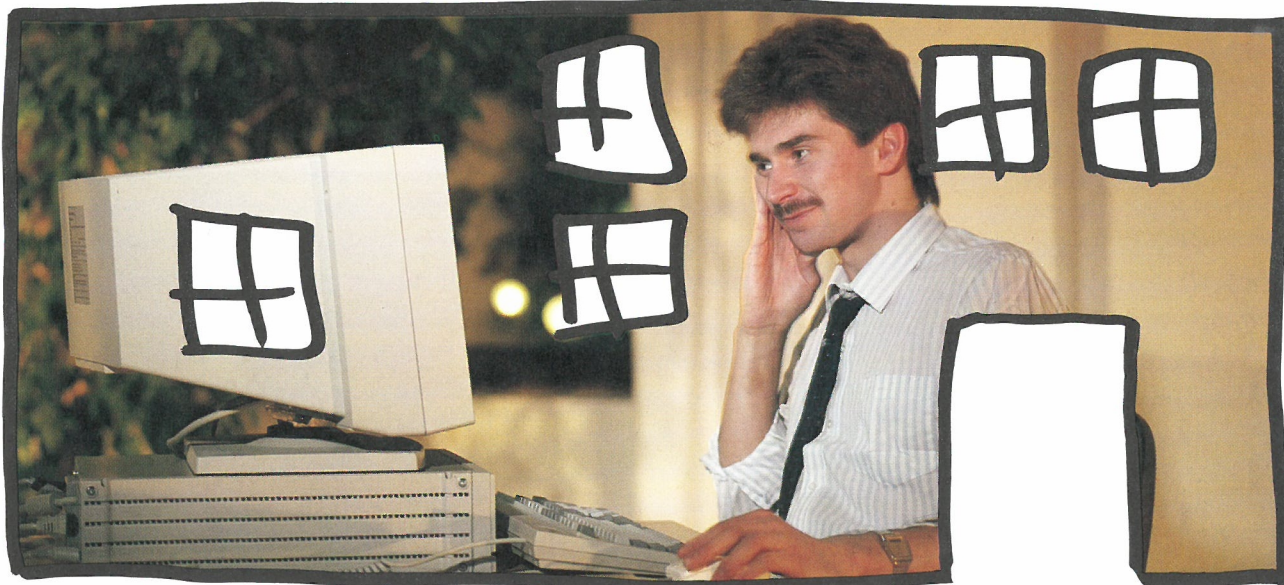


Softari

NOKIA DATAN

OHJELMISTOLEHTI 1/89



**Työryhmäverkko yhdistää
ja palvelee**

Nokia Datan ohjelmistolehti

Vastaavapäätöimittaja:

Anne Typpö

Toimitus:

Jyrki J. J. Kasvi

Softaritiimi:

Raija Visamo, Kirsti Lindfors,
Kari Kaipainen, Tapio Sulkava,
Olli Nurminen, Anne Typpö,

Ulkoasu ja kuvitus:

Taina Sonninen

Osoitteenmuutokset:

Takakannen palvelukortilla

Toimituksen osoite:

Nokia Data

Softari

PL 780

00101 HELSINKI

Toimituksen puhelin:

(90) 567 4064

Painosmäärä:

12 000

Kansi:

Kuvakollaasi, Taina Sonninen

Kirjapaino:

Vientipaino

Julkaisija:

Nokia Data

ISSN 0782-8209

Sisälllys:

Kevättä ilmassa	2
Tiimistä tehoa työryhmiin	3
Nokian ensimmäinen MS OS/2 -sovellus	4
Tieto kulkee verkossa	6
AACS-yhdyskäytäväohjelmisto	8
Partek vastaa ajan haasteisiin	10
Learning Ada Part 3	12
Pieni mutta hiljainen	16
Muistipulmat unholaan	17
Monipuolinen dBASE IV	19
Julkaisuutoimintaa verkossa	21
Lahti MM89 tietokoneen silmin	23
Saamisen Tapsan muistelmät - "Petissä"	25
SQLWindows-kurssi Osa 1	27
Tuoteuutiset	31
Softaaja	32

Kevättä ilmassa

Kevät on uudistumisen aikaa niin luonnossa kuin Softarissakin. Anne Typön työkiireiden lisäaikaan remmiin tarvittiin uutta verta ja avuksi tulikin minä; kynäilevä tietokoneihminen, joka ei vielä tiedä tuleeko hänestä isona lehtimies vai atk-alan ammattilainen. Tämän parempaa tapaa yhdistää molemmat kutsumuksen maailmasta tuskin löytyy.

Myös työryhmäverkot kokevat uudistua kevään myötä. Nokia Datan ToimistoTiimi-työryhmäverkkojärjestelmä siirtää nimittäin Nokian kertaaheittotila MS OS/2 -aikaan. Kerromme tässä numerossa uudistuneesta Tiimistä sekä sen mukanaan tuomista mahdollisuuksista.

ToimistoTiimin lisäksi tuustumme Interleafin TPS-julkaisujärjestelmän uusimman versioon. TPS kun on suunniteltu erityisesti verkkokäyttöön, ja mahdollistaa samojen dokumenttien käsit-

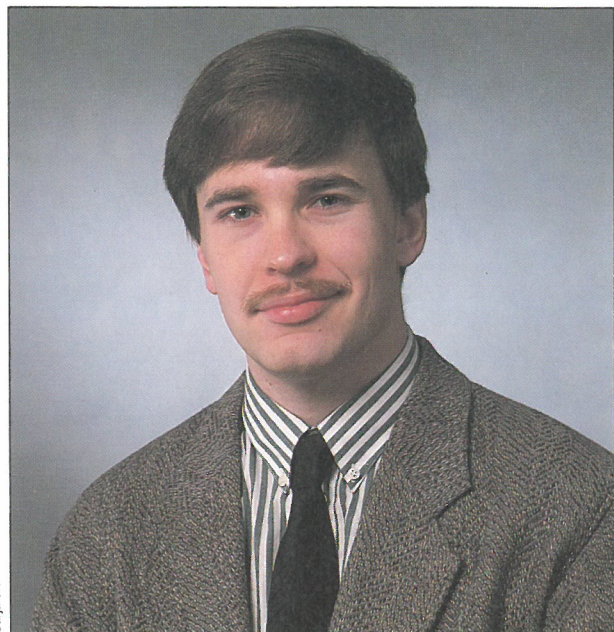
telyn useilta eri tyyppisiltä työasemilta käsin yhtä aikaa.

Jotta verkkoasiana tulisi varmasti tarpeeksi, käymme lopuksi tutustumassa Partekin verkkoratkaisuun, josta he ovat syystä ylpeitä.

Meillä on mukana myös uusi SQLWindows-kurssi, joka jatkuu koko vuoden. Taivoitteenamme on survoa neljään artikkeliin aiheesta kaikki olennainen...

Softarille on nyttemmin perustettu myös oma ilmoitustaulu Nokia Datan Smail-sähköpostiin. Taulun kautta on mahdollista antaa palautetta lehden artikkeleista ja keskustella myös tulevien numeroiden sisällöstä. Taulun käyttöoikeudet saa lähettämällä postia tunnukselle: Kasvi-Jyrki-FRL. Smailin ulkopuolelta yhteyden saa Elisan kautta. Myös kaikki muu palaute on tervetullutta, ja sen voi osoittaa toimitukseen allekirjoittaneelle.

Jyrki J. Kasvi



Sarjafoto

ToimistoTiimi



Risto Laine

Työryhmäverkon tulee tarjota ensisijassa keinoja ryhmän sisäisen tietoliikenteen kehittämiseen verkon ulkopuolelle suuntautuvaa kommunikointia kuitenkin unohtamatta. Uudistunut Tiimi pyrkii tähän uusilla MS OS/2 -pohjaisilla verkkopalvelimilla, jotka huolehtivat paitsi ryhmän sisäisestä sähköpostista myös erilaisista tietoliikenneyhteyksistä ulkopuoliseen maailmaan.

Verkot yhteen

ToimistoTiimi sisältää joukon uusia tietoliikennevaihtoehtoja erityisesti SNA-maailman suuntaan. Myös verkkojen liitettävyyden paraneen lähi-Netbios-liittynän (Netbios Interconnect Program) myötä. Erilliset verkot voidaan liittää näin yhteen, vaikka ne olisivatkin eri tyyppisiä. Lisäksi fyysisesti erillään olevat verkot saadaan yhteen Remote Netbiosin avulla.

Nokia MS Lan Manager vastaa puolestaan Tiimin toiminnallisuuden vahvistamisesta. Lan Managerin ikkunoidun käyttöliittymän ansiosta verkkoa on aiempaa helpompi hallita, minkä lisäksi tietoturvaan on kiinnitetty erityistä huomiota.

Uudet MS OS/2 -palvelimet toimivat moniajoperiaatteella, mikä osaltaan nopeuttaa käyttäjän saamia palveluja.

Tiimi-työryhmäverkkojen tuotepäällikkö Olli Nurminen ToimistoTiimin julkistustilaisuudessa Kirsti Lindforsin kanssa, joka vastaa Tiimin toimisto ohjelmistoista.

Tiimistä tehoa työryhmiin

Työryhmien kommunikoinnista yli kaksi kolmasosaa tapahtuu ryhmän sisällä, ja vain pieni osa suuntautuu sen ulkopuolelle. Tämä on havaittu jo ennen työryhmäverkkojen aikakautta, mutta pätee edelleen.

Olli Nurminen

Avointa hajautusta

Tiimin tärkein perusidea on sen avoimuus, mikä takaa käyttäjille runsaan laite- ja ohjelmistovalikoiman. Eri tarpeisiin voidaan hankkia sopivimmat työasemat ja oheislaitteet verkkopalveluiden siitä kärsimättä. Avoimuuden toteuttamiseksi Tiimistä on tehty myös vahvasti standardoitu järjestelmä.

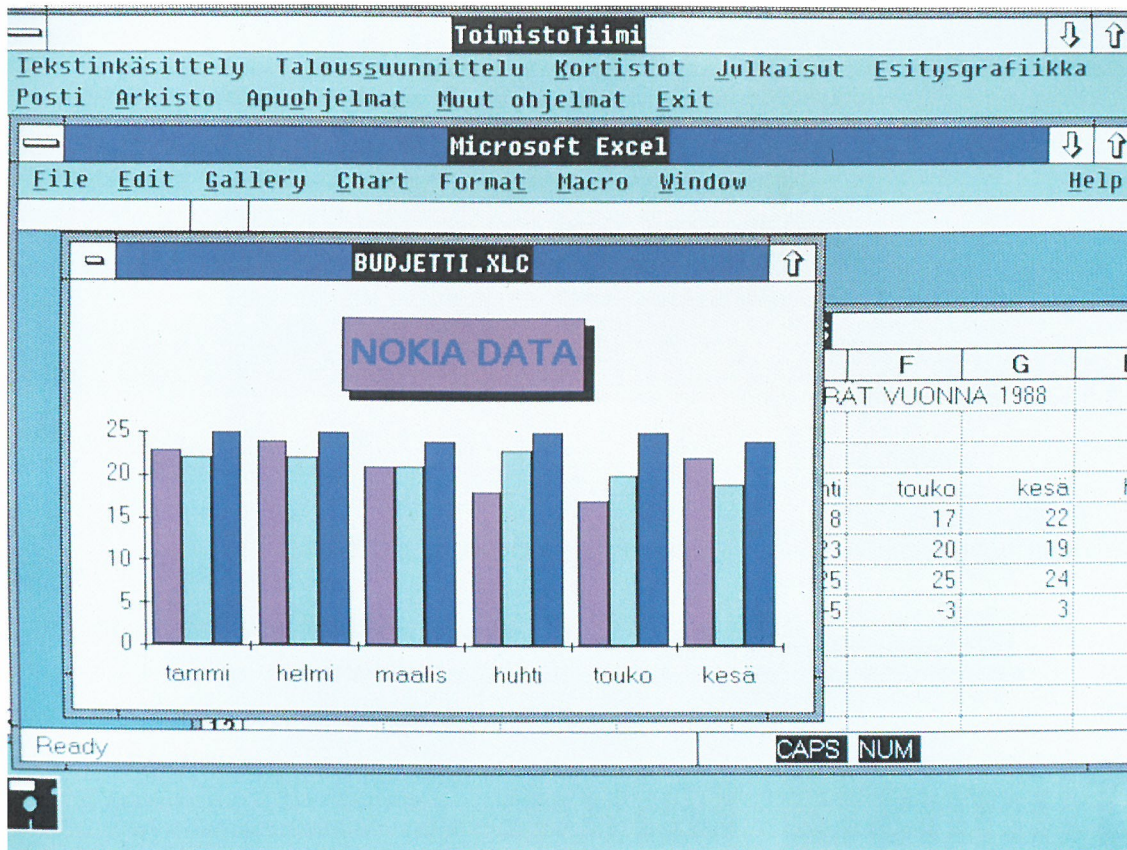
Rakenteeltaan Tiimi perustuu työasema-palvelin -ajatteluun. Tietojen käsittely on hajautettu verkkoon, jolloin varsinainen prosessointiteho on saatu lähelle käyttäjää. Näin yhden käyttäjän ajama raskas sovellus ei pääse häiritsemään muita käyttäjiä. Vastaavasti järjestelmää on kapasiteetin loppuessa helppo laajentaa uusien työasemien, palvelimien tai oheislaitteiden avulla.

Avoimesta ja hajautetusta Tiimistä saadaankin helposti räätälöityä vaihtoehtoisia verkkoympäristöjä erilaisine palveluineen ja tietoliikenneyhteyksineen aina tilanteen mukaan.

Tulevaisuudennäkymiä

ToimistoTiimi on keskeinen osa Nokia Datan tuotevalikoimaa. Sen kehitystä jatketaan jatkuvasti tekniikan kehittymisen myötä eli työryhmäverkon hankkija saa ajan tasalla olevan järjestelmän, joka myös pysyy ajan tasalla.

Kirsti Lindfors



Ulf Ekholm

Nokian ensimmäinen MS OS/2 -sovellus

Kaikissa organisaatioissa suurin osa työntekijöistä tuottaa ja viestii tietoa eri muodoissa. Toimistojärjestelmä palvelee kaikkien käyttäjiensä tietotarpeita työtehtävän luonteesta riippumatta. Toimistojärjestelmä on täten käyttäjiensä yhteinen nimittäjä – yhteinen työkalu. Käyttäjien ja heidän tehtäviensä erilaisuus asettaa suuret vaatimukset myös toimistojärjestelmälle.

Parhaimmillaan toimistojärjestelmä parantaa esimerkiksi projektiryhmän tai johtaja-sihteeriparin tehokkuutta vähentämällä tuotettavien luonnosten määrää, poistamalla päällekkäistä työtä, nopeuttamalla kommunikointia ja tiedon siirtyvyyttä sekä parantamalla tuotetun informaation tasoa niin asiasisällön kuin ulkoasunkin puolesta.

Työryhmäverkko soveltuu erinomaisesti toimiston työympäristöksi. Käyttäjä pääsee hyödyntämään henkilökohtaisia työvälineitään omalla työasemallaan – ja juuri niitä nimenomaisia työkaluja, joita hän itse tarvitsee. Hänen suorittamansa toiminnot eivät häiritse muiden käyttäjien toimintaa eivätkä muut pääse häiritsemään hänen toimintaansa riippumatta siitä kuinka suurta prosessointitehoa vaativia tehtäviä hän suorittaa – onhan hänellä oma keskusyksikkö käytössään. Kuitenkin hän voi niin halutessaan kommunikoida muiden käyttäjien kanssa ja käyttää työryhmäverkon tarjoamia yhteisiä palveluja.

Käyttöliittymä

Tietotekniikkaa taitamattoman käyttäjän kannalta ei ole oleellista millaisia komponentteja, millaisen käyttöjärjestelmän tai verkohallintajärjestelmän hänen työasemansa sisältää. Käyttäjä tarvitsee työasemaltaan palveluja: tekstinkäsittelyä, taulukkolaskentaa, tietokantoja, ...

Käyttöliittymän tehtävänä on piilottaa tekniikka käyttäjältä. Käyttäjän ei tarvitse käydä käyttöjärjestelmän tasolla, hän vain pyytää järjestelmältä palveluja (asiakirjan arkistointi, viestin lähetyt jne.). Käyttäjän ei tarvitse myöskään kahlata valikkorakennetta edestakaisin siirtyäkseen toiminnosta toiseen vaan hän pääsee siirtymään esimerkiksi postituksesta suoraan tekstinkäsittelyyn. Käyttöliittymä sisältää myös joukon opastetoimintoja, jotka osaltaan alentavat käyttöön-otto- ja oppimiskynnystä.

Windows tai Perfect

Työasemalla suoritettaviin toimintoihin – Tekstinkäsittelyyn,

taulukkolaskentaan, tietokantojen ylläpitoon, esitysgrafiikkaan ja viimeistelyjen julkaisujen tuottamiseen, Nokia on muodostanut kaksi tuoteperhettä.

Windows-perhe on käyttöliittymältään graafinen. Se hyödyntää hiirtä ja ikkunoitinta käyttäjän ja sovelluksen välisessä vuorovaikutuksessa. Perfect-perhettä hallitaan puolestaan perinteiseen tapaan näppäimistöltä. Kummankin tuoteperheen ohjelmistot on valittu siten, että ne ovat toiminnoiltaan monipuolisia ja kattavia. Lisäksi Finlannissa on otettu huomioon ohjelmien keskinäisen tiedonsiirron helppous.

Tapio Sulkava kertoi tuoteperheistä enemmän Softari 4/88:n sivuilla 4 ja 5.

Tiedot talteen

ToimistoTiimin palvelimet antavat käyttäjälleen monipuoliset arkisto- ja postipalvelut. Käyttäjä voi arkistoida ja postittaa kaikkea tietoa, jota hän pystyy työasemallaan tuottamaan. Arkisto ja posti eivät ota kantaa tiedon formaattiin. On kuitenkin muistettava, että lukeakseen

postin välityksellä saamansa tai arkistosta noutamansa tiedoston, vastaanottajalla täytyy olla tiedostoa vastaava työväline käytössään.

Arkistopalvelut antavat käyttäjälle mahdollisuuden tallentaa, hakea, etsiä ja poistaa asiakirjoja työryhmäverkon arkistopalvelimelta. Arkisto koostuu käytännössä kansioista, jotka sisältävät puolestaan asiakirjoja. Kuhunkin asiakirjaan liittyy asiakirjan määrittelysivu, joka määrittelee asiakirjaan liittyvät taustatiedot kuten asiakirjan nimen, laatijat, kuvauksen, päiväyksen, avainsanat ja saantioikeudet (yksityinen, yleinen).

Asiakirjoja voidaan hakea arkistosta määrittelysivun tietojen perusteella. Haun tuloksena saadaan lista haun kriteerejä vastaavista asiakirjoista. Lista voi sisältää joko pelkät asiakirjojen nimet tai sekä nimet että valikoituja määrittelysivun tietoja. Käyttäjä ottaa tarvitsemansa asiakirjan ulos arkistosta ja muokkaa sitä omalla työasemallaan. Lopuksi hän palauttaa sen takaisin arkistoon, josta joku toinen käyttäjä voi vuorollaan hakea sen jatkokäsittelyyn...

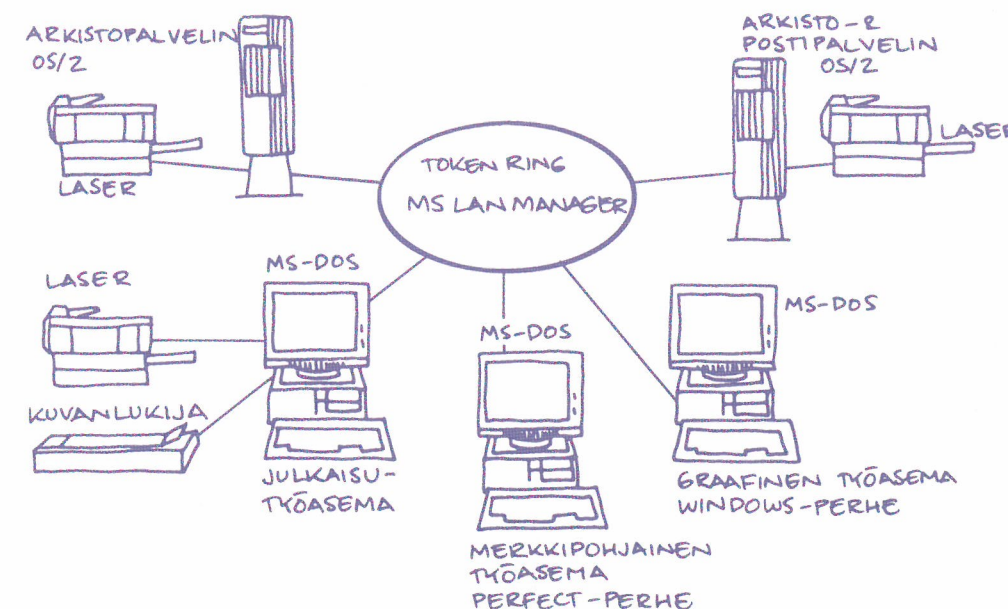
Posti kulkee

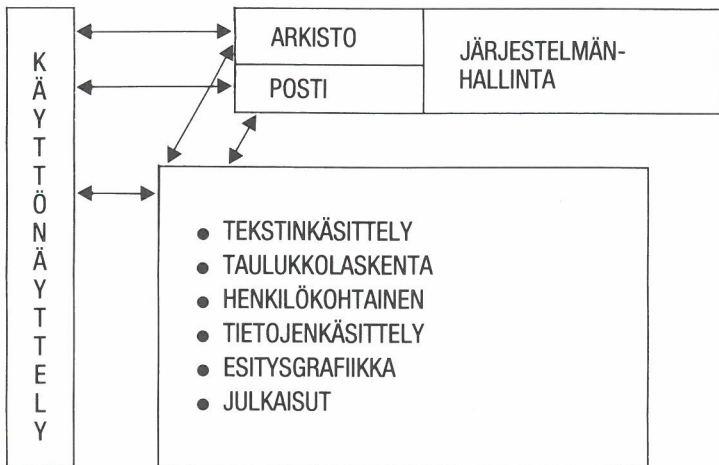
ToimistoTiimin postin avulla käyttäjä voi Tiimi-työryhmäverkon työasemalta lähettää ja vastaanottaa postia työryhmänsä muille jäsenille. Postina voidaan lähettää joko suoraan näppäimistöltä kirjoitettavia yksinkertaisia viestejä tai massamuistissa sijaitsevia tiedostoja.

Käyttäjä saa ilmoituksen itselleen tulleesta postista kirjoittautuessaan sisään Tiimiin sekä työskennellessään postinhallintaohjelmalla tai tuoteperheiden työvälineohjelmistoilla. Postitettavalle viestille tai tiedostolle voidaan määritellä myös liitteitä, jotka liikkuvat varsinaisen postin mukana.

Saapuneesta postista käyttäjälle tulostuu aihe, lähettäjä ja tieto siitä, onko se jo luettu vai ei. Saapuneen postin voi lukea, tallentaa käyttäjän hakemistoon, jatkolähettää, poistaa, tulostaa ja siihen voi tietysti myös vastata.

Sama työryhmäverkon palvelin voi toimia yhtäaikaaisesti arkiston ja postitoimiston lisäksi myös tavallisena tiedosto- ja tulostuspalvelimena.





Standardit ojennukseen

ToimistoTiimi on Tiimi-työryhmäverkon toimistojärjestelmä. Se käyttää MS OS/2-palvelimia ja MS-DOS työasemia. ToimistoTiimin käyttöliittymä noudattaa SAA CUA (Common User Access) ohjeistoa. Postipalvelut perustuvat X.400:aan eivätkä sen arkistotoiminnot ota kantaa asiakirjan formaattiin.

ToimistoTiimi toimii Token-Ring ja EtherNet-verkossa. Tiimi mahdollistaa tietoliikennetydet myös muihin järjestelmiin, kuten esimerkiksi IBM-ympäristöön.

Tiimin käyttöliittymän alle voidaan lisätä myös käyttäjäkohtaisia sovelluksia, jotka voivat olla sekä valmiita ohjelmistopaketteja että käyttäjän oman organisaation tuotteita.

Nokian palvelut

Palvellakseen mahdollisimman hyvin vaihtelevaa käyttäjäkuntaansa Nokia Data on perustanut kaksi uutta palveluyksikköä.

Käyttötuki neuvoa käyttäjiä puhelimitse kaikissa mikrotietokoneisiin ja Nokian tukemiin ohjelmistoihin liittyvissä kysymyksissä.

Kilon Tietokonehtaalle perustettu asennuspalvelu lähtee käyntiin porrastetusti ja on täydessä käytössä kesäkuuhun mennessä. Tällöin käyttäjät voivat tilata työasemansa täysin valmiiksi asennettuina käyttöjärjestelmineen, työryhmäverkkoon liittyvine laitteineen ja ohjelmistoina, tietoliikennelaitteineen ja -ohjelmistoina sekä henkilökohtaisine työvälineohjelmistoina.

Jukka Aspelund

Nokia MS Lan Manager Tieto kulkee verkossa

Pelkät kaapelit työasemasta toiseen eivät tee verkkoa. Toimiva järjestelmä tarvitsee tuekseen monipuolisen ohjelmistopakettin, joka huolehtii niin työasemien välisestä kommunikoinnista, yhteisten resurssien hallinnasta kuin järjestelmän tietoturvastakin.

Nokia MS Lan Manager on Microsoft MS OS/2 Lan Manageriin perustuva moniajoa tukeva avoin työryhmäverkkohjelmisto Token Ring ja EtherNet -verkkoihin. Se huolehtii Tiimin työasemien verkkopalveluista ja on yhteensopiva sekä MS-Netin että muiden MS Lan Manager -pohjaisten järjestelmien kanssa.

Kokoonpanovaihtoehdot

Nokia MS Lan Manager toimitetaan palvelinkohtaisena pakettina, joka sisältää sekä erityisen palvelinversion MS OS/2 1.0 -käyttöjärjestelmästä, palvelimen ohjelmiston että työasemien verkko-ohjelmistot. MS-DOS- ja MS OS/2 -työasemat käyttävät erilaisia ohjelmistoja, ja MS-DOS-versiosta käytetään

myös nimeä MS-DOS Lan Manager. Työasemien käyttöjärjestelmiä paketti ei sisällä, sillä niihin ei tarvita muutoksia.

Pieniä, korkeintaan yhden palvelimen ja viiden työaseman verkkoja varten ohjelmistosta on olemassa huokeampi versio (Entry Level Server). Verkon rajallista kokoa lukuun ottamatta se tarjoaa samat verkkopalvelut kuin täydellinen versio (Enhanced Level).

Palveluita käyttäjälle

Mikä tahansa palvelimen tiedosto, tiedostohakemisto tai levy-partitio voidaan määrittellä koko verkon omaisuudeksi antamalla sille verkkokäyttöä varten 'net share' -komentolla verkkoon näkyvä nimi. Työasemat voivat ottaa tällaisen julkisen hakemiston käyttöön antamalla 'net use' -komentossa sekä palvelimen että kyseisen hakemiston nimen.

Palvelimeen kytkeytyt kirjoittimet voidaan määrittellä julkisiksi verkkokirjoittimiksi luomalla palvelimeen yksi tai useampia kirjoitinjonoja, joissa tulostuspyynnöt odottavat kirjoittimen vapautumista. Samaan kirjoittimeen voi liittyä useita jonoja, joita hallitaan joko automaattisesti jonoprioriteettien avulla tai pysäyttämällä ja käynnistämällä manuaalisesti. On myös mahdollista liittää samaan jonoon useita kirjoittimia, jolloin tulostus ohjautuu ensiksi vapautuvalle.

Palvelimeen liitetyt modeemit tai muut linjayhteydet voidaan myös määrittellä julkisiksi, jolloin niiden käsittely onnistuu työasemilta aivan yhtä helposti kuin työasemien omiin sarjaportteihin kytkettyjen linjojen käyttö.

Rajapintoja sovelluksille

Lan Manager tarjoaa tietoliikennepalvelujen lisäksi myös rajapinnat sovellusten väliseen tiedonsiirtoon. Käytettävissä on sekä yhteyspohjainen että sähkötyyppinen mekanismi.

Lisäksi kaikki palvelimen ja työaseman ohjaustoiminnot ovat kutsuttavissa myös sovelluksista käsin. Tämä mahdollistaa käytännössä erityistehtäviin rakennettavien työryhmäverkkosten

ohjelmoimisen suoraan Lan Managerin päälle.

Komennot vs. ikkunat

Ohjelmiston eli käytännössä verkon hallinta onnistuu kahden vaihtoehdoisen käyttöliittymän avulla.

Komentoliittymä on itse asiassa laajennettu MS-Netin käyttöliittymä. Se pohjautuu erityiseen komentokieleeseen, jonka avulla voi suunnitella verkkohallintaan soveltuvia komentotiedostoja.

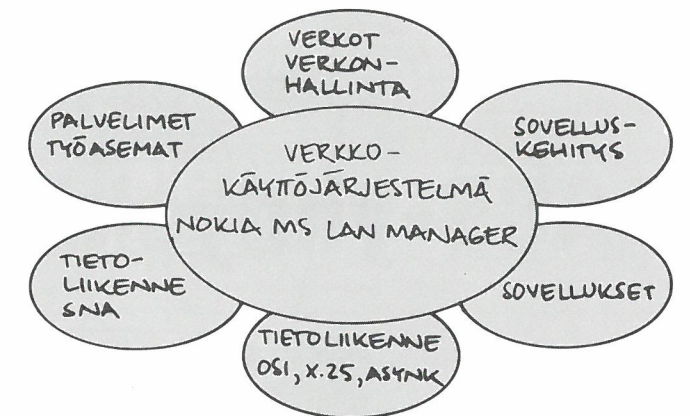
Ikkunoiva liittymä on puolestaan helppokäyttöisempi ja havainnollisempi kuin komentoliittymä. Käyttäjän ei tarvitse muistaa komentoja ulkoa, sillä pelkkä valikoista tunnistaminen riittää. Järjestelmä näyttää luettelon verkon palvelimista, joista kunkin julkisista palveluista saadaan tarvittaessa oma luettelon. Käyttöön otettavat palvelut valitaan yksinkertaisesti valikosta osoittamalla.

Salaus ja suojaus

Kenties mielenkiintoisin Lan Managerin yksityiskohta on mahdollisuus käyttäjäkohtaiseen tiedon salaukseen ja suojaukseen. Jotta työaseman käyttäjä voisi hyödyntää palvelimen palveluita, hänen tulee antaa järjestelmälle henkilökohtainen käyttäjänimi ja salasana. Lisäksi kuhunkin resurssiin liittyy luettelo sallituista käyttäjistä ja operaatioista.

Suojausaste on koko Lan Managerin tavoin laajasti muunneltavissa. Esimerkiksi ympäristössä, jossa suojauksia ei heti haluta ottaa käyttöön, ne voidaan kytkeä kokonaan pois. Vastavasti voidaan turvautua pelkästään MS-Net-tyyppisiin palvelukohtaisiin salasanoihin.

Nokia MS Lan Manager sisältää yksinkertaisen mekanismin lyhyiden viestien lähettämiseen verkon sisällä. Järjestelmä käyttää myös itse tätä mahdollisuutta tiedottaessaan järjestelmän kannalta vaarallisista virhetilanteista järjestelmän ylläpitäjille. Tällaisia tilanteita syntyy esimerkiksi kun palvelimen levytila on vähissä tai käyttöoikeuksia yritetään toistuvasti rikkoa – karjapaimenet kuriin!



Verkko hallussa

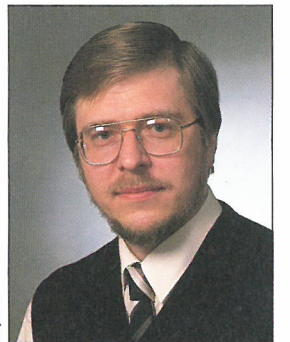
Verkkoa voidaan hallita joko suoraan palvelimesta tai mistä tahansa verkkoon liitetystä työasemasta käsin. Kaikkien toimintovaihtoehtojen saavuttamiseksi työaseman täytyy olla kuitenkin MS OS/2 -tyyppinen.

Hallintaoikeudet omaava käyttäjä voi lisätä ja poistaa julkisia hakemistoja, kirjoitinjonoja ja sarjaliitännäkanavia, päivittää käyttöoikeuksia, valvoa ja muuttaa kirjoitinjonoja sekä tutkia verkkopalvelimen keräämiä lokitiedostoja.

Lan Manager voidaan määrittellä myös käynnistämään jokin komentotiedosto automaattisesti aina tietyssä ajankohtana. Tätä toimintoa voidaan hyödyntää esimerkiksi varmuuskopioinnissa, sähköpostin siivouksessa ynnä muussa systeemin ylläpidossa.

Järjestelmä kerää tarpeen eli määrittelyjen niin vaatiessa tapahtumalokiin tietoja järjestelmän suorittamista toiminnoista ja näiden toimintojen laukaistavista virhetilanteista. Tallennettavat tiedot voidaan määrittellä aina tilanteen mukaan. Niitä voivat olla palvelimen käynnistys, käyttäjän liittyminen tai liityntäyritys, resurssin käyttöönotto tai käyttö...

Kaiken kaikkiaan Nokia MS Lan Manageria käyttävästä työryhmäverkosta on mahdollista rakentaa toimiva ja turvallinen järjestelmä, jota voidaan kehittää muuttuvien tarpeiden mukana.



Jukka Aspelund toimii Nokia Datan MS OS/2 ja Lan Manager -tuoteistusprojektien projekti-päällikkönä.



Risto Laine

Kirsti Lindforsin toimipaikka on Nokia Datan Tuotemarkkinointi, ja hän vastaa tuotepäällikön ominaisuudessa Tiimi-työryhmäverkkosten ja Unix/Xenix-ympäristön toimisto-ohjelmistojen markkinoinnista.

Tiimin yhteys ulkomaailmaan

AACS-yhdyskäytäväohjelmisto

Asynkroninen kommunikointi verkon työasemien ja ympäröivän maailman välillä on usein ollut eräs verkko-ohjelmistojen kompastuskivistä. Nokia Datan ratkaisu tähän ongelmaan on AACS-ohjelmistopaketti, joka toteuttaa asynkronisen yhdyskäytävän Tiimi-työryhmäverkon ja eri isäntäkoneiden välille.

Verkko ja sen työasemat

Ohjelmisto toimii NetBIOS-yhteensopivissa lähiverkoissa. Tällaisia ovat esimerkiksi Tiimi TokenRing, Tiimi EtherNet ja Tiimi NetNet.

Työasemana voidaan käyttää MikroMikko 3TT:tä sekä muita AT-yhteensopivia mikroja. Käyttöjärjestelmäversion täytyy olla joko MS-DOS 3.10 tai uudempi. Myös työaseman tulee luonnollisesti sisältää tarvittava lähiverkkosovitin sekä vastaava lähiverkko-ohjelmisto. Vapaata työmuistia tarvitaan vähintään 256 kilotavua.

Työaseman pääte-emulaattorin tulee kyetä yhteistyöhön AACS:n kanssa. Sopivia vaihtoehtoja ovat esimerkiksi Nokian WS220 ja WS7800. WS220 emuloi DEC:n VT220-päätettä (VT200-, VT100- ja VT52-tilat) sekä Nokian VDU202-päätettä. WS220 sopii käytettäväksi useimpiin tietoliikennetyhteyksiin kuten erilaisiin tietopankkeihin, sähköposteihin ja erityisesti VAX-laitteisiin. WS7800 puolestaan emuloi Honeywellin VIP7813-päätettä, jota käytetään DPS6- ja DPS8-yhteyksiin. AACS tarjoaa keskitetysti

asynkronisia yhteyksiä Tiimi-lähi-verkon työasemien ja eri isäntäkoneiden välille. Yhteyksiä voidaan ylläpitää joko kiinteiden kytkentöjen tai erilaisten modeemien avulla. Yhtä yhdyskäytävää kohti käytettävissä on yhtä aikaa enintään 12 linjaa (8 + yhdyskäytävän sarjaportit). Mikäli tämä ei riitä, samaan verkkoon voidaan kytkeä useampia yhdyskäytäviä.

Yhdyskäytävän vaatimukset

Yhdyskäytäväkoneeksi tarvitaan mahdollisimman tehokas MikroMikko. Koneen tulee luonnollisesti sisältää lähiverkkosovitin ja vastaava lähiverkko-ohjelmisto aina NetBIOS-liityntään asti. Näyttöä ja näppäimistöä ei välttämättä tarvita lainkaan. Työmuistin vähimmäismäärä on 512 kilotavua. Yhdyskäytävän toimiminen tarkoittaa käytännössä myös sitä, että laitetta ei voida samanaikaisesti käyttää mihinkään muuhun tehtävään.

Kun yhdyskäytäväkoneeseen lisätään AAEB-kortti, saadaan normaaliin sarjaporttien lisäksi käyttöön 8 ylimääräistä sarjaporttia. AAEB on lyhyt kortti, johon on kaapelilla liitettyä

kahdeksan sarjaportin liitäntärasia. Kortin liitännät ovat RS232C, 25-pin male ja D-Sub.

Linjojen nimet

Käytön helpottamiseksi voidaan yhdyskäytävän eri linjoille määritellä omat kohdenimet. Jos esimerkiksi useasta linjasta on yhteys samaan isäntäkoneeseen, kullekin linjalle voidaan määritellä oma kohdenimi tai useita linjoja voidaan määritellä vastaamaan yhtä kohdenimeä. Jos kohdenimi osoittaa modeemilla tai korttmodeemilla varustettuun linjaan, sille voidaan määritellä myös puhelinnumero ja modeemin tyyppi.

Avatessaan Tiimin työasemalta asynkronisen yhteyden käyttäjä valitsee emulaattorilla haluamansa kohdenimen. AACS ottaa tämän jälkeen yhteyden kyseiseen kohdenimeen – esimerkiksi automaattisesti modeemin määrittelemään puhelinnumeroon. Työaseman kautta tapahtuva yhteydenotto voidaan automatisoida emulaattoreiden komentokielen avulla. Jos porttia ei ole vapaana, käyttäjä voi pyytää AACS:ää laittamaan pyynnön jonoon.

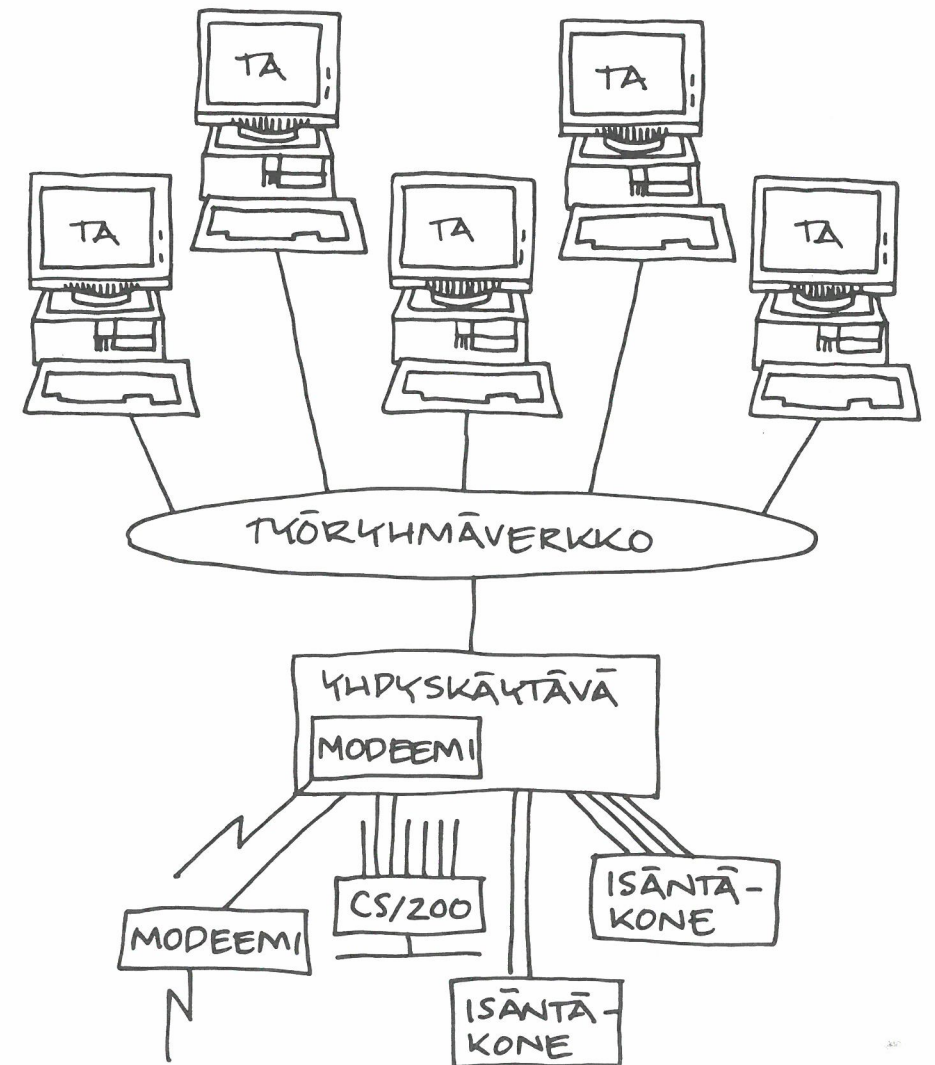
Tieto kulkee

Normaalin pääte-emuloinnin lisäksi AACS ja emulaattori -yhdystelmän avulla voidaan työaseman ja isäntäkoneen välillä siirtää myös sekä teksti- että binaaritiedostoja. Tiedonsiirto tapahtuu joko Kermit- tai PCXFER-protokollalla, mutta mikäli käytössä on verkko-ohjelmisto, voidaan siirto suorittaa myös verkon tiedostopalvelimen ja isäntäkoneen välillä.

Järjestelmä tukee oheiskirjoittimen käytössä sekä hardcopya että isäntäkoneelta tulevan kirjoitintulostuksen ohjausta työaseman kirjoittimelle. Verkko-ohjelmiston avulla kirjoitintulostukset voidaan ohjata myös kirjoitinpalvelimelle.

Tehokas yhdyskäytävä

Yhdyskäytäväksi kannattaa valita mahdollisimman tehokas tietokone -ainakin mikäli siirrettävät tietomäärät ovat suuria, ja halutaan käyttää useita yhteyksiä yhtä aikaa. Oheisessa taulukossa on ohjeelliset arvot linjamääristä ja vastaavista nopeuksista, joilla yhteys toimii ilman että merkkejä häviää kovalla kuormituksella (esimerkiksi kaikilla linjoilla tiedostonsiirto isäntäkoneilta työasemille). ■



Yhdyskäytävä	9600 bps	4800 bps	≤2400 bps
286 8 MHz (esim MM 3TT)	3 linjaa	8 linjaa	>8 linjaa
286 12 MHz (esim MM 3TT m215)	6 linjaa	>8 linjaa	>8 linjaa
386 (esim MM 3TT/386 ilman näytönohjainkorttia tai MikroMikko 3/386)	>8 linjaa	>8 linjaa	>8 linjaa

Ominaisuudet

- ★ Linjanopeus max 9600 bps
- ★ samanaikaisia yhteyksiä jopa 12
- ★ Määrittelymahdollisuuksia
 - kohteiden nimet
 - tietoliikenneparametrit
 - emulaattori valittavissa
- ★ Automaattinen yhteydenotto
- ★ Jonotusmahdollisuus
- ★ Useita portteja yhdelle kohdenimelle
- ★ NetBIOS-standardi

Kun Partek ryhtyi ennen 80-luvun puoliväliä suunnittelemaan uutta toimitaloa Helsingin Sörnäisiin, päätettiin yhtiön tietopalvelut panna samalla kuntoon. Uuden konttorin kun ei haluttu joutuvan oitis johtoremonttiin, vaan uusi työryhmäverkko rakennettiin yhtä jalkaa itse talon kanssa.



Verkolla kaikki palvelut kaikille

Partek vastaa ajan haasteisiin

Sörnäisten verkon ohessa luotiin tietoliikenneyhteydet Partekin eri puolilla maata sijaitseviin yksiköihin, ja tuloksena oli koko Suomen kattava verkko, jolla on yhteyksiä myös maan rajojen ulkopuolelle.

Näin laajoja tietojärjestelmiä ei ole Pohjoismaissa montaa, ja niin verkon omistaja Partek kuin ylläpitäjä Parcompkin ovat verkostaan syystä ylpeitä.

Parcomphan on syntyjään Partekin atk-osasto, joka on eriytetty omaksi yritykseksi, josta Partek omistaa puolet. Partekin tietohallintojohtaja Kalevi Hanttu kuvaa järjestelyä 'piiskan vaihtamisella porkkanaan'.

Keskityksestä hajautukseen

Partek on käynyt vuosien saatossa lävitse normaalin kehityskulun yksittäisestä keskuskoneesta laajaan verkkojärjestelmään.

Pentti Koponen Parcompista kertoo keskitetyn järjestelmän ongelmista:

– Keskitetty tietojenkäsittely oli hyvin haavoittuvaista, Koponen kuvailee. Yksikin epäonnistunut ajo saattoi romuttaa koko aika-aulun, ja kaikki piti aloittaa alusta uudestaan. Samaten järjestelmän muuttaminen ja kehittäminen oli hyvin hankalaa.

Ensimmäinen hajautusvaihe toteutettiin jo taloussovelluksiin käytetyn keskuskoneen sisällä, kun kullekin liiketoiminta-alalle luotiin oma kirjanpitojärjestelmä. Kutakin osaa voitiin kehittää näin erikseen.

Ajan oloon kullekin yksikölle hankittiin omat operatiiviset järjestelmät. Yksi käyttäjä saattoi kuitenkin tarvita usean järjestelmän palveluita, ja kun kutakin isäntäkoneita varten tarvittiin päätteeseen omat johtimet, oli ohjelmassa ongelmia. Parhaassa tapauksessa mikron takapaneelista saattoi löytyä jopa viidet liitäntäjohtot.

Tarvittiin siis verkko, jonka jokaiselta työasemalta on mahdollista saada yhteys jokaiseen verkkoon kytkettyyn isäntäkoneeseen, palvelimeen tai työasemaan laitemerkistä riippumatta. Kun verkkojärjestelmistä oli myös kokemusta toimivan sihteeriverkon verran, toimeen voitiin ryhtyä heti kun aika oli kypsä.

Avoimen Ethernet-tyyppisen verkon toimittajaksi valittiin Nokia. Asennuksen jälkeen verkkoon on kytketty eri merkkisiä ja tyyppisiä työasemia, päätteitä, isäntäkoneita, oheislaitteita ja muita palveluita.

Vakiopalveluja kaikille

Partekin verkkoratkaisu on tuonut mukanaan sen, että istui käyttäjä sitten Sörnäisissä tai Paraisilla, verkon kaikki palvelut ovat välittömästi saatavilla.

Tietopalveluille on tärkeintä niiden toimivuus. Mitään statusverkkoa ei ole haluttu kilpailijoiden kiusaksi rakentaa, vaan palveluja laajennetaan vain sitä mukaa kun vanhat saadaan toimimaan luotettavasti. Tekniikasta ei ole haluttu tehdä itse tarkoitusta.

Järjestelmä koostuu standardeista ja vakiopalveluista, jotka kloonataan kaikille käyttäjille. Myös työasemien, työvälineohjelmien ja jopa työasemien hakemistopuiden standardointiin on pantu Partekin sisällä paljon painoa, sillä mikä tahansa ohjelmisto ei niin vain pyöri työasemassa, jonka muistista melkoinen osa kuluu pelkkään verkkohallintaan. Hyviä esimerkkejä standardoiduista työvälineistä ovat WordPerfect ja Lotus.

Verkossa elävä ihminen

Kalevi Hanttu jakaa verkkojärjestelmän rakentamisen kolmeen osaan:

– Ensin täytyy saada itse verkko toimimaan, sitten on saatava verkon palvelut toimimaan ja lopuksi on saatava ihmiset toimimaan verkon mahdollistamalla tavalla.

Ihmiset käyttävät edelleen verkkopalveluja vain itselleen, eivät verkon osina. Verkko on kuitenkin jo kypsytty näkemään tavallisena työkaluna, eikäukaan vielä tiedä, millaiseksi työ

muuttuu, kun ihmiset todella oppivat elämään verkossaan.

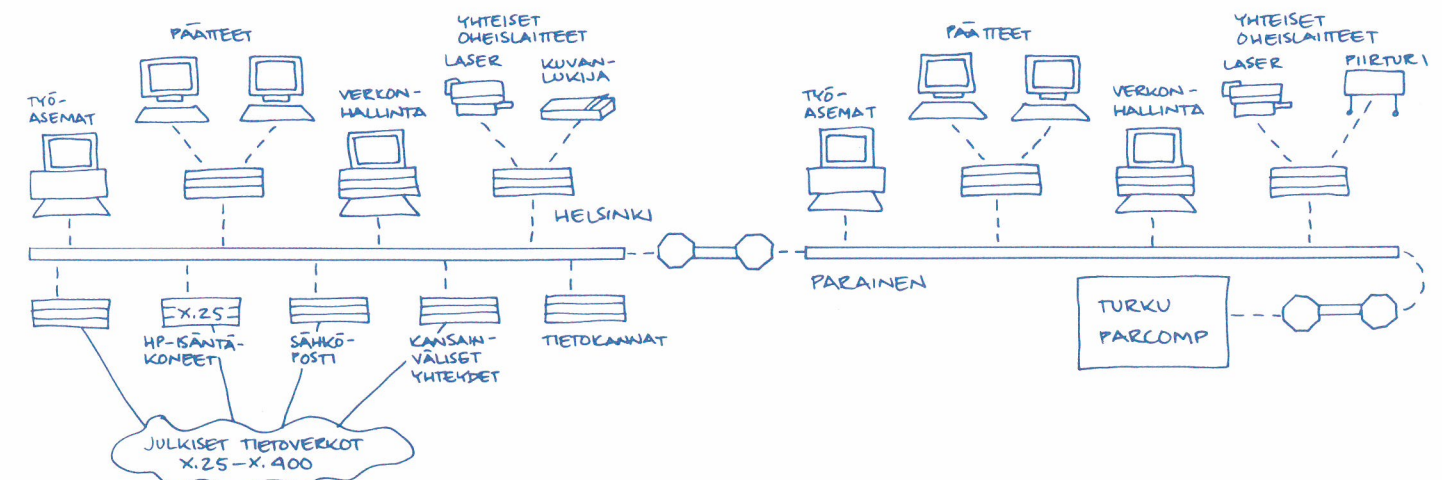
Jo nyt verkossa työskentely on nopeuttanut monia rutiineja. Esimerkiksi laajojen budjettien kokoaminen onnistuu nykyään päivissä kuukausien asemesta. Samaten tiedon siirto eri toimipisteiden välillä vie tätä nykyä minutteja perinteisen postin kuluttamien päivien sijasta.

Tiedot turvaan

Mahdollisuus päästä miltä työasemalta tahansa käsiksi mihin palveluun tahansa asettaa verkon tietoturvan koviille. Laajat salasana-vaaditukset uhkaavat muodostua läpikäymättömiksi myös oikeille käyttäjille siinä missä yksikin suojaamatta jäänyt yhteys jättää järjestelmän avoimeksi niin urkkijoiden kuin virustenkin mellestää.

Partekin verkkoratkaisusta on pyritty tekemään ulkoa katsoen mahdollisimman suljettu tietoliikennettä kuitenkin kuristamatta. Järjestelmän riskikohtia valvotaan ja alan kehitystä seurataan jatkuvasti. Tärkeintä on kuitenkin pitää huolta kulunvalvonnasta ja valistaa henkilöstöä mahdollisista riskeistä, tai kuten Partekin taloushallinnon puolella istuva Karl B. Gunnelius sanoo:

– Mitään tietojärjestelmää ei voida suojata täydellisesti, jos ihmiset eivät ole ajan tasalla, hän toteaa. Kaikki järjestelmän käyttäjät on opetettava kryptaamaan luottamukselliset tiedot ja viestit sekä myös vakuutettava tietoturvan välttämättömyydestä. ■



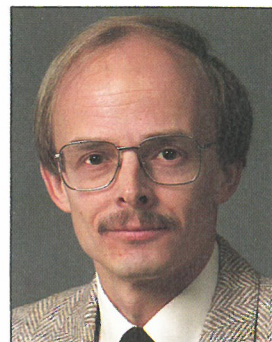
Learning Ada

Part 3

In Parts 1 and 2 we talked about the packages and overall structure of an Ada program. In this article we will concentrate on the internal structure of a single main program or subprogram.



Veikko Kehä toimii Nokia Datan Tuotekehityksessä menetelmäpäälikkänä.



Lassi Juutinen työskentelee Nokia Datan Tuotekehityksessä menetelmäasiantuntijana.

Below we will discuss the data types and basic control structures of Ada. Since Ada is a very rich programming language, we can here present only the most commonly used features that are needed for writing simple programs. For a deeper insight, we suggest you read one of the Ada text books referred to. Several computer-aided instruction courses are also available.

Data Types and Type Checking

A program unit is composed of a declaration part and a statement part. Every identifier (e.g. a variable name) used in a statement part must be declared in the declaration part unless it is a predefined identifier known to all Ada compilers.

Ada has a set of predefined data types, though the programmer usually defines his or her own data types basing on these types.

The predefined data types are given in Figure 2. These types are always available for a program, and they are used in variable and constant declarations, etc

```
ITEM_FOUND : BOOLEAN;
PAGE_LENGTH : constant INTEGER := 60;
```

Here we declare that the variable ITEM_FOUND is of the type BOOLEAN. A constant (e.g. PAGE_LENGTH) is initialized with the value given in the declaration, and it cannot be changed afterwards.

The type of each variable is fixed in variable declarations. It defines the operations that can be performed upon the variable. In Ada, the compiler checks very carefully that the variables are used according to their types. For example, as there are no implicit type conversions in Ada, the compiler gives an error message if an integer value is assigned to a variable of the type FLOAT. Conversion between these two types is possible, but it must be specified explicitly.

Data typing is stricter in Ada than in most other programming languages. A Pascal programmer may be surprised when compiling his or her first Ada program. The compiler finds type mismatches that the programmer had not thought of at all. The benefit of strong typing is that the data types cause no surprises during the program run time.

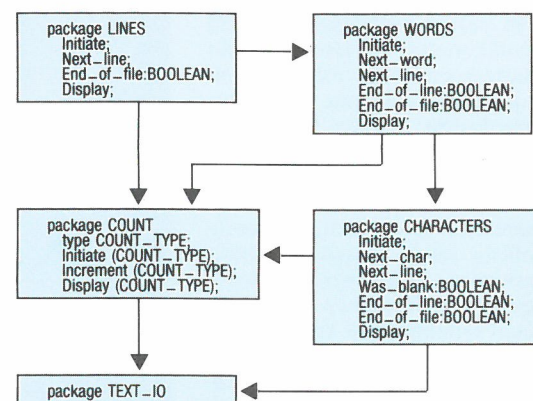


Figure 1: Suggested program structure

Type	Values
BOOLEAN	Literals FALSE and TRUE
INTEGER	Integer numbers
FLOAT	Floating point numbers
CHARACTER	ASCII characters (7-bit)
STRING	Character strings
DURATION	Fixed point values that represent time

Figure 2: Predefined data types

Operator Symbols	
Logical operators:	not, and, or, xor, and then, or else
Relational operators:	=, /=, <, <=, >, >=
Adding operators:	+, -, &
Multiplying operators:	*, /, mod, rem
Other operators:	in, **, abs
& is a catenation operator for the type STRING, in checks if a value belongs to a subtype.	

Numeric Types

The predefined type INTEGER defines integer numbers. The value range of this type is defined by the implementation. In most microcomputers an integer value is represented by 16 bits, but in main frames the length is usually at least 32 bits. Implementation can also support the types SHORT_INTEGER and LONG_INTEGER, whose ranges are likewise defined by the implementation.

This causes portability problems when a program is ported to other computers. To achieve better portability the programmer should define the integer types himself. For example:

```
type SCORE_TYPE is range -1_000_000..1_000_000;
```

The programmer defines the range by adding a **range constraint**, and the compiler selects the proper data type independently of the internal word length. Note: The type names and ranges should correspond to the real world.

In Ada, real numbers can be represented in two ways: floating point numbers represent real numbers with an accuracy of a given number of decimal digits, and fixed point numbers represent real numbers with absolute accuracy. The difference between two consecutive fixed point numbers is a constant, whereas the difference between two consecutive floating point numbers depends of the absolute value of the numbers.

Floating point variables can be declared using the predefined type FLOAT. However, as with the integer types, the range and accuracy may vary from one implementation to another. The suggested way to declare a floating point type is

```
type WEIGHT_TYPE is digits 7 range 0.0 .. 3000.0;
```

The programmer defines the accuracy – in this case seven digits – and the range of values. Again the type names and ranges should correspond to reality.

A fixed point type is declared by defining the difference between two consecutive numbers, called a **delta**, and the range of the values that belong to this type. A fixed-point type declaration could look like this:

```
type MONEY_TYPE is delta 0.01 range 0.0 .. 1_000_000.0;
```

In this way we can use MONEY_TYPE to store Finn marks and pennies to an accuracy of one penny.

Enumeration Types

When abstracting objects from the real world, we often need a type with a set of non-numeric values. This kind of a type is called an **enumeration type**. Examples of enumerations types are:

```
type MOTOR_STATE_TYPE is
  (OFF, FORWARD, BACK);
```

```
type CARD_SUIT_TYPE is (CLUBS, DIAMONDS,
  HEARTS, SPADES);
```

```
type MONTH_TYPE is (JAN, FEB, MAR, APR, MAY, JUN,
  JUL, AUG, SEP, OCT, NOV, DEC);
```

The predefined type BOOLEAN can be represented as an enumeration type:

```
type BOOLEAN is (FALSE, TRUE);
```

The predefined enumeration type CHARACTER contains all the ASCII-values. One problem with the type CHARACTER is that it has only 128 values, which are represented with seven bits. Many computers rely on eight-bit character codes. Scandinavian characters in particular need eight bits. This often causes portability problems.

Array Types

A data structure with many components of the same type is called an *array* in Ada. For example, one could write a declaration of a vector type, and declare a variable of that type:

```
type VECTOR_TYPE is
  array (INTEGER range 1..10) of FLOAT;
VECTOR : VECTOR_TYPE;
```

Here the type of the array components is FLOAT, and the index type is INTEGER and the index values are constrained to the range 1..10. When an array component is referred to, parentheses are used, as in VECTOR(5).

The predefined STRING is an array type whose index type is an integer type (values starting from 1), and components are of the type CHARACTER. The length of a character string is given when a variable is declared:

```
NAME : STRING (1..30) := "Ada Augusta Lovelace ";
```

Strings of different lengths are considered to be of separate types in Ada.

The components of an array can also be arrays:

```
type MATRIX_TYPE is array (1..10) of VECTOR_TYPE;
```

Multi-dimensional array types can also be defined as:

```
type CUBE_TYPE is array (1..5, 1..5, 1..5) of FLOAT;
```

Ada sets no limits on the maximum dimensionality of arrays. The type of array indexes can be any integer type or enumeration type:

```
type RESULT_TYPE is
  array (MONTH_TYPE) of SCORE_TYPE;
```

Record Types

Array types are used to implement objects that are collections of similar objects. But objects in the real world are often composed of dissimilar classes of objects. In these cases we can use record types to model such data. For example, we can declare a record type:

```
type DAY_TYPE is
  record
    DAY : INTEGER range 1..31;
    MONTH : MONTH_TYPE;
    YEAR : INTEGER range 1900..2000;
  end record;
A_DAY_IN_THE_LIFE : DAY_TYPE := (20, JAN, 1989);
```

The record variable is initialized here with an **aggregate**, which is simply a structured value, listing the values of all the record components. A record component is selected with a dot, as in the

following assignment statement:

```
A-DAY-IN-THE-LIFE.DAY := 21;
```

Access Types

In many programming languages there are pointer variables that contain not the data itself, but the address of the data. Pointer types are called **access types** in Ada. With access types we can implement structures like linear linked lists, trees, graphs, stacks, queues, sets, and other more complex real world objects.

Figure 3 shows type definitions for a message buffer. The number of messages varies during the execution of the program, so we cannot use a fixed length array. Instead we construct a linked list of elements containing the message, and define a pointer to the next message. An incomplete type definition for **BUFFER_TYPE** is required, because no identifier can be used before it is declared.

```
type MESSAGE_TYPE is STRING (1..50);
type BUFFER_TYPE;           -- Incomplete type definition
type BUFFER_POINTER_TYPE is access BUFFER_TYPE;
type BUFFER_TYPE is         -- Complete type definition.
  record
    MESSAGE : MESSAGE_TYPE;
    NEXT_MESSAGE : BUFFER_POINTER_TYPE;
  end record;
```

Figure 3: Example of an access type

Control Structures

Not only does Ada have an abundant set of constructs for describing data, but it also contains a collection of powerful statements for creating algorithms. Every program can be composed of one or more of the three basic control structures: sequence, selection, and iteration. In addition, Ada provides parallel processes that are implemented by **tasks**. We will return to tasks in later articles.

Sequence Structures

A sequential program flow is established by writing one statement after another separated by a semicolon. Assignment statements, block statements, and subprogram calls are the basic sequential statements.

A block of statements is a named sequence of statements surrounded by the reserved words **declare**, **begin**, and **end**. For example, if we want to swap the values of two integer variables A and B, we can write a block containing three assignment statements:

```
SWAP:
  declare
    TEMP : INTEGER;
```

```
begin
  TEMP := B;
  B := A;
  A := TEMP;
end SWAP;
```

Selection Structures

Selection is implemented in Ada with **if** and **case** statements. An example of an **if** statement is:

```
if BUFFER_SIZE > MAXIMUM_BUFFER_SIZE then
  PROCESS_PROGRAM_ERROR;
elsif BUFFER_SIZE = MAXIMUM_BUFFER_SIZE then
  PROCESS_BUFFER_OVERFLOW;
else
  ADD_NEW_MESSAGE;
end if;
```

In an **if** statement the decisions are made according to Boolean values. For selections depending on the value of some integer or enumeration type a **case** statement is used:

```
case BUFFER_SIZE is
  when MAXIMUM_BUFFER_SIZE / 2 =>
    SEND_OVERFLOW_WARNING;
  when MAXIMUM_BUFFER_SIZE =>
    PROCESS_OVERFLOW;
  when others => null;
end case;
```

The expressions in the case alternatives must be static - i.e. the compiler must be able to evaluate them. A null statement doesn't do anything, its purpose is for the compiler to check that the programmer hasn't forgotten to write a statement. Unfortunately we can't assign names for **if** or **case** statements in the same way as for blocks. In the case of nested statements such names could improve the readability of the program.

Iteration Structures

Three kinds of loop statements are used for iteration in Ada. Ada permits a basic loop, a **for** loop, and a **while** loop. We can apply an **exit** statement within the loop to end the iteration. An example of a basic loop is:

```
DISPLAY_MESSAGE:
  loop
    SYMBOL := READ_CHARACTER (MESSAGE_BUFFER);
    exit when END_OF_MESSAGE (MESSAGE_BUFFER);
    DISPLAY (SYMBOL);
  end loop DISPLAY_MESSAGE;
```

Here is an example of a **for** loop:

```
CALCULATE_YEARLY_SUM:
  for I in MONTH_TYPE loop
    SUM := SUM + MONTHLY_POINTS (I);
  end loop CALCULATE_YEARLY_SUM;
```

A **while** loop looks like this:

```
PROCESS_CUSTOMER_FILE:
  while DATA_AVAILABLE (CUSTOMER_FILE) loop
    INPUT_BUFFER := GET_NEXT (CUSTOMER_FILE);
    PROCESS_CUSTOMER (INPUT_BUFFER);
  end loop PROCESS_CUSTOMER_FILE;
```

Input and Output

Ada does not contain statements proper for input and output. Instead there are predefined packages for basic I/O.

The package **TEXT_IO** is used for textual I/O. We can write a string of characters on the screen:

```
TEXT_IO.PUT ("Hello world!");
```

Line feeds are controlled by calling either **PUT_LINE** or **NEW_LINE** procedures. The corresponding procedures for reading are **GET** and **GET_LINE**. If a file name is given as the first parameter in an I/O call, the input and output operations are applied to a file instead of standard channels, which usually means the keyboard and screen. Thus a character is read from the keyboard by the statement:

```
TEXT_IO.GET (SYMBOL);
```

The variable **SYMBOL** now contains the typed character. When we read a line, it is important to know how many characters have been read. The **GET_LINE** call returns the number of characters read in a parameter.

```
TEXT_IO.GET_LINE (LINE_BUFFER, LAST);
```

LAST indicates how many characters have been read this time.

For numeric I/O there are generic subpackages in **TEXT_IO**. The programmer must instantiate a package for each type. For example, integers are written as follows:

```
package INT_IO is
  new TEXT_IO.INTEGER_IO (INTEGER);
```

```
...
COUNT : INTEGER;
...
COUNT := 5;
...
TEXT_IO.NEW_LINE;
TEXT_IO.PUT ("The count is ");
INT_IO.PUT (COUNT, 2);           -- Width of field is 2
TEXT_IO.PUT (" ");
```

These statements output "The count is 5.". Note, that only one value can be written with one call of **PUT**.

The other predefined packages for I/O are **DIRECT_IO** and **SEQUENTIAL_IO** for handling values of any type.

Concluding Remarks

We have now presented the classes of data types and basic control structures available in Ada. Figure 4 is a simple example of a complete main program.

The advantage of Ada is that programs are very easy to read. The programmer can also make his programs maintainable. Of course the possibility of misusing a programming language always remains. But the strong typing in Ada enables the compiler to find many programming errors that would not be discovered in other programming languages until run time. After compilation, an Ada program has less errors than programs written in most other languages. ■

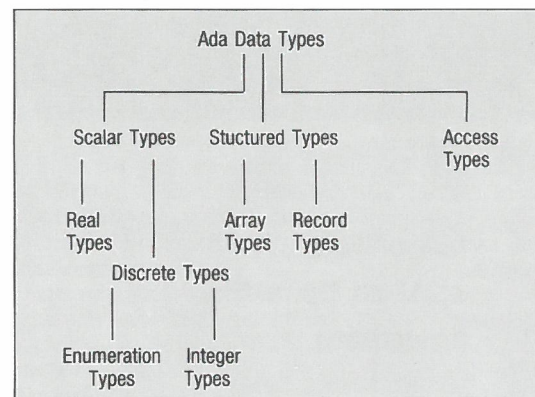
```
with TEXT_IO;

procedure CF is
  -- This program prints the frequency of all printable
  -- characters in a file given as the standard input file:
  -- CF < INPUT.TXT
  -- if the input file has not been created with Ada I/O,
  -- treating of end of file situation may cause problems
  -- that are not handled by this example.
  CHAR : CHARACTER;
  type COUNT_TYPE is range 0..10-000;
  type FREQUENCY_TYPE is
    array (CHARACTER) of COUNT_TYPE;
  FREQUENCY : FREQUENCY_TYPE := (others => 0);
  package COUNT_IO is
    new TEXT_IO.INTEGER_IO (COUNT_TYPE);

begin
  READ_FILE:
    while not TEXT_IO.END_OF_FILE loop
      TEXT_IO.GET (CHAR);
      FREQUENCY (CHAR) := FREQUENCY (CHAR) + 1;
    end loop READ_FILE;

    TEXT_IO.PUT_LINE ("CHARACTER  FREQUENCY");
  PRINT_TABLE:
    for C in CHARACTER loop
      if C in ' ' .. 'ü' and FREQUENCY (C) > 0 then
        TEXT_IO.PUT (C);
        COUNT_IO.PUT (FREQUENCY (C), 12);
        -- Output field width is 12 characters
        TEXT_IO.NEW_LINE;
      end if;
    end loop PRINT_TABLE;
end CF;
```

Figure 4: Sample program



Answer to the Exercise

The last number of Softari contained an exercise on designing a program that reads a text file and counts and displays the number of lines, words, and characters in the file.

Figure 1 shows one solution: each box represents an Ada package and its visible operations. All the functions return a Boolean value, and all the parameters are of the type **COUNT_TYPE** declared in the package **COUNT**. The arrows show how the packages use one another. Lines, words, and characters are counted in separate packages, and the use of the **TEXT_IO** package is hidden in the lower packages.

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 1983.
- /4/ Amoroso, S., Ingargiola G.: Ada. An Introduction to Program Design and Coding, Pitman, 1985.

MikroMikko 3TT m215

Pieni mutta hiljainen



MikroMikko 3TT m215 on MikroMikko-perheen uusin jäsen, jonka suunnittelu on pitkälti pohjautunut edellisten MikroMikko 3TT -mallien käyttäjien kokemuksiin.

Tekniikka pelaa

MikroMikko 3TT m215 on hiljainen, vähän pöytäteline viedä käyttäjäystävällinen mikrotietokone. Sitä voidaan käyttää joko henkilökohtaisena koneena tai Tiimi-tiimiverkon työasemana.

3TT m215 toimii niin MS-DOS- kuin MS OS/2-ympäristössäkin täysin yhteensopivasti. Vakiokäyttöjärjestelmänä toimitetaan MS-DOS 3.3.

LIM/EMS 4.0 -standardin mukaista perusmuistia saa suoraan emokortille 2 tai 4 megatavua eli normaalikäyttöön vähintäänkin riittävästi.

3TT m215 käyttää Intelin 80286 prosessoria, joka pyörii

12,5 MHz:n kelloaajuudella. Se on siis kolmanneksen nopeampi kuin edeltäjänsä. Jos tämä vauhti ei vielä riitä, lisää puhtia saadaan 10 MHz:in matematiikkaprosessorilla.

Hiirelle löytyy vakiona niin sarja-, rinnakkais- kuin PS/2-tyyppinenkin hiiriliitäntä. Lisäksi vapaana on vielä kaksi pitkää korttipaikkaa esimerkiksi verkkokorttia ja tietoliikennekorttia varten.

Käytetty levyke koko on 3,5 tuumaa ja tallennuskapasiteetti 1,44 megatavua. Levykeasemat lukevat myös tavallisia 720 kilotavun levykkeitä ilman ongelmia. Järjestelmän mukana toimitettavat kiintolevyasemat voivat olla kapasiteetiltaan joko 20 tai 40 megatavua.

30 tuntia viikossa

Keskusyksikkö on erittäin pieni ja mahtuu hyvin pienellekin työpöydälle. Se voidaan sijoittaa myös työpöydän alapuolelle erityisellä ripustustelineellä.

Näppäimistö on standardin mukainen Nokia 102 -näppäimistö. Liitosjohto on erityisen pitkä, jotta se yltäisi tarvittaessa myös pöydän alle.

MikroMikko 3TT m215:een vakiona toimitettavat näyttövaihtoehdot ovat Nokian 15 tuuman positiivinäyttö ja 14 tuuman VGA-värigrafiikkänäyttö. Molempien näyttöjen magneettijohdot ja elektrostaattinen säteily alittaa suositusraajat. Lisäksi suuret värinäyttömät näytöt säästävät niitä keskimäärin 30 tuntia

viikossa tuijottavia silmiä tulevaakin elämää varten.

Tiedot turvassa

Koneen massa- ja käyttömuistissa olevat tiedot on mahdollista suojata kolmella eri tavalla. Ensinnäkin käyttäjä voi määritellä tarpeen mukaan käyttäjä-, järjestelmä- tai näppäimistösalasanoja. Lisäksi näppäimistö voidaan lukita avainlukituksen lisäksi myös ajan kulumisen perusteella. Jos näppäimistöä ei esimerkiksi viiteen minuuttiin käytetä, se lukittuu. Lukituksen saa poistettua vain käyttäjän määrittelemällä salasanalla.

Suosikkiperheet

Käyttäjällä on tietenkin täysi vapaus valita markkinoilta löytyvistä standardiohjelmistoista itselleen parhaiten sopivat vaihtoehdot. Nokia Data tukee kuitenkin erityisesti Windows- ja Perfect-tuoteperheisiin kuuluvia ohjelmia. Avaintekijänä tuoteperheissä on ohjelmistojen yhtenäisen toimintaympäristö ja sovelusten välinen helppo tiedonsiirto.

Keskittämällä ohjelmistotarjontansa tuoteperheisiin Nokia Data takaa koulutuksen ja käytöneuvonnan tason sekä ohjelmistojen yhteensopivuuden myös tulevan tuotevalikoiman kanssa. Keskittymällä samaan ohjelmistoperheeseen asiakas puolestaan optimoi oman osaamisensa ja turvaa investointinsa arvon.



Anneli Martonen toimii Nokia Datan Tuotemarkkinoinnissa MikroMikoista ja työasemista vastaavana tuotepäällikkönä.

Muistipulmat unholaan

Extended-muisti vaihtuu expanded-muistiksi

Oikean muistikortin hankinta 286-pohjaisiin kokoonpanoihin on ollut usein hankalaa. Nyt mikrojen muistin käyttömahdollisuudet laajenevat 386-suorittimen ja Windows/386-ohjelmiston ansiosta. Mikä ennen oli extended-muistia voi nyt olla expanded-muistia.

Viime numerossa tutuimme extended-muistien toimintaan. Tällä kertaa käsittelemme LIM/EMS-määrittelyn mukaisten expanded-muistien peruskäsitteitä.

Osoiteavaruus

Reaalitilassa olevan suorittimen osoitettavissa olevan muistialueen koko on yksi megatavu eli 1024 kilotavua. Tästä ensimmäiset 640 kilotavua ovat MS-DOS-käyttäjärjestelmän käytettävissä olevaa perusmuistia.

Systeemi muisti varaa yhden megatavun kokoisesta muistista loput eli sijoittuu 640 ja 1024 kilotavun väliselle alueelle. Tätä 384 kilotavun kokoista muistialuetta hyödyntävät esimerkiksi EGA- ja VGA-näytönohjaimet, BIOS, SCSI-massamuistiohjain, expanded muistikortit sekä Windows/386 expanded-muistin hallintaohjelma.

Mahdollinen lineaarinen lisämuisti on extended-tyyppistä muistia, jota voidaan käyttää suojatussa tilassa. Epälineaarinen lisämuisti on expanded-muistia, josta vain osa sivutetaan kerrallaan sovellusohjelman käytettäväksi.

LIM/EMS-muistin kehitys

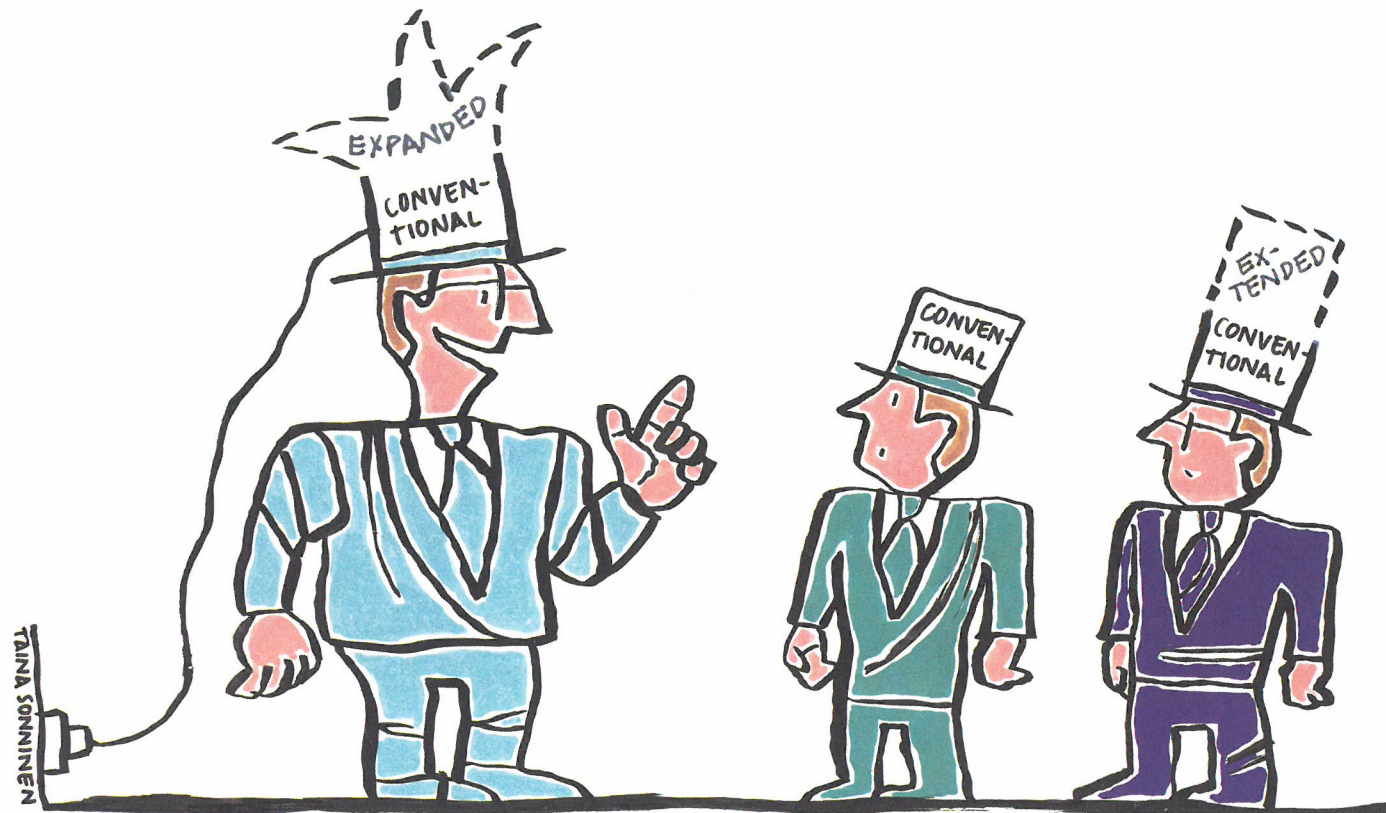
Sovellusohjelmien kehittyessä ja työryhmäverkkojen yleistyessä on yhä useammin törmätty käy-

tettävissä olevan muistin riittämättömyyteen. Jotkut sovellukset kiertävät tämän ongelman expanded-muistin avulla, mutta se ei ole kuitenkaan mikään patenttilääke. Vain ne sovellukset, jotka on erityisesti ohjelmoitu hyödyntämään expanded-muistia, voivat käyttää sitä.

Aluksi expanded-muistin hallintaan kehitettiin useita erilaisia menettelytapoja. EMS- ja LIM/EMS-määrittelyjen takana olivat Lotus, Intel ja Microsoft. Vastaavasti AST, Quadram ja Ashton-Tate kehittivät oman EEMS-määrittelynsä. Yhdenmukaisen käytännön luomiseksi kehitettiin uusi LIM/EMS 4.0, joka tukee sekä LIM/EMS 3.2:ta että EEMS-määrittelyjä. Lisäksi mukana on joukko aivan uusia piirteitä.

LIM/EMS 4.0 ei ota kantaa laiteympäristövaatimuksiin. Se määrittelee ainoastaan ohjelmistotason liittymän sovellusohjelmiston ja expanded-muistin hallintaohjelman välille.

Pelkkä LIM/EMS 4.0 -yhteensopivan muistikortin käyttö ei kuitenkaan riitä, sillä myös sovelluksen on osattava käyttää uuden määrittelyn uusia toimintoja saadakseen siitä kaiken irti. Vanhat sovellukset eivät siten piiristä entisestään, vaikka uusi määrittely sisältää tuplasti enemmän toimintoja kuin vanha LIM/EMS 3.2. Lisäksi LIM/EMS 4.0 tukee expanded-muistia 32 megatavuun saakka.



Muistinhallinnan toiminta

EMM-muistinhallintaohjelma (Expanded Memory Manager) on expanded-muistin toimintaa hallitseva ajuri. Se mahdollistaa laiteympäristöstä riippumattoman muistin käytön, kunhan ajurin määrittely on sisällytetty CONFIG.SYS-tiedostoon. Useiden expanded-muistikorttien asennusohjelmat luovat nämä määrittelyt automaattisesti.

EMM latautuu perusmuistiin käyttöjärjestelmän käynnistyksen yhteydessä. Samalla voidaan suorittaa expanded-muistin testi sekä tulostaa näytölle tunnistetun expanded-muistin määrä.

Looginen muistisivu on 16 kilotavun kokoinen pala expanded-muistia. Vastaavasti muisti-ikkuna mahdollistaa sivujen ottamisen sovellusten käyttöön. Ikkuna toimii yhdyskäytävänä, jonka avulla EMM-ohjelma siirtää sovelluksen pyytämän sivun 286-suorittimen reaali-tilan osoitevaruuteen.

LIM/EMS-määrittelyn mukaan oletusarvoisesti 4 ikkunaa muodostaa yhtenäisen 64 kilotavun kokoinen lohkon, joka sijaitsee 640–1024 kilotavun välisellä muistialuella. Expanded-muistikortin asennuksen yhteydessä kyseisen lohkon sijainti on määriteltävä siten, ettei se osu päällekkäin muiden samaa aluetta käyttävien ohjelmien kanssa.

Jokaiseen ikkunaan liittyy sivurekisteri, joka osoittaa, mikä muistisivu ikkunaan on sijoitettu. EMM sijoittaa ikkunaan uuden sivun muuttamalla ikkunan sivurekisterin sisältöä.

Muita ominaisuuksia

Niin kutsutun backfill-piirteen avulla on mahdollista kasvattaa perusmuistin kokoa aina 640 kilotavuun asti lisäämällä siihen osa expanded-muistista. Tästä ominaisuudesta on hyötyä vain silloin, kun perusmuistia on alun perin vähemmän kuin 640 kilotavua.

Expanded-muistia hallitseva EMM-ohjelma on suunniteltu ympäristöön, jota käyttää kerrallaan vain yksi sovellus. Niinpä EMM ei suojaa muistia kovin hyvin. Sovellusta ei esimerkiksi estetä saamasta käyttöonsa toisen sovelluksen muistisivuja.

MikroMikko 3TT m215 ja 386-pohjaiset MikroMikot eivät tarvitse erillisiä expanded-muistikortteja, koska osa niiden varsinaisesta muistista voidaan määrittellä ohjelmallisesti LIM/EMS 4.0:n expanded-muistiksi. Kun expanded-muistia lisäksi käytetään vain MS-DOS-ympäristössä ja muiden käyttöjärjestelmien merkitys kasvaa jatkuvasti, erillisten expanded-muistikorttien merkitys vähenee jatkossa huomattavasti.

Käännöskukkasia:

Conventional memory – Perusmuisti
Mapping – Muistisivun valinta muisti-ikkunaan
Page – Muistisivu
Page Frame – Muisti-ikkuna
Page Register – Sivurekisteri
Protected Mode – Suojattu tila
Real Mode – Reaali-tila

Laaja käyttöalue

dBASE IV:n perustana on relaatiotyyppinen tietokantajärjestelmä, joka soveltuu niin kortistojen ja rekisterien ylläpitoon kuin sovelluskehitykseenkin.

dBASE IV:ää on helpointa hallita valikkoperustaisen ohjauskeskuksen välityksellä. Toisena vaihtoehtona on suora komentokieli, jonka avulla voi määrittellä sovelluksia myös erityistarpeisiin. Lisäksi komentokielellä voidaan käsitellä tietokantoja ja suorittaa hakutoimintoja. Kätevampää on kuitenkin käyttää esimerkkikyselyä, jolloin ei ole tarpeen osata komentokielen kyselykäskyjä. Viimeisenä vaihtoehtona on standardin mukainen SQL-kyselykieli.

Toistuvien rutiinien määrittelyä ja sovelluskehitystä varten on useita apuvälineitä. Näyttölomakkeilla saadaan tiedolle haluttu esitysmuoto näytöllä. Raporttimallit tekevät saman tempun kirjoittintulostukselle. Tarrojen tulostusta varten on oma apuvälineensä, jolla esimerkiksi osoitetarrat saadaan kirjoittimelle lomakemuodossa rinnakkain.

Sovelluskehittimen avulla on puolestaan mahdollista kehittää käyttäjän tarpeiden mukaisia valikkoperustaisia sovelluksia.

Uutta ytyä tietokantoihin

Puntarissa dBASE IV

Ohjauskeskus

dBASE IV:n keskeisin käyttöliittymä on ohjauskeskus. Se yhdistää sovellukseen liittyvät tietokannat ja toiminnot yhdeksi kokonaisuudeksi luettelokäsitteen (Catalog) avulla. Päänäytössä on kuusi vierekkäistä valikkoa, joihin sovelluksen tiedostot on ryhmitelty käyttötarkoituksen perusteella. Valittavissa ovat tietokannat, kyselyt, näyttölomakkeet, raporttimallit, tarramallit ja sovellusohjelmat. Ohjauskeskus suorittaa kaikki tiedon haut, päivitykset ja raportoinnit. Kunkin tehtäväkohtaisen valikon yläpäässä vaihtoehtona on CREATE-toiminto, jonka avulla luodaan uusi olio tai sovellus.

Käyttäjän on varsin harvoin tarpeen turvautua dBASE:n komentokieleen tai dBASE-ohjelmiin. Sikäli kuin ohjelmien käyttö on tarpeen, valikkoperustaisia sovellusohjelmia voidaan kehittää ohjauskeskuksen sovelluskehittimen avulla ja liittää ne ohjauskeskuksen sovellusvalikkoon.

Ohjauskeskusta käytettäessä ei tarvita tietoa tiedostojen tyypeistä. Toimintoa valitessaan käyttäjä näkee valittuna olevan tiedoston käyttötarkoitusta kuvaavan huomautuksen. Lisäksi valitun tiedoston nimi siirtyy valikkonsa ylimmäksi merkiksi siitä, että tiedosto on käytössä. Samalla tulevat käyttöön tiedoston mahdollisesti liittyvät muut toiminnot. Esimerkiksi tietokannan mukana saattaa tulla näyttölomake, jolla tietokannan tietoja selataan ja raporttimalli, jolla tietokannan sisältö saadaan tulostettua paperille.

Esimerkkikysely

Ohjauskeskuksen esimerkkikyselyn avulla tietokantatoimintoja on mahdollista kehittää vuorovaikutteisesti ilman komento-kieltä. Kyselyjärjestelmä luo kyselyistä tarvittavat dBASE-komennot, jotka puolestaan muodostavat virtuaalisen tietokannan, joka lopulta sisältää itse valitun tiedon. Tietoa voidaan valita useammasta tietokannasta, se voidaan lajitella ja siitä voidaan laskea yhteenvedoja. Kysely voi myös päivittää tietokannan tietoja. Tämä vähentää osaltaan dBASE-ohjelmien tarvetta.

Näyttölomakkeet ja raportointimallit

Tieto voidaan esittää näytöllä halutussa muodossa näyttölomakkeiden avulla. Näyttölomakkeeseen voi myös liittää ehtoja ja tarkistuksia syötettävään tietoon ja muotoilla näytettävien kenttien esitystapaa. Lomakkeella voidaan esittää myös tietokannoissa mahdollisesti olevat vapaamuotoiset tekstikentät (memo fields).

Kirjoittintulosteet muotoillaan raporttimallien avulla. Ne voivat olla lomaketyyppisiä tai muodostua sarakkeisiin jaettuista riveistä (columnar report). dBASE IV sisältää kirjoittintulostuksen erikoistapauksena myös tarratulostuksen. Tarralomakkeella määrätään tarran malli ja sijainti tulosteella.

Näyttölomakkeita ja raporttimalleja voidaan soveltaa tietokantojen lisäksi myös esimerkkikyselyihin. Tällä tavoin lomake-

dBASE IV on Ashton-Taten tietokantaperheen uusi tulokas. Se on edeltäjänsä dBASE IIIPLUS:aa huomattavasti monipuolisempi ja laajempi, mutta kuitenkin sen kanssa täysin tiedostoyhteensopiva. Lisäksi kaikkia dBASE IIIPLUS:lla tehtyjä sovelluksia voidaan käyttää sellaisenaan dBASE IV:ssä.

keilla ja malleilla voidaan valita ja käsitellä useista tietokannoista yhdistettyä tietoa.

SQL-kyselykieli

dBASE IV:n SQL on standardin mukainen täydellinen toteutus, ja se noudattaa SAA-arkkitehtuuria. SQL-komentoja voidaan antaa suoraan komentotilasta tai niitä voidaan upottaa itse sovel-lusohjelmaan. Ennen SQL-käskyjen käyttöä dBASE on asetettava SQL-tilaan, jossa myös suurin osa dBASE-komentokielen käskyistä on edelleen käytössä.

dBASE kääntää SQL-komennot ensin dBASE-komentokielle ja vasta tämän jälkeen suorittaa ne. dBASE IV:n myöhemmän version toteutetaan yhteys Ashton-Taten ja Microsoftin yhteistyönä kehittämään lähiverkoissa toimivaan SQL-tietokantapalvelimeen. Tämä mahdollistaa tehokkaan käyttäjä-palvelin-sovellusarkkitehtuurin myös tietokantojen käsittelyssä.

Sovelluskehitin

dBASE IV:n sovelluskehitin tuottaa hierarkisia valikkoperustaisia sovelluksia. Sovellus muodostuu päävalikosta, josta kutsutaan haluttua toimintoa. Toiminto voi olla näyttö-, raportti-, tai tarramalli, kysely, alivalikko tai dBASE-ohjelma. Sovelluskehitin osaa hyödyntää myös dBASE:n komentokieltä.

Sovelluksen voi kehittää yhtä hyvin top down kuin bottom up -menetelmällä. Top down -menetelmässä kehitys aloitetaan päävalikosta, johon sijoitetaan siitä kutsuttavat toiminnot. Tämän jälkeen määritellään kukin toiminto omalla kehittimellään ja liitetään sovellukseen. Vaihtoehtoisesti aluksi voidaan kehittää erillisiä toimintoja ja käyttää niitä ohjauskeskuksesta käsin. Toimintoja yhdistävä sovellus kehitetään sitten myöhemmin tarpeen mukaan.

Kukin kehitin muodostaa käyttäjän määrittelemistä sovel-lusolioista sen tyyppin mukaisen suunnittelutiedoston. Sovelluskehitin käyttää koodin generointiin kunkin sovelluksen suunnittelutiedoston lisäksi myös erilaisia sovellustyyppikohtaisia mallitiedostoja. Yhdessä ne määräävät, miten sovelluksen koodi lopulta kirjoitetaan.

Mallitiedostot on kirjoitettu erityisellä mallikielellä (template language). Esimerkiksi dBA-

SE-sovelluksen koodin luomista varten on kaksi mallitiedostoa, joista toista käyttäen tuotetaan sovelluksen tekstimuotoinen dokumentti ja toisen avulla itse dBASE-kielinen ohjelma.

Mallikielen avulla on mahdollista sovittaa sovelluskehittimen koodin generointi hyvin pitkälle yksilöllisiä tarpeita vastaavaksi. Koodigeneraattori voidaan ohjata tuottamaan tarpeen tullen jopa jotain aivan muuta kuin varsinaista dBASE-ohjelmaa. Tämä avaa mahdollisuuksia, joita ei voida vielä edes täysin ennakoida.

Komentokieli

dBASE:n komentokieli on pitkälle kehitetty neljännen sukupolven sovelluskehityskieli. Se on lohkorakenteinen ja sisältää joukon ohjausrakenteita sekä välineet aliohjelmien hallintaa varten.

Tietorakenteista löytyy perustietotyyppejä lisäksi esimerkiksi kaksitulotteiset taulukot. Komentokieli sisältää myös suuren määrän syöttö- ja tulostusrutiineja muun muassa valikkojen ja ikkunoiden hallintaan. Lisäksi dBASE-sovelluksesta käsin voidaan käyttää binaarisia aliohjelmia.

dBASE:n komentokieli sisältää sekä korkean tason toimintoja, että lukuisia yksinkertaisia, järjestelmän yksityiskohtaisen ohjauksen mahdollistavia rutiineja. Tässä suhteessa dBASE:n komentokieli poikkeaa MS-DOS-ympäristön tietokantatarjonnasta edukseen.

Verkkoympäristön vaatimukset

dBASE IV tukee työskentelyä verkkoympäristössä. Järjestelmä huolehtii automaattisesti tietue- ja tiedostolukituksista, joten yksittäiskäyttöön kehitetyt sovellukset toimivat verkkoympäristössä ilman muutoksia. Automaattisen näytönpäivityksen myötä kaikkien työasemien näytöillä oleva tieto pysyy ajan tasalla.

Verkkokäytössä keskeisen tärkeään tietosuojaan on dBASE IV:ssä kiinnitetty erityistä huomiota. Jokaisella käyttäjällä on oma tunnus ja salasana. Käyttäjät on mahdollista koota myös ryhmiksi. Tietokannat voidaan salakirjoittaa, jolloin niitä ei voi lukea ulkopuolisilla ohjelmilla. Lisäksi kullekin tietueelle tai si-

tä suuremmalle yksikölle voidaan määritellä oma käyttöoikeustaso kahdeksasta eri vaihtoehdosta. Jokaisella käyttäjällä, joka haluaa päästä tietoon käsiksi, on oltava vähintään sama käyttöoikeustaso kuin kyseisellä tiedolla. dBASE IV huolehtii tarvittaessa automaattisesti tiedon eri näkemyksistä siten, että tietokannoista esitetään vain ne kentät joihin käyttäjällä on pääsy.

Tiedon eheyden takaamiseksi ohjelma käyttää tapahtumape-rustaista tiedon päivitystä (transaction processing). Tällöin useampi yhteen kuuluva operaatio määritellään tapahtumaksi. Järjestelmä takaa tapahtuman suorituksen joko kokonaisuudessaan tai ei ollenkaan. Esimerkiksi laitevian tai sähkökatkon vuoksi osittain suoritettu tapahtuma voidaan peruuttaa kokonaan ennen uutta yritystä.

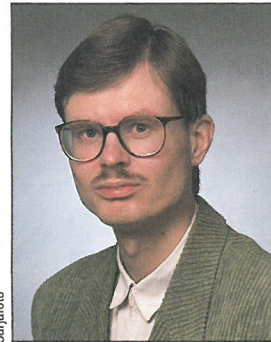
Järjestelmän toiminta ja versiot

dBASE IV on toimintaperiaatteeltaan kääntävä tulkki. Se kääntää automaattisesti sovelluskehittimien tuottamat sovellustiedostot kun niitä käytetään ensimmäisen kerran.

dBASE IV on edeltäjiinsä nähden yhteensopiva eli dBASE IIIPLUS:lle kehitetyt tietokannat ja sovellukset toimivat dBASE IV:ssä sellaisenaan. Lisäksi dBASE IV:llä voidaan kehittää sovelluksia dBASE IIIPLUS-ympäristöön kunhan pitäydytään yhteensopivassa komentokielen osajoukossa.

dBASE IV -ohjelmistoa on saatavana perusversiona ja suunnittelijan versiona. Jälkimmäiseen sisältyy edellisen lisäksi mallikieli ja ohjelmisto ajonaikaisten sovellusten tekemiseen. Suunnittelijan versioon sisältyy myös oikeus levittää ajonaikaisia sovelluksia rajoituksetta ja käyttöoikeus kolmelle käyttäjälle verkkoympäristössä. Käyttöoikeuksia verkkoympäristöön toimitetaan viiden käyttäjän erissä LAN Pack:n yhteydessä.

dBASE IV on laaja järjestelmä. Se vaatii runsaasti levytilaa, ja kiintolevy on käytännössä välttämätön. dBASE IV:n käytön pahin ongelma lienee suuri keskusmuistin tarve. Ensimmäinen versio vaatii 516 kilotavua vapaata keskusmuistia toimiakseen kaikilta osin. Etenkin verkkoympäristössä vaatimusta on vaikea täyttää. ■



Sarjafoto

Martti Melen toimii ohjelmistoasi-
antuntijana Nokia Datan Kehityk-
sen Productivity Tools -ryhmässä
vastuualueenaan tietokantaohjel-
mistot.

**Interleafin TPS-ohjelmis-
topehe (Technical
Publishing Software) on
jo jonkin aikaa tuonut
julkaisutoiminnan ilot ja
surut myös työryhmä-
verkkojen käyttäjien
ulottuville. Saman doku-
mentin työstäminen kun
on onnistunut niin SUN-
työasemalla kuin Mikro-
Mikollakin.**

**Nyt TPS:stä on tullut
markkinoille uusi versio,
joka sisältää paljon muu-
takin mukavaa kuin
kyvyn verkkotyöskente-
lyyn.**

Siinä missä työryhmäver-
kon laitemerkki-riippu-
mattomuus on eittämättä
etu, on sillä myös haittansa.
Vaikka koneiden liittämiskortit
kommunikoivat keskenään suju-
vasti, voi tiedon sisäinen esitys-
muoto olla joka järjestelmässä
erilainen – eri ohjelmistojen
omista systeemeistä puhumatta-
kaan.

Interleaf on kuitenkin oivalta-
nut, että vaikka työasemat olisi-
vat epäyhteensopivia, ohjelmi-
stot voivat olla hyvinkin yhteen-
sopivia. Ja kiintolevytilahan on
verkon yhteistä omaisuutta...

Ei siis ihme, että jo TPS
3.0:lla on ollut vankka asema
teknisessä dokumentoinnissa.
Uuden version uudet ominaisuu-
det laajentavat käyttöaluetta
kuitenkin myös hallinnon ja
markkinoinnin puolelle.

TPS-ohjelmistoperhe

Interleafin TPS-perhe koostuu
useista eri järjestelmille tehdyistä
ohjelmista. SUNin, Apollon
ja DECin työasemille löytyy kul-
lekin oma TPS-paketti siinä mis-
sä MS-DOS-koneille on oma
PCEditor ja PCViewstation.

Varsinainen TPS 4.1 jakautuu
itse asiassa kolmeen osaan. Oh-
jelmiston sydämenä on Perus
TPS, joka sisältää kaikki tekstin-

Interleaf TPS 4.1

Julkaisutoimintaa verkossa (ja vähän muuallakin)



käsittelytoiminnot sekä tärkeim-
mät grafiikka- julkaisunhallinta-
ja taitto-ominaisuudet. Grafiikka-
ja julkaisunhallintatoimin-
noilla laajennettuna ohjelma laa-
jenee täydeksi TPS:ksi, joka ky-
kenee esimerkiksi taulukoiden
editointiin, isometrisen grafiikan
käsittelyyn, värierotteluun sekä
tiedon salaukseen ja valikoin-
tiin.

UPS (University Publishing
Software) on TPS:n korkeakou-
luille suunnattu sisärjestelmä.
Edullisemmän hankintahinnan
lisäksi myös sen ominaisuudet
tukevat erityisesti yliopistokäyt-
töä.

PCEditor mahdollistaa TPS-
dokumenttien käsittelyn myös
MS-DOS-tiöasemilla. Käytän-
nössä PCEditor on täysverinen
tekstinkäsittelyohjelma, jonka
avulla on mahdollista määritellä
myös tekstin ilmiäsu varsinaises-
sa TPS-dokumentissa. Teksti voi-
daan tulostaa sekä laserkirjoitti-
mella lopullisessa muodossaan
että matriisikirjoittimella vedok-
sena.

Kevään mittaan markkinoille
tuleva PCViewstation tarjoaa
puolestaan helpon tavan TPS-
julkaisujen levittämiseen MS-
DOS-ympäristössä. Sen avulla
on käytännössä mahdollista
unohtaa paperi koko julkaisutoi-
minnasta. Dokumenttien sivuja

voidaan selata ja lukea
PCViewstationin avulla sekä et-
siä haluttuja aiheita tai tarken-
taa lähemmäs paremman reso-
luution saavuttamiseksi. Myös
paperitulostus käy toki päinsä.
Esimerkiksi jatkuvasti päivitty-
vien laajojen teknisten asiakirjo-
jen ylläpito ja jakelu onnistuu
näin kätevästi, eikä joka nurkka
ole täynnä vanhentunutta A4-
saastetta.

Pelkän katselemisen lisäksi
PCViewstationin avulla voi
myös kommentoida itse doku-
menttia. Näin kukin tekijöistä
voi selata lopullisen dokumentin
oikolukuvedosta omalla MS-
DOS-tiöasemallaan ja liittää sii-
hen mielipiteensä. Samaten esi-
merkiksi kiertokirjeiden kom-
mentointi onnistuu PCViewstati-
onin avulla kätevästi.

Monta tekijää

Verkkokäytössä TPS-dokumen-
tien tekoön voi osallistua useita
tekijöitä. Julkaisutoiminnan ve-
täjä jakaa dokumentin useisiin
eri osiin, joihin eri tekijöillä on
pääsy. Osa voi keskittyä pelkäs-
tään kirjoittamiseen, osa huo-
lehtii kuvituksesta, jotkut päivi-
tävät tietokantoja tai sitten teh-
täviä voidaan yhdistellä juohe-
vasti.

Koska kukin tekijöistä käsittelee suoraan lopullista dokumenttia, varsinaista yhdistelyvaihetta ei tarvita. Samanaikainen työskentely on kuitenkin mahdollista vain dokumentin eri osissa. Kun yksi osa on käsiteltyssä, se on muilta suljettu. Muuten jomman kumman työstäjän panos menisi hukkaan.

Vielä laajempi joukko tekijöitä saadaan mukaan kun julkaisuun luetaan tekstiä tietokannoista. Eräessä käytännön järjestelmässä kootaan vuosittain tiedot kaikista Yhdysvaltain korkeakouluista yhteen kirjaan. Ennen työ hoidettiin käsin kyselylomakkeiden, latojen, oikolukijoiden ja taittajien avulla. Nyt kaikki korkeakoulut on koottu tietokantaan, josta koko kirja saadaan luettua kerralla yhdeksi TPS-dokumentiksi. Kun tiedot taannoin ehtivät vanhentua ennen kuin kirja ehti edes painoon, voivat tekijät nyt aikataulun asemesta keskittyä kirjan ulkoasuun ja tilastojen esitysmuodon parantamiseen.

Hallittua toimintaa

Osa dokumentista voidaan myös piilottaa eli koko julkaisu ei välttämättä tulostu käyttäjän näytölle tai kirjoittimelle. Käytännössä tämä mahdollistaa turhiksi koettujen tietojen poisjättämisen tai salattavien tietojen kätkemisen. Julkaisusta saadaan monta versiota eri käyttäjien tarpeisiin yhden version vaivalla.

Vaikka TPS päivittääkin sivutuksen ja palstoituksen automaattisesti jokaisen muutoksen jälkeen, sivutus voidaan myös lukita tarpeen niin vaatiessa. Tämä on erityisen kätevää mikäli sisällysluetteloon ei haluta päivityksen yhteydessä muutoksia. Dokumentille on myös mahdollista määritellä päivitysmerkin-

tä, joka tulostuu kaikkien päivityskohtien marginaaliin.

Numerointiin TPS tarjoaa monia eri vaihtoehtoja oli sitten kyse sivunumeroista, viitteistä, kappaleista tai hakemistoista. Mielenkiintoisimpia ovat alaviitteet, jotka voidaan kytkeä niin tekstiin kuin itse sivuunkin. Alaviite luo myös itselleen tilaa sitä mukaa kun sen koko muuttuu ja vaihtaa tarvittaessa sivua sivutuksen päivittyessä. Mikäli käyttäjä määrittelee samalle sivulle useampia viitteitä, niiden numerointi päivittyy automaattisesti.

Graafikka hallussa

Sen lisäksi, että TPS kykenee yhdistämään dokumentteihinsa erilaisia kuvatiedostoja CAD-, piirto- ja kuvanlukujärjestelmiltä, se sisältää myös itsessään joukon monipuolisia grafiikka-toimintoja.

Teksti- ja grafiikkatiedostojen lisäksi TPS osaa lukea myös taulukkolaskentatiedostoja – ja laatia niiden pohjalta kaavakuvia. Pylväs-, viiva-, piirakka-, pinta-, ja hajontakaaviot syntyvät automaattisesti niin lineaarisina kuin logaritmisinakin. Lähtötiedot voidaan kirjoittaa myös suoraan näppäimistöltä.

Sekä viiva- että digitoitua grafiikkaa varten ohjelmistosta löytyy joukko omia rutiineja mukaan lukien eri harmausasteiden painottamisen ja muokkaamisen. Lisäksi niin tekstiä, viivakuvia kuin digitoitua grafiikkaakin voidaan yhdistää yhteen kuvaan.

Viivagrafiikkarutiineista löytyvät oikeat työkalut myös isometristen kuvien piirtoon. Viivoja ja tasoja voidaan kääntää isometrisessä apuviivastossa tasosta toiseen. Lisäksi käyttäjä voi projisoida ortogonaaliobjekteja kaikille kolmelle isometriasolalle.

Tiedosta taittoon

TPS:stä puhuttaessa sen varsinaiset sivuntaittoiminnot jäävät helposti verkkokäytön ja grafiikan jalkoihin. Sen avulla on kuitenkin helppo luoda esimerkiksi matemaattisia kaavoja kaikenkarvaisine erikoismerkkeineen.

TPS:n peruskäsite on kirja, jonka alle kaikki yhteen kuuluvat dokumentit ja niihin puolestaan kuuluvat tiedostot tai tiedostoviitteet kootaan. Kokonaisuus voi olla jaettu sivuihin, kappaleisiin tai kokonaisiin kirjasarjan osiin, mutta niin kauan kuin ne ovat yhden kirjan sisällä, niitä voidaan käsitellä kokonaisuutena esimerkiksi sivunumeroinnin ja sivutaiton osalta.

Kirja on rakenteeltaan hierarkkinen eli ylin taso voi jakautua osiin, jotka jakautuvat kappaleisiin, jotka voidaan tarvittaessa jakaa edelleen. Käyttäjän ei kuitenkaan tarvitse avata uutta osaa siirtyessään esimerkiksi kappaleen viimeiseltä sivulta seuraavan ensimmäiselle, vaan dokumentti näkyy hänelle yhtenä kokonaisuutena.

Suoran kopioinnin asemesta TPS mahdollistaa myös viittaukset toisaalla olevaan tekstiin tai grafiikkaan. Näin esimerkiksi usein sivun alalaidassa toistuva huomautus ei vie muistitilaa kuin yhdessä paikassa. Vastavasti kopiot päivittyvät sitä mukaa kun alkuperäistä muutetaan.

Kuvien ja grafiikan suhde voidaan määritellä joustavasti. Teksti voi joko kiertää koko grafiikkaruudun, olla välittämättä siitä tai myötäillä ruudun sisällä olevaa varsinaista kuvaa. Tekstiä voidaan myös kallistaa, ja kallistettua tekstiä editoida.

Interleaf Suomessa

TPS tukee kaiken kaikkiaan 13 eri kieltä tavutuksen ja 11 kieltä oikeinkirjoituksen tarkistuksen osalta. Suomessa toimitettaviin TPS-paketteihin installoidaan vakiona suomi ja Englannin englanti, mutta myös muut kielet (US englanti, UK englanti, ruotsi, norja, tanska, saksa, hollanti, ranska, espanja, italia, kanadan ranska, suomi ja portugali) ovat saatavissa. Suomeksi ja portugaliksi TPS osaa vain tavuttaa.

Nokia Data markkinoi Interleafin julkaisuohjelmistoa Suomen lisäksi Norjassa ja Tanskassa. Suomessa Interleafin tuotteita löytyy jo yli 50 järjestelmästä, mikä vastaa noin 50% markkinaosuutta UNIX-pohjaisesta julkaisuutoiminnasta. Lisenssejä on yli 200, ja Suomesta löytyy myös Interleaf-käyttäjäkerho IFUG.

Koska TPS:n hinta on äskettäin uuden version myötä laskenut huomattavasti, markkinoiden uskotaan laajenevan nopeasti myös muihin kuin pelkässä julkaisukäytössä oleviin järjestelmiin.

Interleaf TPS 4.1

Syöttö

CAD-tiedostot:

HPGL, Calcomp 925, Calcomp 960, IGES

Taulukkolaskentadiagrammit:

Lotus 1-2-3, Multiplan, Excel

Grafiikkatiedostot:

MacDraw, MacPaint, Postscript, TIFF

Tekstietiedostot:

WordPerfect, MS Word RTF v. 4, MacWrite, DCA, ASCII, nroff, troff, Wang, SCRIBE, Digital DX

Tulostus

laserkirjoittimet:

PostScript, Imagen, Interleaf, Interpress (tulossa)

Valoladontakoneet:

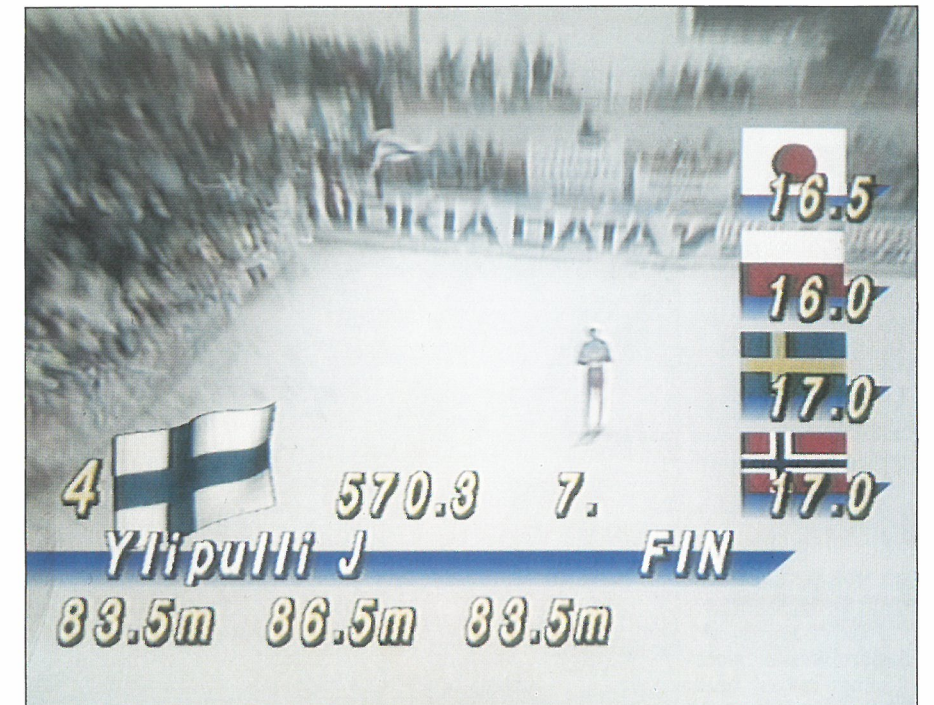
Linotype Linotronic 100/300, Compugraphic 8600G, 9600, Monotype Lasercomp Series Mk2, Pioneer

Saatavuus

SUN 2, SUN 3, SUN 386i

Hintatiedot (yksittäiskappaleiden lähtöhinnat)

– TPS 17.500 mk
– UPS 8.600 mk
– PCE 4.600 mk
– Päivitys WPS -> TPS 9.900 mk



Lahden MM-kisat olivat silkkaa juhlaa paitsi suomalaisille urheilijoille myös suomalaiselle tulospalvelujärjestelmälle. Laitteiston Lahden kaupungille oli toimittanut Nokia Data ja ohjelmistot Raimo Inberg Oy.

Arvonnasta maalivalvontaan

Lahti MM89 tietokoneen silmin

Nykyisen tulospalvelujärjestelmän juuret juontavat vuoden 1978 MM-kisoihin, jolloin Lahden kaupunki tilasi Oy Nokia Ab Elektronikalta edellisen tulospalvelujärjestelmän uudistamista. Kun tulospalvelun modernisointi tuli ajankohtaiseksi, Nokia Data saattoi tarjouskilvassa esittää vanhan järjestelmän uudistamista. Useat kilpailevat järjestelmät olivat myös varsin jäykkiä eikä niiden soveltaminen tuleviin urheilutapahtumiin olisi ollut mutkatonta.

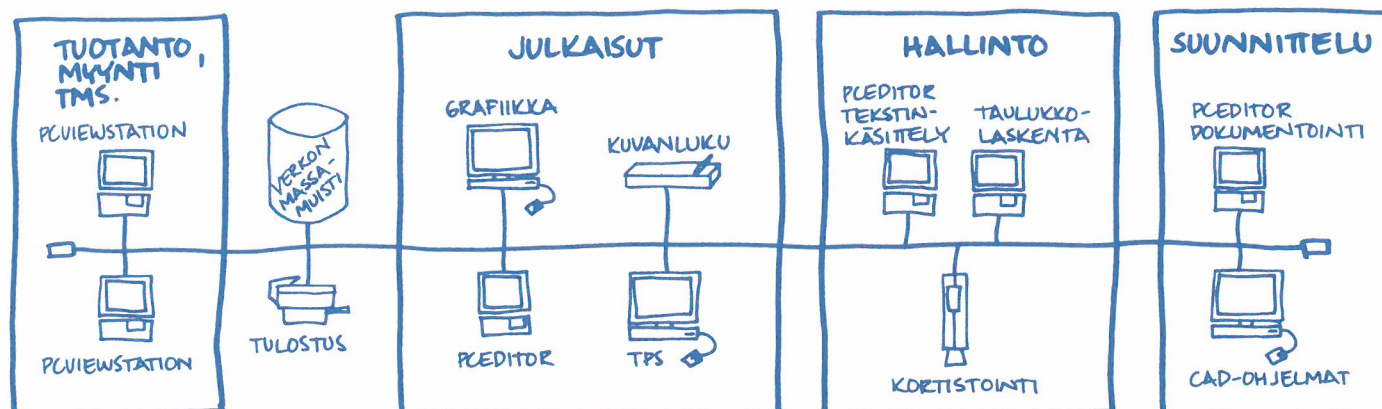
Henkilörekisteri on ensisijaisen tärkeä myös turvajärjestelyissä eli kulkulupien ylläpito ja tarkistus onnistui kisojen aikana joustavasti.

Akkredointitietoja käytettiin hyväksi myös eri lajien lähtöjärjestysten arvonnassa. Useissa arvokisoissa arvonnat suoritetaan edelleen käsin, mutta Lahden kisoissa käytettiin kahta mikrotietokonetta, joista toisella arvottiin satunnaisesti arvontaryh-

Ihmiset paikoilleen

Akkredointi oli yksi järjestelmän tärkeimpiä tehtäviä. MM-

Työryhmäverkko



Nykyaikaisen tulospalvelujärjestelmän valvomo muistuttaa lähinnä lennonjohtotornia.



män seuraava kilpailija ja toisella kyseisen kilpailijan lähtönumero.

Tietokoneistetun arvonnin seurauksena lähtöluettelot saatiin välittömästi arvonnin jälkeen niin tiedotusvälineille kuin tulospalvelujärjestelmällekkin.

Ladulle

TV-lähetyksissä nähtyjen tilan-
tietojen tuottamiseksi tarvittiin paitsi kalustoa myös ihmisiä niin lähtöpaikalla, maalissa kuin väliaikapisteissäkin.

Tärkeimpiä ja tarkimmin mitattuja tietoja olivat lähtö- ja maaliintuloajat. Normaalisti tähän riitti lähtöportin ja maalin infrapunavalokellon seuranta, mutta esimerkiksi ennenaikaisissa ja myöhästyneissä lähdöissä oli se vaara, että lähtö osui järjestelmän tunnistaman ± 3 sekunnin lähtöikkunan ulkopuolelle, jolloin lähettäjän täytyi ilmoittaa lähdöstä erillisellä varalähtöpainikkeella.

Maalissa oli puolestaan kaksi aktivointipainiketta, joista toinen sai tulostaululla ja toinen TV-ruudulla juoksevan vertailuajan pysähtymään seuraavasta kennon ohi menevästä säärestä. Jokaisesta valokennon pulssista välittyi tieto tulospalvelujärjestelmälle, jonka lokia sitten ver-

rattiin toimitsijoiden ilmoittamaan maaliintulojärjestykseen.

Väliaika-asemien ennakko-päätteiltä ilmoitettiin tulospalvelulle aina noin puoli minuuttia etukäteen hiihtäjän lähestyvän asemaa. Näistä tiedoista muodostettiin erillinen lähestyjien lista, jonka avulla esimerkiksi TV-ohjaaja saattoi siirtää kuvan hyvissä ajoin oikealle asemalle.

Varsinainen väliaika kellotettiin väliaikapäätteen näppäimen painalluksella. Esimerkiksi 50 kilometrin hiihdossa maastossa oli kaikkiaan viisi eri asemaa ennakko- ja väliaikapäätteineen.

Hyppyrillä

Mäkihyppysä järjestelmän tuli kyetä käsittelemään paitsi pituus- ja tuomaripisteet myös mittaamaan hyppääjän nopeus hyppyrin nokalla. Mittauslaitteen toimi Alge Timer * S3, joka puolestaan välitti tuloksen sekä hyppyrin nokalla olevalle näyttöyksikölle että tulospalvelujärjestelmälle edelleen TV-ruuduille tulostettavaksi.

Hyppykierros ja kilpailijan numero ilmoitettiin järjestelmälle tuomaritornissa olevan keskitimen näppäimistöltä. Hypyn pituus ja tuomaripisteet kulkivat tämän keskitimen kautta, ja tietoja oli mahdollista korjata ja

muuttaa aina seuraavan kilpailijan hyppyvuoroon asti. Saatuaan kaikki tarvitsemansa tiedot keskitin laski lopuksi yhteispisteet ja lähetti ne tulospalveluun.

Keskittimellä oli kolme tulostuslinjaa – yksi tulospalveluun, toinen kuuluttamon päätteelle ja kolmas tuomaritornissa olevalle tulospalvelun varamikrolle.

Tulosteita joka lähtöön

Tulospalvelujärjestelmä syötti tietoja kilpailutilanteesta jatkuvasti eri järjestelmiin. Tärkeimpiä olivat TV-lähetyksien tilan-
tiedot sekä kisapaikan tulostaulu, mutta lisäksi järjestelmään kuului kisa-alueen sisäinen TV-verkko, telset ja vino pino paperitulosteita.

Televisiointia varten oli olemassa käytännössä kaksi erillistä järjestelmää. TV A eli Ylen grafiikkalaitteisto vastasi normaalisti tulosteiden graafisesta ilmasusta, mutta kykeni tarpeen vaatiessa käsittelemään myös tulostietoja itsenäisesti. TV B vastasi puolestaan tulostettavien tulostietojen käsittelystä ja valinnasta ennen niiden lähettämistä varsinaisille esityslaitteille. TV B kykeni myös määrittelemään millaista grafiikkaa ja millaisia merkkejä milloinkin tulostettiin.

Kisojen sisäinen TV-verkko käsitti kahdeksan kanavaa, joista enimmillään kaksi oli varattu Ylen kisalähetyksille. Yhdellä kanavalla oli jatkuvasti päivittyvä tapahtumalista, jolle tulostettiin käynnissä olevan kilpailun 24 viimeisintä tapahtumaa. Muilla kanavilla esitettiin 20 sekunnin välein päivittyviä tapahtumapaikkojen tulostilastoja.

Telset-yhteyden kautta välitettiin tulostietoja Päijät-Vision kaapeli-TV:lle.

13000 lampusta koostuvasta tulostaulusta oli vastaamassa PDP-11/05-tietokone, joka oli puolestaan yhteydessä tulospalveluun. Tiedonsiirto tapahtui 4800 baudin nopeudella.

Käyttöä riittää

Lahden kisojen tulospalvelujärjestelmä ei jää kisojen jälkeen keräämään pölyä vaan sitä hydynnetään useissa tulevaisuuskisoissa. Esimerkiksi vuonna 1991 Lahdessa järjestetään ampumahiihdon MM-kisat, joissa järjestelmä on jälleen kuvassa mukana. Mukautumiskyky vaihteleviin tulospalvelutarpeisiin kun on ollut yksi järjestelmän suunnittelun tärkeimmistä kynnysheidoista. ■

Kevin Puskas

Saarisen Tapsan muistelmat osa IV

Petissä

**–Herättää! Katso-
kaa noita koiria ja
katsokaa noita
seiniä! Eikö niissä
ole reikiä ilman
nauloja?**

Tietokoneasiantuntija Tapsa Saarinen heräsi omaan huutoonsa, nousi hätäpäissään ylös kuin töihin lähtökseen, kunnes aisti hiestä läpimärän yöpukunsa ja palasi takaisin levyttämään kapeaan vuoteeseensa peiton alle. Kolmatta viikkoa kestänyt raju tauti kuumineen oli ehkä vihdoin siis hellittämässä. Tapsa nousi uudelleen ja lähti keittiöön vettä juomaan. Kaikki astiat olivat likaisia, hän valitsi vähiten harmaantuneen kupin ja ryysti nesteeseen ahnaasti janoiseen kurkkuunsa.

Tapsa oli virittänyt tyyntä vuoteen päätyyn niin, että pystyi makuulta tutkimaan ikkunas-
taan kaupungin ytimen, vanhan torin elämää. Siitä saattoi hyvin seurata kotikaupungin symbolin, mustamakkarakarajonon etene-
mistä pakkasessa hengityspilvi huuruten. Tuttuja vilahteli ohi päivän mittaan – muttei poikennut. Tuskin tiesivätkään Tapsan alennustilasta. Hänhän oli tunnettu miehenä; joka ei sairastanut, ja raahautui tarpeen tullen työpaikalleen ilman toista jalkaa. Sairaslomaa hän ei ollut pitänyt koskaan aiemmin.

Mutta nyt oli Tapsan takki tyhjä. Ehjätäkään jalat eivät tah-
toneet kantaa, lukea ei jaksanut ja toripöydillä isoa mustaa ahmi-
vien äijäporukoiden juttujen

huulilta lukeminen oli raskainta, mihin hän oli päiväkausiin pystynyt. Ne tarinat Tapsa tunsi vuosien takaa, hän oli usein ihmetellyt miten kaikki maan parhaat urheilijat ruokatunnillaan etsiytyivätkin juuri tälle torille muistelemään, kuinka juuri hänestä olisi tullut huippufutari, mutta kun mutta kun jne. niitä loputtomia verukkeita ja selityksiä. Jollakin oli vaimo ja lapset pilanneet uran, toisella nivelsiteet rutkahtaneet juuri ennen lopullista urannousua ja kolmas joutunut valmentajan silmätikuksi ilman omaa syytään. Maajoukkue-
tason kiekkoilijoita parveili torilla aina pari kenttällistä, mutta yhteinen mielipide oli, että kaupungista ei löytynyt yhtään kunnan valmentajaa. Niiden syyt jatkuva kaljaputkikin oikeastaan oli. Suomi on urheilumaa.

Tapsa havahtui postin kolahdukseen, siellä makasivat entisetkin hurjassa läjässä, jonka yli naapurin poika nosti hänelle pikku korvausta vastaan hakeman-
sa kauppakassin. Vieraat joutuisivat sisään tullessaan jo melko lailla aittomaan, mutta eipä ollut ketään odotettavissakaan, koska myyntitykki Mäkinen oli vielä tiettävästi Rautayhtiön ukkojen kanssa perinteisellä hiihtoviikolaan Ivalossa. Sekin reissu jäi Tapsalta ensi kertaa sivuun. Juomatta jäävät nesteekään eivät Tapsaa harmittaneet, joten taudin kuvan täytyi olla musta ja murheellinen.

Ero Leenan kanssa oli tapahtunut jo kuukausia sitten. Tuore aviovaimo ei jaksanut käsitellä asiantuntija Saarisen kiintymystä rakkaaseen mikroonsa, jonka näyttöruutu köllötti parivuoteen vierellä heidän yhteiselämänsä todistamassa, eikä Tapsa suostunut sitä lukuisista pyynnöistä huolimatta eteiseen siirtämään. Työaikojaan Tapsa ei oppinut



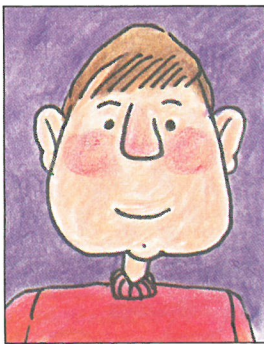
SQLWindows-kurssi

Osa I – perusteet

myöskään muuttamaan, vaan kamppaili vanhaan tapaan haasteellisten projektien kimpussa vuorokaudet ympäri. Satunnaiset kaljareissut myyntitykki Mäkinen kanssa, jotka joskus saattoivat todella venähtää, täyttivät lopulta Leenan mitan. Kysymyksen ”Mikro vai minä?” ei Tapsa osannut vastata oikein mitään ja Tapsan edelleen miettiessä keräsi Leena tavaransa ja lähti. Saarinen jäi keittiöön haukkaamaan lauantaimekkaraa Rauta-Yhtiön driverin sitkeä vika päässään pyörien.

Nyt Leena odottaa jossakin muualla hänen lastaan, ei tietokoneen missä. Toivottavasti lapsi on terve vaikka minun onkin mietiskeltävä Tapsa peittoja korviin kiskoissaan. Jos se on poika, kasvatat siitä futarin ja näytän noille torilla norkoaville makkararöyhtäilijöille. Mutta on se kumpi vaan, niin yritän pitää huolen siitä, ettei ainakaan atk-asiantuntijaksi rupea. Ei tämä ole ihmistä varten. Parinkymmenen vuoden aikana en ole pitänyt lomaa oikeastaan lainkaan, kuhkinut kaiket yöt sekalaisissa konesaleissa rakentelemassa lopulta melko tarpeettomia eksoottisia virityksiä, vaikka standardeillakin olisi varmasti pärjätty. Kehityskin on niin kovaa, että jos vapaa-aikaa jää, kannattaa sekin uhrata vaikka uusien lehtien ja manuaalien tutkimiseen. Synkkien pohdiskelujen jälkeen päättyi Tapsa kuitenkin ennen nukahtamistaan siihen tuttuun tulokseen, että ”on tää kaikesta huolimatta helvetin hienoakin ollut”.

Puhelin soi.
– Saamisen Tapsa.
– Tasanen täällä, moro.
– Moro, moro.
– Meinaatko enää ollenkaan hommiin tulla? Täällä on, jumalauta, kasa avoimia keikkoja pöydälläsi ja Rauta-Yhtiö soittaa pari kertaa päivässä ja kysyy, koska linjaongelmat ratkeavat.
– Mä olen kipeä. Kauhea kuume.
– Olethan sinä ennenkin sairastanut työtä tehnyt!
– Mutta nyt ei kykene. Hyvä kun vessaan omin avuin jaksaa.
– Minä otan toisen tilallesi!



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

– Senkun. Tiedät hyvin, ettet toista samanlaista löydä.
– Valitettavasti näin on. Ainoa hyvä puoli taudista on se, että voin liittää ylityörahasi omaan edustusbudjettiini, terve!
– Hauska kuulla, sairas vaan itellesi!

Tiukka-Tasanen oli jokin aika sitten nimennyt Tapsan mukaan ”tulevaisuuden projektiin”, jonka muut jäsenet olivat kaikki nuoria juuri valmistuneita diplomi-insinöörejä. Tapsan piti tuoda ryhmään kokemusta ja näkemystä, kuten Tasasen visioon kuului. Heti alusta Tapsa jäi ryhmän ulkopuolelle. Nuo nuoret puhuivat aivan eri kieltä, tahtoivat käyttää valmisohjelmia ja kehittämiä ja vieroksivat Tapsan himoa konekieliseen koodaamiseen. Saariseen suhtauduttiin kuin vanhojen hyvien aikojen liikkuvaan muistomerkkiin, jolle ei tiedetty mitä tehdä. Kun muut lähtivät iltapäivällä skimaamaan, Tapsa jäi yöksi rakentamaan osuuttaan valmiiksi ja saikin porukan kunnioituksen hankittua, kun huomattiin tämän vanhan fossiilin norjalaisvilapaidassaan tekävän viiden miehen työt. ”Tapsa tekee, muut hiihtää ja visioi”, oli Tiukka-Tasanen mielipide uudesta iskuryhmästään, jonka aikataulut yllättäen alkoivat pitää. Saarinen sai lempinimen ”konekielen kaljuuntuva kunniaivanhus”.

Tapsa ei kuitenkaan viihtynyt, ja virkistykseksi hän ryhtyi jahtaamaan nuorta DI-tyttöä, jolla oli joka päivä eri asu, mutta muuten tuntui ymmärtävän vähän tietokoneidenkin päälle. Projektin saunaillassa Tapsa vihdoin onnistui. Muiden vatkaessa näkemyksiään alan tulevaisuudesta ja rakentaessa funktiionaalista semantiikkaa, Tapsa suostutteli tytön saunan puolelle, missä tämä suostui Tapsan ehdotuksiin sillä ehdolla, että ensin käytäisiin yhdessä avannossa. Näin tehtiin, ja sieltä oli keuhkokuumeeksi muuttunut tautikin peräisin. Tapsa päätti jatkossa varoa nuoria DI-tyttöjä, jotka harrastavat avantouintia. Paljon puhutaan turvaseksistä, mutta ei muisteta urheilun vaaroja!

Ovikello soi, ja hetken kuluttua myyntitykki Mäkinen työnty omilla avaimillaan yli lehtipöydän peremmälle. Mäkinen näytti viikon hiihtoreissun jäljiltä vähintäänkin rispaantuneelta.

– Terveisiä Ivalosta.
– Moro, sinä näytät kamalalta.
– Niin sinäkin, eikö kuume vieläkin hellä?
– Ei millään, kohta kolme viikkoa petissä, arvaa harmittaako?
– Arvaa harmittaako Tasasta?
– Joo, se soitti äsken. Miten Ivalon keikka meni?
– Armoitonta sikailua, perinteiseen tyyliin. Kämppä jäi sentään ehjäksi. Menessä jo pojat tyhjänsivät pari muovikassillista. Kun päästiin perille, niin naapurin poroisäntä oli sovittuun tapaan vastassa ja kuljetti meidät majalle lumikelkalla. Siihen meni kolme reissua, minä ja Kujala jäätin viimeiseksi. Kun poromies tuli meitä hakemaan, rojahti Kujala ahkion pohjalle tuhina miehenä ja se halesi. Poromies kiroi perkeleesti ja lähti hakemaan uutta peliä kotaa. Kun se palasi, Kujala oli jo sammunut hankeen, ja poro otti vain sen kyytiin, löi minulle suket käteen ja sanoi, että sinähän hiihdät. Jumalauta, siihen meni kaksi tuntia ja meinasin eksyä tunturiin. Se taisikin olla viikon ainoa urheiluluosus. Täytyy myöntää, että tuli vähän otettua. Käytiin me sentään avannossa uimassa ...

– Avannosta et kyllä puhu minulle mitään!
– Puhelin keskeytti Mäkinen kerroksen. Tapsa väentäytyi pysähtymään.
– Saarisella.
– Leena täällä, hei!
– Hei, missä sinä olet?
– Sairaalassa, meille on syntynyt poika, tuletko katsomaan?
– Tulen ja heti.
– Tapsa säntäsi eteiseen ja kaapasi palttoon pyjamansa päälle. Mäkinen hämmästeli potilaan ripeitä otteita.
– Tapsa hei, mihin menet? Minulta jäi kertomatta kun yhtenä yönä ...
– Kerro myöhemmin! Minusta on tullut futarin isä! Siivoa täällä sillä aikaa, yritän saada Leenan tulemaan takaisin! ■

SQLWindows on ohjelmatuote, jonka avulla voidaan tehdä MS-Windowsin päällä pyöriviä näyttäviä sovelluksia huomattavasti helpommin kuin Windows SDK:lla tai C-kielellä. Lisäksi SQLWindowsilla tehty sovellus voi käsitellä SQLBase-tietokantoja, mutta sitä voi käyttää myös pelkän Windows-käyttöliittymän tekemiseen.

Aloitamme nyt neljän artikkelin sarjan, jonka tarkoituksena on opettaa SQLWindows-ohjelmoinnin perusteet. Ensimmäisessä osassa tutustumme SQLWindowsiin ohjelmointiympäristönä tekemällä pienen SAL-kielisen ohjelman. Samalla opimme olio-käsitteen merkityksen. Tarkoituksena on myös tehdä SAL-ohjelman (SQLWindows Application Language) vakiorakenne mahdollisimman tutuksi.

Seuraavissa osissa teemme valikon, keskusteluikkunan tiedon syöttöön ja listboxin tiedon tulostamiseen. Viimeisessä osassa käsittelemme table window-rakennetta, jonka avulla SQLWindows käsittelee tietokantoja.

Kurssi on kohdistettu lähinnä sellaisille henkilöille, jotka jo osaavat käyttää Windows-sovelluksia ja ovat jo ohjelmointeet jollakin ohjelmointikielellä. SQLWindowsissa pärjää toki aika pitkälle pelkästään piirtämällä eli ohjelmointitaitoa tarvitaan vasta, kun määritellään piirretyn näytön osille (olioille) toimintoja.

SQLWindows ohjelmointiympäristönä

Jos olet päätenyt siihen, että Windows-käyttöliittymä sopii hyvin sovelluslukeeseesi, tulee sinun ottaa huomioon myös SQLWindowsin laitteistovaatimukset.

Koska SQLWindows on hyvin monipuolinen paketti, sen ominaisuudet vaativat myös mahdollisimman tehokkaan koneen. Parhaita ovat tietysti 386-koneet, vaikka 286:detkin käyvät. Lisämuistia SQLWindows osaa hyödyntää (muiden Windows-pohjaisten ohjelmistojen tavoin) suoraan vain, jos lisämuistikortti on tyyppiltään expanded (LIM/EMS 4.0). Extended-muistia se pystyy hyödyntämään heittovaihtoon SQLBasen tapaan SMARTDRIVE:n avulla.

Kaiken kaikkiaan SQLWindows pyörii siis parhaiten 386-koneessa, jonka työmuistin kooksi on expanded-kortin avulla saatu vähintään 2 megatavua.

Aloitus

Tämä kurssi lähtee siitä, että sinulla on koneessasi SQLWindows-versio 1.1.0 asennettuna toimituksen mukana seuraavien ohjeiden mukaisesti. Huomaa, että vielä tässä versiossa RAMDRIVE.SYS tai VDISK.SYS aiheuttaa ongelmia tähtäessään eli selviät pahoilta kolhuilta jättämällä sen kokonaan pois CONFIG.SYS:istä.

Kun olet saanut SQLBase-tietokantakäsittelijän (yksittäisympäristössä DBXLOCAL.EXE) ladatuksi ja Windowsin käyntiin, käynnistä SQLWIN.EXE ja ota työn alle SQLWindows-ohjelma NEWAPP.APP. Se on SAL-kielinen sovelluspohja, joka tulee toimituksen mukana. Näyttölläsi sen tulisi näyttää kuvan 1 mukaiselta.

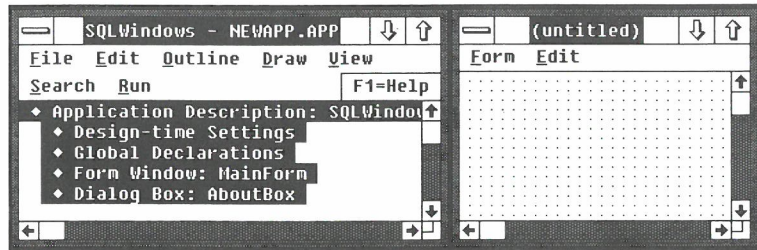
Ensimmäiseksi kannattaa tallentaa sovellus toiselle nimelle, ettet vahingossa tuhoa alkupeleistä NEWAPP.APP-tiedostoa. Tallentaminen käy vasemmanpuoleisen ikkunan File-valikosta valitsemalla Save As -vaihtoehto ja muuttamalla tiedoston nimeksi esimerkiksi KOE.APP.

Kuvassa 1 näkyy kaksi ikkunaa, joista oikeanpuoleinen (Form Window = Pääikkuna) tulee näkymään sovelluksesi käyttäjälle. Vasemmanpuoleinen ikkuna sisältää SAL-kielisen sovelluksen rungon. Sitä kutsutaan hieman hämäävästi nimellä Outline eli samalla nimellä kuin yhtä sen valikkovaihtoehtoa.

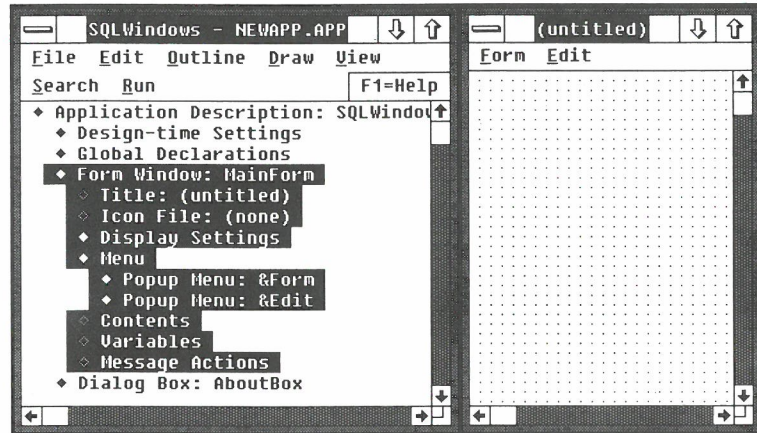
Näiden ikkunoiden sisällöt vastaavat aina toisiaan. Siis muutokset, joita teet Outlineen, näkyvät välittömästi Pääikkunassa ja päinvastoin. Yleinen työskentelytapa onkin valita Outlinen Draw-valikosta piirrettävän olion tyyppi (painonappi, radionappi, datakenttä jne.) ja sen jälkeen piirtää se Pääikkunaan. Tällöin SQLWindows automaattisesti lisää tarvittavat kohdat Outlineessa olevan SAL-ohjelman oikeaan kohtaan. Joudut täyttämään Outlineen vain ne osat, joita SQLWindows ei pysty itse päättämään.

Outline-ikkunassa näkyvä SAL-kielinen sovellusrunko näyttää hyvin lyhyeltä. Tosiasiasa siinä näkyvät vain sovelluksen pääotsikkorivit. Kunkin rivin alussa on kuvio, jota kutsutaan ohjekirjassa manteliksi. Jos tämä manteli on kokonainen (täytetty), se sisältää lisärivejä. Saat ne näkyviin valitsemalla ensin tutkittavan rivin ja sen jälkeen esimerkiksi valikosta Outline vaihtoehdon Expand One Level. Tietyn rivin voit valita näpättämällä hiiren vasenta näppäintä kyseisen rivin alussa, kun kohdistin näyttää oikealle kenolta nuolelta. Kokeile Outline-valikon eri vaihtoehtoja!

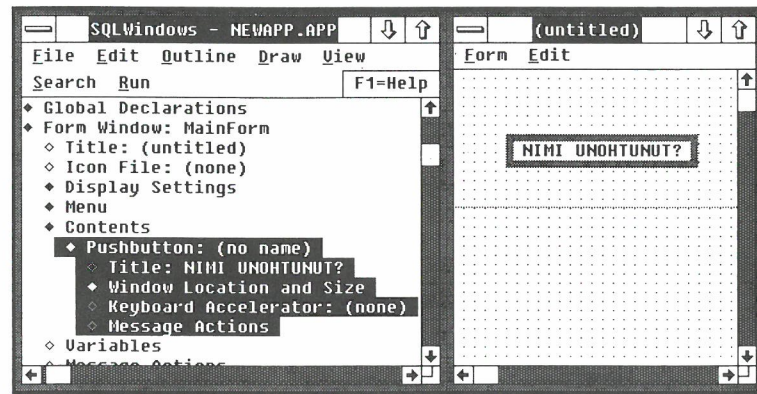
Kuvassa 2 näet Pääikkunaa vastaavan kohdan Outlineessa avattuna seuraavalle tasolle. Kun vertaat sitä Pääikkunaan, huomaa yhtäläisyydet kohdissa Title ja Menu. Kohdissa Display Settings ja Menu sekä Popup menu -riveillä on alakohtia, mutta kohdissa Title, Icon File, Contents, Variables ja Message Actions ei ole alakohtia, sillä niiden



Kuva 1: Sovelluspohja NEWAPP.APP.



Kuva 2: Pääikkunaa vastaava rivi Outline:ssa avattuna seuraavalle tasolle.



Kuva 3: Painonappi piirretty Pääikkunaan.

rivien alussa on pelkät tyhjä mantelinkuoret.

SQLWindows on erittäin tarkka sisennysten käytöstä. Esimerkiksi kuvassa 2 kaikki Form Window:iin kuuluvat asiat on sisennetty yhdellä pykälällä. Koko SAL-sovelluksella on erittäin tarkoin määritelty rakenne eli tietyt otsikkorivit tietyssä järjestyksessä ja niille vain tietyt, sallitut alakohdat. Esimerkiksi Edit-valikko sallii rivien kopioinnin paikasta toiseen vain, kun kohdeteksti on valittu kopioitaville riveille sopiva paikka tai sellaisen paikan otsikkorivi.

Olion piirtäminen

Piirtäminen aloitetaan valitsemalla Outline-ikkunan Draw-valikosta halutun olion tyyppi. Va-

litse vaikka painonappi (Push Button) ja vie kohdistin Pääikkunaan. Kun löydät painonapille sopivan paikan, näpätät hiiren vasenta näppäintä.

Tee painonapista vaikkapa kuvan 3 mallin mukainen. Kun teksti "untitled" on tummennettu, voit kirjoittaa painonappiin haluamasi tekstin ja lopettaa enterillä. Painonapin tekstiä voidaan muuttaa myöhemmin painamalla hiiren oikeaa näppäintä joko Pääikkunassa painonapin kohdalla tai Outline:ssa vastaavan Title-rivin alussa. Jos teksti ei mahdu painonappiin, voit venyttää napin kokoa kun kohdistin näyttää kaksipäiseltä nuolelta painonapin sopivalla sivulla tai kulmassa.

Pääikkunaan tulee helposti ylimääräisiä olioita. Ne saat poistetuksi valitsemalla poistet-

tavan olion ja painamalla delete-näppäintä. Valinnan voit suorittaa kohdistimella jos se näyttää olion keskellä nelipäiseltä nuolelta tai valitsemalla olion ensimmäisen rivin Outline-ikkunassa. Näitä ylimääräisiä olioita syntyy helposti, kun Pääikkuna on jäänyt alimmaiseksi, ja sitä yritetään saada esiin. Helpoin tapa saada Pääikkuna päällimmäiseksi on valita Outline:sta Form Window -rivi ja sen jälkeen View-valikosta Show Window.

Olioon liittyvä ohjelmanpätkä

Kuvassa 3 näet tilanteen, jossa painonappi on juuri piirretty. Outlinesta huomaat, että SQLWindows on automaattisesti kirjoittanut painonappiin liittyvää SAL-kielistä ohjelmaa niin pitkälle kuin se on pystynyt. Se on kirjoittanut kyseiseen olioon liittyvät otsakerivit ja tallentanut kohtaan "Window Location and Size" tiedot siitä, missä kohdassa näyttöä olio sijaitsee. Jotta ohjelmasi menisi käännöksen läpi virheettömästi, on sinun vielä annettava oliolle ohjelmamuuttujan nimi.

Muuttujanimeksi voit antaa esimerkiksi pbNimi. Muuttujien nimeämisessä kannattaa käyttää niin sanottua unkarilaista merkintätapaa eli muuttujan nimen alussa on pienillä kirjaimilla kuvattu muuttujan tyyppi (pb = Push Button) ja varsinainen nimiosa alkaa isolla kirjaimella. Tämä tapa auttaa saamaan ohjelmasta helposti luettavan. Huomaa, että SQLWindowsissa isoilla ja pienillä kirjaimilla on C-kielen tapaan eri merkitys!

Jotta tämä painonappi-olionime ei olisi pelkkä ikkunankoristus, tulisi siihen tietysti liittää jokin toimintaa kutsuva "Message Actions", mutta tarkastellaamme ensin hieman olioiden taustalla olevaa ajattelutapaa.

Olio-käsite

Oliopohjaisesta ohjelmoinnista (Object Oriented Programming) on monia käsityksiä. Seuraava yksinkertainen ajattelutapa tukee SQLWindows-ohjelmointia.

Olio-ajattelun pohjana voi pitää käsitettä "abstrakti tietotyyppi", joka tarkoittaa sellaista korkean tason tietotyyppiä, jolle on määritelty kaksi asiaa: ensin sen rakenne ja toisaalta siihen liittyvät toiminnot. Esimerkkinä voisi pitää vaikkapa pankkitiliä, jota vastaa tietty rakenne ja johon saadaan kohdistaa tiet-

tyjä toimenpiteitä (tililtä otto, tilille pano, tilin avaus, tilin lopetus jne.). Tämä ajatusmalli näkyy myös sellaisissa ohjelmointikielissä (Ada, C++ jne.), jotka käyttävät luokka-käsitettä (esim. jokin muuttuja kuuluu pankkitili-luokkaan).

Jos vertaamme tätä SAL-kielen ohjelman olioon, niin kuvassa 3 olevan painonappi-olion rakenne on kuvattu kohdissa Title, Window Location and Size ja Keyboard Accelerator ja siihen liittyvä toiminta tullaan kuvaamaan kohdassa Message Actions.

Juuri se ominaisuus, että olio ei ole hereillä, ellei se saa tiettyä sanomaa, jota se on varautunut käsittelemään, tekee tällaisesta abstraktista tietotyypistä olion.

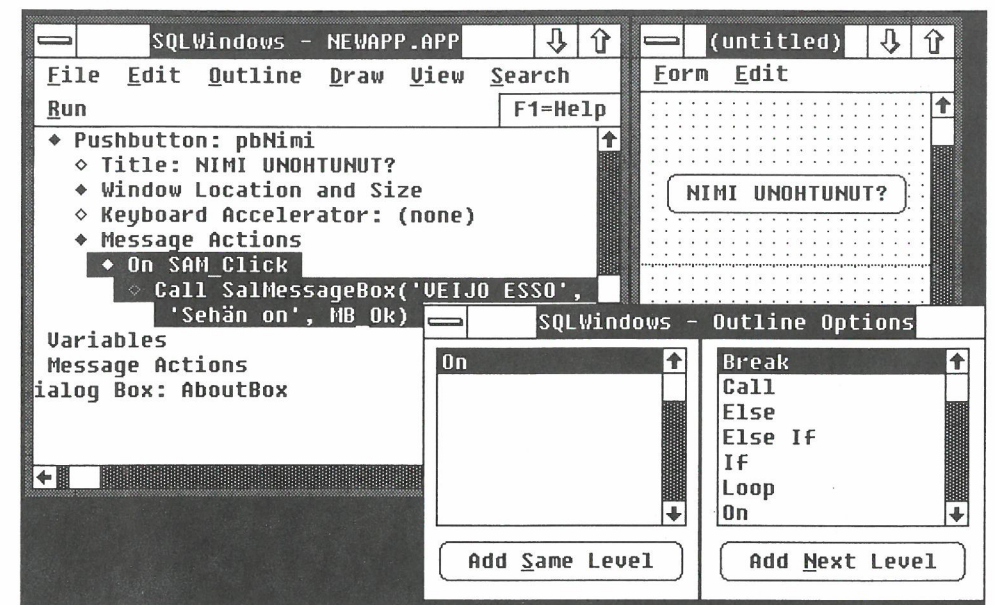
Oliot SAL-kielisessä ohjelmassa

Kuvassa 4 näet painonappi-olion, johon on lisätty reaktio tilanteeseen, jossa käyttäjä näpätää hiiren näppäintä painonapin kohdalla. Tämä on siis toiminto, joka halutaan painonapin Message Actions -kohtaan.

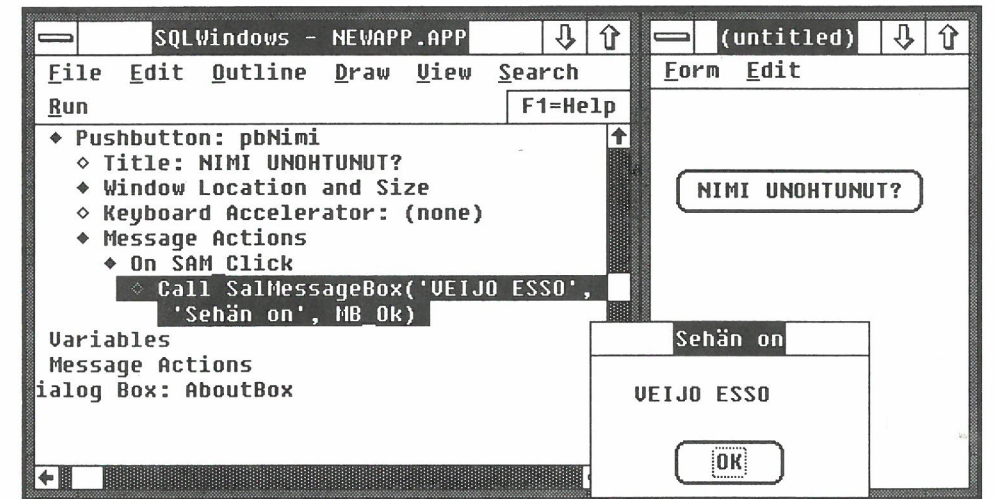
Ennenkuin alat kirjoittamaan toimintoa, valitse Outline valikosta View vaihtoehto Outline Options. Saat näkyviin pienen ikkunan, jota kannattaa aina käyttää, kun kirjoitetaan ohjelmaa. Se helpottaa turhien virheiden välttämistä huomattavasti.

Outline Options -ikkuna näkyy kuvan 4 oikeassa alareunassa. Se näyttää aina valittavissa olevat varatut sanat, joiden valikoima riippuu siitä, millainen rivi Outline:ssa on valittuna. Esimerkiksi kuvassa on valittuna rivi "On SAM-Click" eli näpätys-sanomaan reagoiminen. Tällöin Outline Options -ikkunasta voitaisiin valita samalle sisennystasolle toiseen sanomaan reagoiminen (On ...) tai SAM-Click-sanoman sisälle jokin oikeanpuoleisen list boxin varatuista sanoista (Break, Call, ...).

Jos sinulla on vielä tyhjä Message Actions -kohta, valitse se rivi tummennetuksi, jonka jälkeen näet Outline Options -ikkunassa ainoana vaihtoehtona On-lauseen eli tulevan sanomaan reagoimisen. Windows lähettää oliolle omista ja käyttäjien reaktioista sanomia, tässä tapauksessa sanoman SAM-Click. Koska painonappi on olio, se ei ole hereillä, ellei se saa sopivaa sanomaa. Siispä Message Actions -kohdan seuraavan sisennystason rivit ovat On-alkuisia, koska muita rivejä



Kuva 4: Olion reagoiminen sanomaan.



Kuva 5: Sovelluksen testaus.

ei koskaan kuitenkaan suoritetaisi.

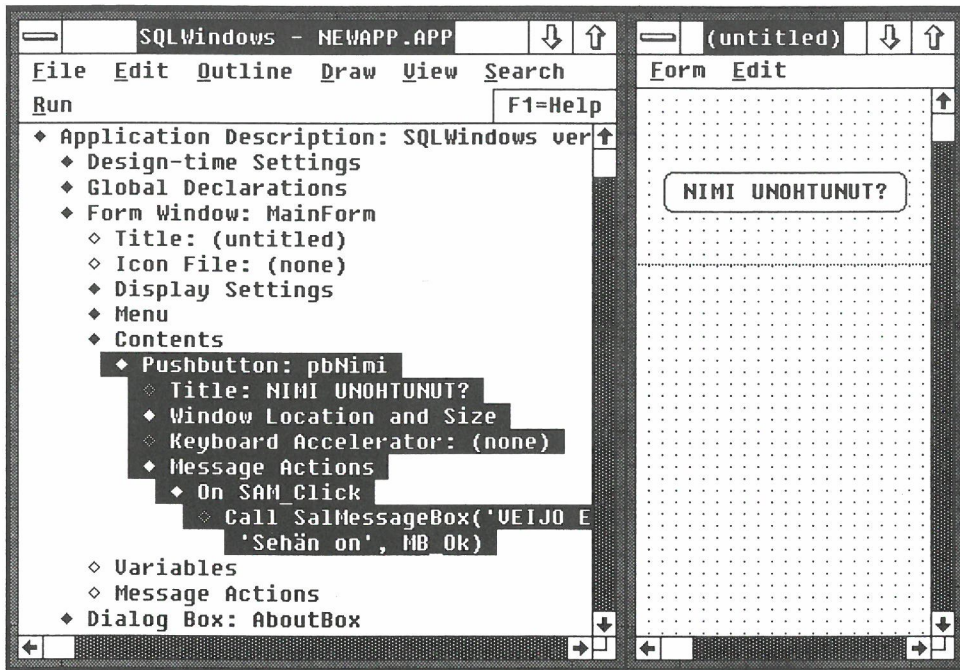
Olio-ajattelu näkyy myös siinä, ettei ohjelmakoodin mistään osasta hypätä painonapin Message Actions -kohtaan, vaan vasta käyttäjän toiminta voi käynnistää tietyn toiminnon. Näin päästään sovellukseen, jossa käyttäjä käskää sovellusta eikä päinvastoin.

Kun olet saanut On-lauseen kirjoitetuksi ja painanut enteriä jossain kohtaa riviä lähettääksesi rivin tulkittavaksi, näet Outline Options -ikkunassa kuvan 4 mukaiset vaihtoehdot. Jos On-rivilläsi oli virhe, tulkki huomauttaa siitä heti ja kysyy, säädettäänkö rivi virheestä huolimatta. Säätetyn virheellisen rivin alkuun ilmestyy huutomerkki, joka määrittelee sen kommenttiriviksi. Kun korjaat riviä, poista huutomerkki alusta siten ettei ri-

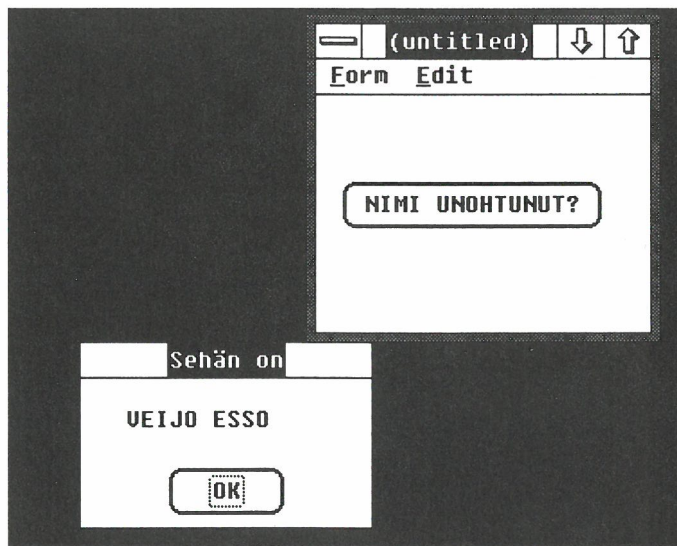
vin alkuun jää yhtään välilyöntiä!

Valitse sitten Outline Options -ikkunasta Call seuraavalle sisennystasolle. Kutsukaamme sillä valmista SAL-kielen funktiota (= aliohjelmaa) SalMessageBox, joka heittää näytölle pienen ikkunan, jonka sisällön määrittelemme kyseisen funktion parametreina. Ensimmäinen merkkijono-tyyppinen parametri näkyy ikkunan varsinaisena tekstinä, toinen ikkunan otsikkona ja kolmas kertoo, mitä painonappeja ikkunaan halutaan.

Vakiomerkkijonojen ympärille tarvittavat heittomerkit saat kätevimmin 102-näppäimistöltä *-merkin alta. Huomaa myös isojen ja pienten kirjainten erillaisuus SAL-kielessä! Riviä voit jatkaa CTRL-enter:llä. Tulostuvan Message Box -ikkunan muodon voit nähdä kuvassa 5.



Kuva 6: NEWAPP.APP-sovellus-pohjaan tekemämme muutokset.



Kuva 7: Käyttäjän näkemä "valmis" sovellus.

painamalla Delete-näppäintä.

Kun sovellus on kääntynyt, voit testata sen toimintaa esimerkiksi näpäyttämällä painonappia. Animaation vuoksi näet koko ajan, missä kohtaa Outlinea suoritus on menossa. Voit lisäksi kelata Outline-ohjelmaa, mutta jos haluat muuttaa sitä, sinun on tultava pois User Mode:sta. Se taas tapahtuu joko valitsemalla Pääikkunasta Form-valikosta Exit, system box:ista Close tai Outline:n Run-valikosta User Mode. Pääikkunassa näkyy taustapisteytys, jos ollaan Design-tilassa, eikä User Mode:ssa.

SAL-sovelluksen rakenne

Kuvassa 6 näet alkuperäiseen NEWAPP.APP-sovellukseen tekemämme muutokset tummennettuina riveinä. Siinä ovatkin vakio-osia kaikki muut, paitsi että alkuperäinen Form Window:n (Pääikkuna) Contents-kohta oli tyhjä. Kun piirsimme Pääikkunaan painonapin, SQLWindows vei automaattisesti siihen kuuluvat rivit Pääikkunan Contents-kohtaan.

Form Window:ssa on nyt sitten painonappi-olio, jolla on tietty rakenne ja toimintaosa, jossa on varauduttu sanoman SAM_Click vastaanottamiseen. Toisaalta kuvasta 6 näkyy, että myös Pääikkuna on olio, sillä siihenkin liittyy tietty rakenne (menu, contents, ...) ja toiminta (Message Actions, joka lukee toiseksi alimmaisella rivillä). Pääikkuna saattaisi reagoida esimerkiksi sanomaan SAM_Create (oliota ollaan juuri piirtämässä näytölle) ja ottaa tällöin vaikkapa tietokantayhteyden.

Harjoitellaksesi vielä sanomiin reagoimista, voit lisätä Pääikkunaan Datafieldin, joka reagoisi SAM_Create-sanomaan siten, että se saisi jonkin arvon Set-lauseella. Miksipä painonapin näpäytys ei voisi muuttaa sen arvoa?

Tällainen olio-pohjainen ajattelutapa tekee ohjelmista helposti muutettavia. Jos käyttäjä esimerkiksi haluaa jonkin toiminnan siirrettäväksi painonapista valikkoon, hän valitsee kaikki painonapin Message Actions -kohdan On SAM_Click -vaihtoehdon rivit (pitämällä kutakin riviä näpäyttäessään SHIFTiä alhaalla) ja sen jälkeen kopioi ne Menu-otsakkeen alla olevan valikon haluttuun Actions-kohtaan Edit-valikon avulla. ■

Remote Netbios yhdistää Tiimit

Remote Netbios -siltatuoteperehen avulla voidaan yhdistää kaksi Tiimi-työryhmäverkkoa etäislinjan ylitse. Samalla järjestelmä mahdollistaa etäistyöasemien kytkemisen Tiimiin mukaan.

Käytettävissä olevat linjavaihtoehtot ovat X.25 ja puhelinlinja, ja yhdistäminen voi tapahtua sekä MS-DOS- että MS OS/2 -palvelimien kautta. Yhdistettävät verkot voivat olla tyypiltään Token Ringejä, EtherNettejä tai muita NetBIOS-verkkoja. Lisätietoja: Olli Nurminen, puh. (90) 5671.

Eri tyyppiset verkot yhteen

Netbios Interconnect Program mahdollistaa eri tyyppisiin NetBIOS-verkkoratkaisuihin pohjautuvien Tiimi-työryhmäverkkojen yhdistämisen paikallisesti. Ohjelmiston avulla voidaan kytkeä esimerkiksi Token Ring EtherNetiin, Token Ring Token Ringiin tai vaikka NetNet Token Ringiin.

Kytkeytyvien verkkojen palvelut ovat molempien verkkojen työasemien käytettävissä. Ohjelmiston avulla työryhmäverkon saa kytkettyä myös runkoverkkoon, josta löytyy yhteisiä palveluja koko työryhmäverkkojoukolle. Netbios Interconnect Program asennetaan yli 1 megatavun työmuistilla varustettuun mikrotietokoneeseen, johon lisätään liitettävien verkkotyyppien mukaiset verkkosovittimet. Protokollavaihtoehdot ovat DCL/Netbeui ja XNS.

Lisätietoja: Olli Nurminen, puh. (90) 5671

SNA:lla yhteys keskuskoneeseen

Tiimin SNA-tuoteperehellä ja tietoliikennepalvelimella saadaan Tiimi-työryhmäverkko yhdistettyä jopa neljään eri tyyppiin isäntäkoneeseen yhtä aikaa. Yhteys voidaan toteuttaa myös etäisesti modeemiyhteyden kautta.

Ohjelmisto toimii kuin 3174/3274-määrittelyjen mukaiset työasemat, esimerkiksi S74. SNA-tuoteperehettä tai sen osia voidaan käyttää Tiimi Token Ringissä, EtherNetissä tai yksittäisissä työasemissa.

Ohjelmisto vaatii toimiakseen VGA-, EGA- tai CGA-yhteensovivan näytön sekä vähintään 640 kilotavua työmuistia. Lisätietoja: Olli Nurminen, puh. (90) 5671

Freelance Plus 3.0

Freelance Plussasta on tullut markkinoille versio 3.0. Se sisältää runsaasti uusia piirteitä kuten Portfolion, jolla voidaan helposti ryhmittää kuvia aiheittain sekä Screen Shown, jolla Portfoliolla ryhmiteltyjä kuvia voidaan esittää näytöllä.

Kuvat yhdistetään tietokantaan dynaamisten viittausten avulla. Uuteen versioon on lisätty myös uusia kuvioita, kirjasintyylejä, symboleja sekä mahdollisuus kolmiulotteiseen grafiikkaan. Kuten edeltäjänsä Freelance Plus 3.0 tukee useita eri tulostinvaihtoehtoja. Uutuutena mukava ovat myös PostScript-yhteensovivat laitteita.

Uudella versiolla voidaan tulostaa myös TIFF-tiedostoja. Freelance Plus 3.0 tukee 8087-, 80287- ja 80387-matematiikkaprosessoreita. Vapaata muistia tarvitaan ennen ohjelman käynnistämistä vähintään 508 kilotavua (ei tue extended- eikä expanded-muistia). Lisätietoja: Antoine Trux, puh. (90) 567 2136.

Windows 2.1

Microsoftin Windowsista on tullut uusi versio sekä 286- että 386-koneille. Sekä Windows/286 2.1 että Windows/386 2.1 on mahdollista saada ohjekirjat mukaan lukien myös suomenkielisenä.

Uusien versioiden asennusohjelmista on tehty aiempaa helppokäyttöisempiä siinä missä ylempää muistialuetta käyttävä ajuri HIMEM.SYS on uusittu kokonaan. Lisäksi kirjoitinohjaimien määrää on lisätty entisestään. Lisätietoja: Antoine Trux, puh. (90) 567 2136

Köyhän miehen laser

DeskJet on 300 pistettä tuumale tulostava mustesuihkukirjoitin, joka tuottaa laser-tasoisista jälkeä tavanomaisen kirjoittimen hinnalla. Kirjoitinta voidaan käyttää oman ajurin puuttuessa HP LaserJet -ajurilla, sillä se käyttää samaa HP PCL -sivunkuvauuskieltä kuin LaserJet. DeskJet on tarkoitettu henkilökohtaiseen PC-käyttöön, jossa tulostustarve on noin 50 sivua päivässä. Luonnostekstin tulostusnopeus on 240 merkkiä sekunnissa ja laatuja jälkeä kirjoitin tuottaa noin kaksi sivua minuutissa.

Yksi mustesäiliö riittää joko noin 1,2 miljoonaan luonnoslaatuiseen merkkiin tai arviolta puoleen miljoonaan laatuja jälkeä tulostettuun merkkiin (n. 500 sivuun).

Laitteesta löytyy vakiona rinnakkais- ja sarjaliitäntä. Lisäksi sitä voidaan laajentaa Epson FX-80 sovitinkortilla. Vakiotekstityyppi on Courier, mutta lisätekstityyppejä voidaan määritellä joko erillisillä kirjainaseteilla tai sovellusohjelmien omien tekstityyppien kautta...

Lisätietoja: Priitta Tasanko, puh. (90) 5671



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

Sovelluksen testaaminen

Nyt olisi tietysti hauska nähdä, miten hieno sovelluksemme toimii. Testaaminen käy kätevästi aktivoimalla ensin Slow Animation -tila Outline Run-valikosta, jolloin näemme, missä kohtaa Outlinea ollaan kullakin hetkellä menossa ja sitten samasta valikosta varsinainen User Mode -tila.

Ennen tätä kannattaa yleensä aina sulkea, ikonisoida tai ainakin pienentää kaikki turhat ikkunat (esim. Outline Options ja Pääikkuna), jotteivät ne turhaan kuormittaisi muistia käännöksen aikana. Jos sovellus on todella suuri, eikä expanded-muistia ole

käytettävissä tarpeeksi, sovelluksen voi tallentaa Make Runtime -muotoon ja testata sen toimivuutta käyttäjän tavoin.

Ennen käännöksen alkamista SQLWindows kysyy, halutaanko sovellus tallentaa. SQLWindowsin mielestä sovellustasi on muutettu, vaikka vain ikkunoiden koko olisi muuttunut.

Käännösvaihetta nopeuttaaksesi voit riisua Global Declarations -kohdan alakohdasta External Functions kaikki sellaiset funktiomäärittelyt pois, joita et usko sovellukseksi tarvitsevan. Ne saa tarpeen vaatiessa helposti kopioiduksi sinne takaisin. Funktioiden poistaminen tapahtuu samoin kuin olioidenkin eli valitsemalla poistettavan rivin ja

00003 HELSINKI

Sopimus 00380/166
VASTAUSLÄHETYS
PL 780/10
Softari
Nokia Data

Softari 1/89

Palvelukortti

☐ Haluan esitteitä seuraavista
Nokian palveluista ja tu...

☐ Pyydän, että minuun otetaan
yhteyttä.
puhelin: _____
Olen kiinnostunut _____

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

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

postimaksun
maksaa
Nokia

JOUKKOJULKAISU



8/00010

Amatsoneista androgyyneihin

Paljon on pelätty tietoyhteiskunnan nielevän tuttuja töitämme ja tuovan tilalle tukun outoja. Suurisuimmat tietoyhteiskunnan tallaat löytyvät riittämättömyydessään ja joustamattomuudessaan roikkuvista uroista, jotka ovat kahlinneet itsensä tarpeettomiin perinteisiin. Tietenkin, hehän työstä elävät ja meidät elättävät sillä aikaa, kun hameväki puuskuttaa työn ja kodin väliä.

Miehet tietävät mitä puolustavat, ja siksi onkin niin väärin, että tietoyhteiskunta suosii akkavaen elämäntapaa. Äidit ovat tottuneet pongaamaan sulavasti roolista toiseen ja rakentamaan yhteiskuntaa itselleen ja läheisilleen – tärkeässä mutta hiljaisessa tehtävässään pahemmin loppuun palamatta. Joka-akka osaa tehdä töitä kotona.

Tietoyhteiskunta suosii heikompaa astiaa, joka iän kaiken on opetellut tasapainoilemaan ihmissuhteissa ja jolla on halu elämän vaalimiseen. On todettu, että hameenkantajat, joilla on myös naisellisuutensa lisäksi himppu miehiä ominaisuuksia, pärjäävät työelämässä. Saman tutkimuksen mukaan menestyvät myös äitipojat, joista ei ole kovalla kädellä väännetty pehmeitä ominaisuuksia pois.

Tieto on lisääntyvä raaka-aine, jonka hallinta on valtaa. Tietoa on osattava käyttää luovasti. Naaraat ovat hakoja luovuutta peräänkuuluttavalla mainosalalla pikemminkin kuin perinteisillä aloilla. Mainonnan tekijän on osattava uusiutua, olla luova, joustava, sopeutuva ja kestävä.

Pehmous on entistä enemmän in. Ei lälärimäisyys, mutta joustavuus kuten sinitarra. Kestääkö menestynyt nainen moitteet kovuudestaan samaan aikaan, kun menestyneitä miehiä kehuutaan lujudesta? Vai onko elämä kymmenlapsisen perheen äitinä sittenkin antoisampi ja viisaampi valinta kuin bisnesmaailman pinnalliset pyörteet ja tekotärkeys?

Naisen menestymistä uralla uudistuvassakin yhteiskunnassa vaikeuttaa ainakin yksi asia: naiset osaavat miehiä paremmin katsoa elämää kokonaisuutena, ajattelevat muidenkin kannalta eivätkä tuijota elämää pelkän kapean uraputken läpi.

Työpaikalle tullutta tietokonetta käyttävät ensimmäiseksi naiset, mutta miksi he eivät käytä konetta oman asemansa parantamiseen? Onko osasyynä se, että uratädit viihtyvät parhaiten spagettiorganisaatioissa – tehtävien mukaan muodostetuissa projekteissa ja matriisirakenteissa, jotka ovat monen mielestä tulevaisuuden organisaatiomuotoja?

Akat hallitsevat luovia hässäköitä, mutta eivät näytä menestyvän jäykissä valtahierarkioissa, joissa viettävät aikaansa taantumukselliset ukot ja naikkoset, jotka pitävät itseään äijiä tyhmempinä. Jäykissä organisaatioissa menestyneen naisen on turha odottaa tukea kumpaakaan sukupuolta edustavalta työtoveriltaan.

Voisiko nainen tuoda bisnekseen ja tekniikkaan uusia näkökulmia? Naisia ei kiinnosta muusta elämästä irrallaan oleva tekniikka. Diplomi-insinööriksi nainen harvemmin haluaa, mutta kun tekniikkaa katsellaan laajemalta kantilta, alkaa nainenkin kiinnostua.

Teekkaripojan pakkomielteisiin kuuluu saada ala-arvoisen ympäristötentistä, kun taas opiskelemaan lähtevä tyttöjä kiinnostaa yhdyskunta- ja ympäristötekniikka tai arkkitehdin ura. He haluavat tekniikkaa, joka ottaa luonnon ja ihmiset huomioon.

Tietotekniikan hyväksikäyttö voisi olla hyvä askel kohti vuorovaikutusyhteiskuntaa, jossa henkiset arvot syrjäyttävät materiaaliset, teollisen yhteiskunnan arvot. Äitihahmot rakentavat vuorovaikutusyhteiskuntaa. Teollinen aika ei tukenut naisen menestymistä. Ei äitikään nähnyt materiaalisia arvoja tukemisen arvoisina.

Mutta tietoyhteiskunta ja sitä seuraava vuorovaikutusyhteiskunta on naiselliset (tai inhimilliset?) piirteet hyväksyvä maailma. Ihmiskunta palaa kotiin tuhlauksreissultaan.

Softaaja