size
  Message Actions
  Check Box: home
  ...
  Pushbutton: OK
  Title: OK
  Window Location and Size
  Keyboard Accelerator: Enter
  Message actions
  On SAM Click
  Call SalEndDialog(hWndForm, 1)
  Pushbutton: Cancel
  Title: PERUUTA
  Window Location and Size
  Keyboard Accelerator: (none)
  Message Actions
  ON SAM Click
  Call SalEndDialog(hWndForm, 2)
Variables
Message Actions
```

Mikko

Kurssiaikataulu

Syksy 1989



Huomio!

Nitecin syksyn 1989 kurssiaikataulut ovat ilmestyneet. Tilaa ne itsellesi puhelimitse koulutussihteeriltämme.

Mikko-kurssit	Taina Niinivaara	(90) 567 2051
	Tarja Mäenpänen	(90) 567 2343
Tietoliikenne ja DPS-kurssit	Pia Kaartinen	(90) 567 2545
UNIX-, SUN-, S2500- ja Tandem-kurssit	Kirsi Stubb	(90) 567 3518

Microsoft Universityn valtuuttama koulutuskeskus

Tietotekniikan koulutuskeskus NITEC

NOKIA DATA
Tietoajon kärjessä

Ilmoitukset taittavat Marilla

Mari on SUNin ja Apollon työ-asemissa pyörivä ilmoitusten ja graafisten kuvioiden luomiseen soveltuva WYSIWYG-taitto-ohjelma. Marin suunnittelussa on pantu erityistä painoa hankalien ilmoitusten käsittelyyn eli esimerkiksi sävykuvien, monimutkaisten sävyalueiden ja kierretyn tekstin määrittely onnistuu helposti.

Tulostus tapahtuu PostScript-kielellä. Ilmoitus voidaan laatia ideasta painovalmiiksi yhdellä ohjelmalla.

Lisätietoja: Pekka Kekolahti, puh. (90) 1241.

Verkkotuotepalvelut monipuolistuvat

Vuonna 1988 yritysten sisäiset tietoliikenneverkot yhdessä sähköpostin kanssa olivat kaksi eniten yleistynyttä atk-järjestelmää. Kehityksen seurauksena Nokia Datan asiakaspalvelut-yksikkö on kehittänyt uuden Verkkotuotteiden tukipalvelut -kokonaisuuden.

Kokonaisuus koostuu kahdesta osasta, **sopimuspalveluista** ja **asiantuntijapalveluista**. Sopimuspalvelut rakentuvat neljästä erilaajuisesta kokonaisuudesta, jotka kukin sisältävät kaikki alemman tason palvelut.

Alimmalla tasolla on **puhelinpalvelu**, jonka kautta asiakas saa neuvoja verkko-ohjelmistoista. Lisäksi hänelle toimitetaan postitse ohjelmistojen uusimmat päivitysversiot.

Seuraavalla tasolla on **peruspalvelu**, joka sisältää puhelinpalvelun lisäksi asiakasvierailut ohjelmistovikojen ilmetessä sekä tiedotuksia uusista ohjelmistotuotteista.

Päivityspalvelu sisältää myös ohjelmistopäivitysten testauksen, ohjelmistotiedotepalvelun sekä tarvittavat ohjainkorttien EPROM-päivitykset. **Laajennettu palvelu** sisältää lopulta tilattujen ohjelmistopäivitysten asennuksen, kokoonpanon ja testauksen.

Kokonaisuuden toinen puoli — verkkotuotteiden asiantuntijapalvelu — sisältää puolestaan ohjelmistopäivitysten asennuksen lisäksi työryhmäverkkojen ja lähiverkkojen loogisen suunnittelun ja toteutuksen sekä verkon laajennusten suunnittelun ja verkkojen suojausten hyväksikäytön. Lopuksi annetaan vielä käyttöneuvontaa.

Esiasennus lisää toimintavarmuutta

Nokia Datan asiakaspalvelut-yksikkö on perustanut esiasennuskeskuksen tietokonetehtaan kupeeseen Espoon Kiloon.

Esiasennus tarkoittaa käytännössä sitä, että tilattu järjestelmä asennetaan, konfiguroidaan ja testataan ennen toimitusta. Testaus nopeuttaa lopullista asunnustyötä ja ehkäisee ongelmia.

Lisätietoja: Lasse Yrjänä, puh. (90) 5691.

DPS 6 PLUS kehitty

DPS 6 PLUS -järjestelmien käyttöjärjestelmäksi on kehitetty levy pohjainen monen käyttäjän monen tehtävän, virtuaalimuistia ja monisuoritinympäristöä hyödyntävä HVS 6 PLUS. Siihen saatavat tietoliikenneohjelmistot mahdollistavat suurtenkin DPS 6 PLUS -verkkojen toteuttamisen sekä liittymisen OSI- ja SNA- ympäristöihin.

OSI/DSA on puolestaan ohjelmisto, jolla voi rakentaa tietokoneverkkoja, jotka koostuvat DPS 6 PLUS, DPS 6 ja GCOS 8 -järjestelmistä sekä muista OSI-suosituksia noudattavista järjestelmistä. Kaikki sovellustason palvelut ovat käytettävissä lähiverkon, suoran yhteyden ja pakettiverkon kautta.

DPS 6 PLUS -järjestelmille on kehitetty myös oma lähiverkko-ohjain, jolla ne saadaan kytkettyä suoraan Ethernet-työryhmäverkkoon. Verkkoon liitetyt mikrotietokoneet voivat käyttää DPS 6 PLUS -järjestelmien levyasemia ja tulostimia kuin omia paikallisia oheislaitteitaan.

Yritysgrafiikka Windowsissa

Graph Plus on Windows-ympäristössä toimiva yritysgrafiikkaohjelma, jonka tulostukseen sopivat kaikki Windows 2:n ymmärtämät oheislaitteet.

Ohjelma piirtää erilaisia pylväs-, viiva-, alue-, piirakka-, piste- ja taulukkokuvaajia joihin voi lisätä tekstejä, laatikoita, nuolia ja Designer-kuvia. Kuvaajat voidaan piirtää joko kaksi- tai kolmiulotteisina regressioineen kaikkineen.

Graph Plussalla tehtyt kuvat voidaan siirtää mm. Designeriin, Windows Writeen, Drawiin ja PageMakeriin. Lähtötietoja se

Asiakaspalvelukeskus on muuttanut

Nokia Datan Espoon Haukilahdessa sijainnut asiakaspalvelukeskus on muuttanut Valimotie 21:een Helsingin Pitäjänmäkeen.

Myös puhelinnumerot ovat muuttuneet siten, että kaikki (90) 124-alkuiset palvelunumerot ovat poistuneet käytöstä. Uusi palvelunumero on (90) 567 3700.

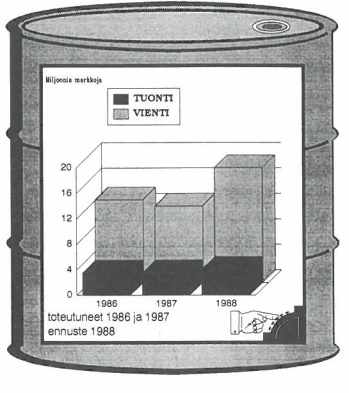
Alfaskop-päätejärjestelmä-valikoima laajenee

Nokia Data täydentää Alfaskop System 91 -päätejärjestelmänsä Alfaskop System 74 -järjestelmällä. Uusi järjestelmä on liitännäisyhteensopiva IBM 3174 -tuotteiden kanssa eli Alfa 74 -ohjaimiin voidaan liittää myös muiden toimittajien päätteitä.

Alfaskop System 74 koostuu Alfa 74 -pääteohjaimista, VDU 191 ja VDU 192 -päätteistä sekä tulostimista. Paikallisohjainmalleja on yksi ja etäisohjaimia on valittavana useita käyttötavan, päätemäärän ja liitännätavan mukaan. Etäisohjain voidaan liittää Token Ring -verkkoon, jolloin siirtonopeus on neljä megabittia.

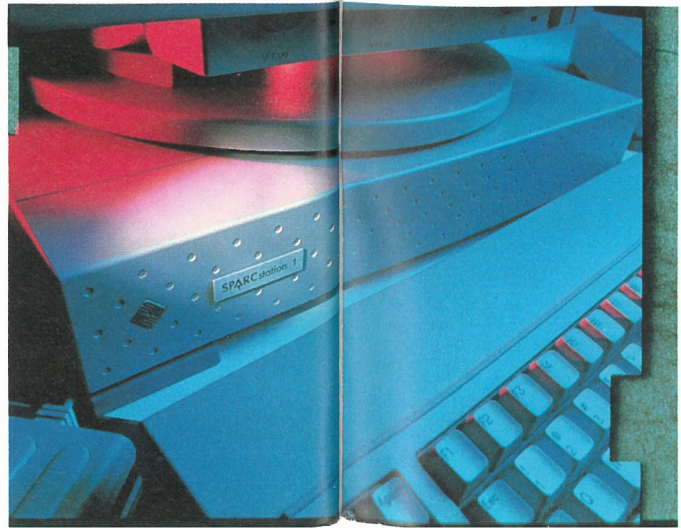
Useimpiin ohjaimiin voidaan liittää 32 koaksiaaliliitäntäistä laitetta ja mallista riippuen kaksi, kahdeksan tai 24 ASCII-päätettä. Optiona on saatavissa myös Token Ring Gateway, jonka avulla Token Ring -verkon työasemat voivat olla ohjaimien kautta yhteydessä keskuskoneeseen.

Lisätietoja: tuotepäällikkö Jarmo Häkkinen, puh. (90) 5671



saa esim. Lotuksesta, Multiplanista ja DIF-tiedostoista.

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



Uusi UNIX tuoteperhe Mikko 100:aan

Nokia Data on ryhtynyt markkinoimaan uutta PK-yrityksien tietojenkäsittelyyn tarkoitettua Mikko 100 UNIX-tuoteperhettä.

Mikko 100 -koneissa on Intelin 80386-suoritin ja SCO Xenix System V, joka voidaan päivittää V/386:ksi ilman, että sovelluksiin tarvitsee tehdä muutoksia.

Järjestelmässä on mukana sekä 5,25" (1,2 Mt) että 3,5" (1,44 Mt) levykeasemat. Liitäntöjä varten laitteistosta löytyy vakiona yksi Centronics-rinnakkaisportti ja neljä tahdistamattonta RS232-sarjaporttia.

Malli 52:n AT-väylässä on vapaana viisi pitkää korttipaikkaa. Suorittimen kellotaajuus on 20 MHz, ja suorittimen tueksi voidaan liittää 20 MHz:n 80387 tai Weitekin aritmetiikkasuoritin. Muistivaihtoehdot ovat 4, 8 ja 16 megatavua.

Malli 55:n yhdestä 8-bittisestä, viidestä 16-bittisestä ja kahdesta 32-bittisestä korttipaikasta SCSI-

Uudet Sun-tuotteet markkinoille

Uutuuksissa on sekä työasemia että Sun-työasemaverkon palvelimia, ja ne ovat kaikki yhteensopivia aiempien Sun-tuotteiden kanssa.

Kenties mielenkiintoisin uutuus on SPARCstation 1 -työasema, joka tuo RISC-työasemat uuteen hintaluokkaan. Enemmän SPARC-station 1:stä alla olevassa uutisessa.

SPARC-tuoteperheen SPARC-arkkitehtuuri on muutenkin muotoutumassa RISC-teknolo-

Hintaluokkansa tehokkain työasema

SPARCstation 1 on Sunin uusi RISC-pohjainen UNIX-työasema, jonka tietojenkäsittelyteho on vakuuttavat 12,5 MIPSiä ja 1,4 MFLOPSia.

Uutuus vastaa teholtaan kahdetoista VAX 11/780-tietokonetta. Työmuistia on kahdeksan megatavua ja se on laajennettavissa 16 megatavuun. Liitäntäpuolella löytyy kaksi sarjaliitäntää sekä Ethernet ja SCSI.

ohjain ja lisätoiminneohjain (sallasanat ja LCD) varaavat kaksi 16-bittistä korttipaikkaa, muut ovat vapaita. Kellotaajuus on 25 MHz, ja keskusyksikkö on varustettu 64 kilotavun pikamuistilla (cache). Aritmetiikkasuorittimena voidaan käyttää joko 25 MHz:n 80287:ää tai Weitekin aritmetiikkasuoritinta. Muistia voi olla 8, 16 ja tuonnempana 24 megatavua.

Malliston käyttöjärjestelmänä toimii SCO Xenix System V ja jatkossa Nokia SCO UNIX System V/386. Lisäksi saatavana on UNIX System V versio 3.2 -kehitysympäristö C-kääntäjiineen ja muine apuvälineineen.

Heti saatavilla olevia sovelluksia ovat muiden muassa Kaptah (terveysasemat), Dens (hammaslääkärit), Työkalupakki (tuotannonohjaus), Intime Open (yritysohjelmistot), Terttu (yritysohjelmisto) ja Reila (kunnossapito). Lisää sovelluksia kehitetään kaiken aikaa.

Lisätietoja: Matti Virtanen, puh. (90) 5671

gian teollisuusstandardiksi. SPARCsystem 300 -sarjan työasemat ja palvelimet ovat 16 MIPSeineen Sunin nopeimpia koneita.

Uusien Sun 3 -tuoteperheen laajennusten lisäksi markkinoille tulee joukko grafiikkatuotteita: GX- ja GXP-grafiikkakihihdyttimet, PHIGS+- ja GXP-grafiikkakirjastot sekä TAAC-ohjelmisto.

Käyttöliittymät noudattavat OPEN LOOK -stadardia, jonka lisäksi saatavilla on Open Windows Desk Set -apuohjelmistot.

Lisätietoja: tuotepäällikkö Pertti Manner, puh. (90) 5671

GX-grafiikkakihihdyttimellä varustettuna SPARCstation 1 pystyy piirtämään 400 000 yksiuulotteista ja 175 000 kolmiulotteista vektoria sekunnissa. Sen grafiikka on siis noin kymmenen kertaa nopeampaa kuin saman hintaluokan teknisissä työasemissa.

SPARCstation 1 osaa käsitellä myös digitoitua ääntä sekä pyörittää DOS-ohjelmistoja optiona toimitettavan DOS Windows-emulointiohjelmiston avulla.

Lisätietoja: tuotepäällikkö Pertti Manner, puh. (90) 5671

Tandemin suorituskyky/hinta-suhde paranee

Nokia Data tuo Suomen markkinoille Tandemin uudet NonStop CLX 700 -järjestelmät. Uudet järjestelmät ovat NonStop CLX 720, 740, 760 ja 780. Kussakin on vastaavasti kaksi, neljä, kuusi tai kahdeksan suoritinta. Jokaisessa suorittimessa on kahdeksan megatavua muistia, joka voidaan laajentaa 12 tai 16 megatavuun. 700-sarjan rinnalla markkinoille tulee uusi CLX 600 -sarja — CLX 601E ja 602E, joissa on vastaavasti yksi tai kaksi keskusyksikköä. Aiemmista yhden tai kahden suorittimen järjestelmistä ne eroavat 30 % edullisemman hintansa puolesta. CLX 601E maksaa 310 050 markkaa ja CLX 602E 462 150 markkaa.

Lisätietoja: tuotepäällikkö Timo Itälä, puh. (90) 5671

Designer päivitty

Designerista on julkaistettu versio 2.0, joka sisältää runsaasti uusia piirteitä.

Uusi versio käyttää Bézier-käyriä esimerkiksi vapaalla kädellä piirtämiseen. Lisäksi olioiden väritys voidaan määritellä portaattomasti. Ohjelma myös muuntaa pistekuvat automaattisesti vektorimuotoon, kun toiminto aiemmassa versiossa oli valinnainen.

Designer 2.0:n mukana tulee ajonaikainen Windows 2.1. Windowsin laiteajurien lisäksi Micrografx toimittaa omat ajurinsa, jotka ovat monesti kehittyneempiä kuin Windowsin standardi-ajurit.

Designer 2.0 osaa hyödyntää expanded-muistia.

Lisätietoja: Antoine Trux, puh. (90) 567 2136.

Word 4.0

Microsoftin Word-tekstinkäsittelyohjelman versio 4.0 on saatavissa sekä suomen että englantinkielisenä.

Perustoimintojen lisäksi uusi versio osaa käyttää tyylietiedostoja ja sanastokirjastoa, joiden avulla toistuvien samantyyppisten dokumenttien käsittely nopeutuu huomattavasti.

Laajojen dokumenttien hahmottamista helpottaa erityinen jäsennystoiminto, jonka avulla asiakirjasta näkyvät vain halutut otsikkotasot.

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

00003 HELSINKI

Sopimus 00380/166
VASTAUSLÄHETYSPL 780/10
Softari
Nokia DataNokia
maksaa
postimaksunSoftari 2/89
Palvelukortti

Softari-kysely

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

	–	0	+	Tahdon tietoa
Käyttäjät resurssien tasalle	☐	☐	☐	☐
Enemmän kuin käyttöliittymä	☐	☐	☐	☐
Pankkijärjestelmän sydän	☐	☐	☐	☐
Lyhyt historia – pitkä tulevaisuus	☐	☐	☐	☐
Välkky, ei välkä, välkky, ei välkä	☐	☐	☐	☐
90-luku voi alkaa	☐	☐	☐	☐
Muistia millä mällätä	☐	☐	☐	☐
Ventura laajeni uusiin ympyröihin	☐	☐	☐	☐
Puhuu Suomea ja ymmärtää hiirtä	☐	☐	☐	☐
Pureeko se	☐	☐	☐	☐
Learning Ada – Part 4	☐	☐	☐	☐
Tien päällä taas	☐	☐	☐	☐
Verkkotietokanta on 3:n kauppa	☐	☐	☐	☐
Excel oppooa järjestelmiin	☐	☐	☐	☐
SOLWindows-kurssi – Osa 2	☐	☐	☐	☐
Tuoteuutiset	☐	☐	☐	☐

Vastanneiden kesken arvotaan viisi MM4-T-paitaa.

Jos voitaa paidan, kokoni on: _____

- ☐ Korjaan yllämainitut osoitetiedot.
☐ Poistan yllämainitut osoitetiedot
☐ Lisään allamainitut osoitetiedot
☐ Softarin ☐ NETin rekisteristä/rekisteriin

yritys _____

osasto _____

lähiosoite _____

postinro _____ postitoimip. _____

nimi _____

JOUKKOJULKAISU



8/0010

MIKKONEN

