

Title	Tietokone työvälineenä
Author(s)	Norros, Leena; Norros, Ilkka
Citation	Tiede ja edistys 1985: 3, s. 199 - 208
Date	1985
Rights	This article may be downloaded for personal use only

VTT
<http://www.vtt.fi>
P.O. box 1000
FI-02044 VTT
Finland

By using VTT Digital Open Access Repository you are bound by the following Terms & Conditions.

I have read and I understand the following statement:

This document is protected by copyright and other intellectual property rights, and duplication or sale of all or part of any of this document is not permitted, except duplication for research use or educational purposes in electronic or print form. You must obtain permission for any other use. Electronic or print copies may not be offered for sale.

Tietokone työvälineenä

Seppo Mustosen haastattelu

Leena ja Ilkka Norros

Yhteiskuntatieteellisen ja teknisen tutkimuksen eräs uusi yhteinen tutkimuskohde on automaatiotyö. Tämänkin lehden sivuilla on keskusteltu työn tulevaisuudesta, sen yleisen merkityksen mahdollisesta muutoksesta, työn ja vapaa-ajan suhteesta, työn ammattitaitovaatimusten kehityksestä jne.

Eräs tärkeä konkreettinen näkökulma automaatiotyöhön voisi olla sen tutkiminen keskeisen välineensä, tietokoneen kannalta. Sillä, minkälaiseksi tietokone välineenä ja käytäntönä (ihminen-kone-järjestelmänä) muovataan, on suuri vaikutus koko modernin kulttuurimme luonteeseen.

Monimutkaisiin teollisiin tuotantoprosesseihin kohdistuneiden tutkimusten perusteella voi esittää arvelun, että automaation käytön laatua ja tasoa määrää se, kuinka hyvin välineen konstruktiossa toteutunut toi-

mintatapa ja syntynyt tieto kyetään omaksumaan. Vahva työnjaollinen ero suunnittelun ja käytön välillä näyttäisi samalla olevan kyseenalaiseksi.

Tämän kysymyksenasettelun puitteissa haastattelimme Helsingin yliopiston tilastotieteen professoria Seppo Mustosta, joka tunnetaan ennen kaikkea SURVO-järjestelmän luojana ja edelleen kehittäjänä. SURVO on vuorovaikutteinen tietojenkäsittelysysteemi, jota käytetään tilastollisten analyysien suorittamiseen kymmenissä tutkimuslaitoksissa Suomessa ja ulkomailla. Se on toteutettu ensin pien- ja nytemmin myös mikro-tietokoneella. Kollegojen keskuudessa Mustonen tunnetaan tietokoneen käyttötapojen väsymättömänä uudistajana, keskustelijana sekä ohjelmoijavirtuosina.

L.N., I.N.

Mitä on käyttäjävälisyys?

T&E: Kehität SURVO-järjestelmää tutkimustyön apuvälineenä. Miten suhtaudut sellaisiin vaatimuksiin kuin "helppokäyttöisyys" ja "käyttäjävälisyys"?

S.M.: Tällä hetkellä systeemit kilpailevat helpoudella ja vaivattomuudella. Hyvän markkinaosuuden saamiseksi systeemiä tehdään mukaviksi aloitteleville käyttäjille. Tottunutkin käyttäjä on uuden systeemin suhteen aloittelija, ja hän saattaa arvostaa erilaisia piirteitä kuin tottuneena käyttäjänä. Aloittelijan kannalta avulias systeemi saattaa olla riesa vakiintuneelle käyttäjälle. Esim. jos käyttö on keskusteluvaa, tiedetään usein jo etukäteen, mitä järjestelmä tulee seuraavaksi kysymään. Joissakin uusissa ratkaisuissa onkin pyritty siihen, että systeemi mukautuisi käyttäjän tietojen tasoon ja tarjoaisi sopivia oikoteitä. Harvoin nämä ristiriitaiset vaatimukset on saatu yhdistetyksi.

Olen nykyisin vakaasti sitä mieltä, että itse asiassa monien ammattikäyttöön tar-

koitettujen systeemien kohdalla "helppokäyttöisyys" sanan tavanomaisessa mielessä on ihan älytön vaatimus. Niiden pitääkin tuntua vaikeilta aluksi. Vaativaan käyttöön tarkoitettu tietosysteemi on vähän samantapainen kuin musiikki-instrumentti. Käyttäjältäkin täytyy vaatia jotain, että se rupeaa soimaan. On aika helppo kuvitella, että jos viulu keksittäisiin näinä aikoina, niin veikkaisin että tietokoneasiantuntijat sanoisivat heti, että ei tästä soittimesta ole mihinkään, kun siinä on niin huono käyttäjälähtäjä. Kestäisi varmaan vuosia, ennen kuin ensimmäiset virtuoosit sitten näyttäisivät, miten se soi ja mitä kaikkea siitä saa irti. Kun viulu on siis todistettavasti jotain ihan muuta kuin jokin mekaaninen sähköurku . . .

On valitettavaa, että ihmiset hakevat nykyisin vain helppoja ratkaisuja. Ettei olisi työtä eikä mitään vaivaa, että tämä olisi ikäänkuin onnellinen olotila. Minusta tuntuu, että ihmisen pitäisi päinvastoin tällaisissa asioissa nauttia siitä, että hänellä on ongelmia että hän yrittää opetella uutta ja

joutuu ponnistelemaan. Mutta systeemin täytyy olla sellainen, että se palkitsee käyttäjän, motivoi hänen vaivannäkönsä. Siinä matkan varrella oppiessaan uusia asioita hän huomaa pystyvänsä tekemään yhtä ja toista, mitä heti alussa ei ole tullut ajatelleeksi. En siis yhtään ymmärrä, miksi kaikkien systeemien pitäisi olla aloittelijoille mieluisia.

Ohjelmistoratkaisuissa on usein automaatiikkaa väärissä paikoissa. On joninjoutavia asioita, joissa on hyvä, että systeemi on tietyllä lailla älykäs ja osaa auttaa käyttäjää. Hyvä esimerkki on se, että piirrettyyn kuvaan halutaan asteikkomerkinnät. Niiden tekeminen on sellainen asia, jonka systeemin voi kuvitella osaavan suorittaa. Kuitenkin useat ohjelmat ovat olleet tähän asti tässä suhteessa aika avuttomia, ja käyttäjän täytyy sanoa, millainen asteikko valitaan. Toisaalta joissakin hyvin suurissa asioissa ohjelmat saattavat olla hyvinkin automaattisia, esim. niin, että ne piirtämättä käyttäjän toiveista tekevät tietyn analyysin aina vakiotavalla. Automaatiikka on siis painottunut väärin.

Helppokäyttöisyys — siinä pitää olla tällainen *luonnollinen vastaavuus*: jos meillä on vaikea tehtävä, käyttämisen tulee olla tietyssä mielessä vaikeampaa kuin jos meillä on helppo asia. Alkuaikoina tilanne oli päinvastainen. Me voimme helposti tehdä hankalia monimuuttujalaskelmia, mutta oli vaikea harrastaa yksinkertaista taulukointia, tunnuslukujen laskemista tai ennen kaikkea piirtämistä. Kuvat ovat hyvin tärkeitä.

”Konttorikonemiehet”

T&E: Miten työsi tietokoneiden parissa alkoi?

S.M.: Minun kohdallani tämä lähti liikkeelle siitä, että Olli Lehto (nykyisin mate-

matiikan professori ja Helsingin yliopiston rehtori) oli Kaapelitehtaalla matemaatikko. Siellä alettiin 50-luvun lopulla huomata, että tietokoneala oli vahvasti tulossa. Siihen aikaan puhuttiin matemaattikkokoneista ja nähtiin, että matemaatikot olivat avainasemassa, niinkuin varmasti olivatkin. Insinöörien ohella he olivat ainoita, jotka saattoivat perehtyä uusiin mahdollisuuksiin.

Kaapelitehtaalle päätettiin perustaa elektroniikkaosasto, jossa Lehto oli tärkeänä puuhämiehenä. Hän värväsi ihmisiä matemaatiikan laitokselta. Muiden muassa Martti Tienari (nykyään HY:n tietojenkäsittelyopin professori) Reino Kurki-Suonio (Tampereen TKK:n professori) ja minä tulimme silloin Kaapelitehtaalle.

Olin matematiikan laitoksella assistenttina syksyllä 1960 ja siirryin 1961 alusta Kaapelitehtaalle. Professorina olen ollut vuodesta 1967 lähtien. Kaapelitehtaan aikoina tein yhteistyötä tilastotieteen professori Törnqvistin kanssa, joka johti postin pakettiliikenteen kehittämiseen kohdistunutta tutkimusta. Tästä käytännöllisestä aiheesta tein matemaattiseen verkkoteoriaan liittyvää työtä, josta syntyivät pro gradu -tutkielma, lisensiaatin työ sekä myös väitöskirjani. Väittelin v. 1964 Helsingin yliopistossa.

Väitöskirjani koski etäisyysjakaumia liikenneverkostoissa. Aihe oli hyvin konkreettinen, sellainen jota matemaatikot perinteisesti vierastivat. Muistan, kuinka menin erään vanhemman kollegan luo, jolta arvelin saavani hyviä vihjeitä topologisiin ongelmiini. Kuullessaan, mihin tarkoitukseen tietoja tarvittiin, hän ystävällisesti kieltäytyi keskustelemasta tästä aiheesta ja sanoi minulle, että tämä ei ole matematiikan tarkoitus. Puhtaan matematiikan harrastajien tyylihan on ollut hieman vähek-



G. Kamrowski

Gérôme Kamrowski: Panoramagrafi (1943).

syä käytäntöä. Meitä sanottiin konttorikonemiehiksi.

Teimme Nokialla aluksi erillisiä ohjelmia tilastollisia sovelluksia varten. Minulla oli vastuullani juuri tällaiset tilastolliset tehtävät. Teimme aika paljon töitä mm. yliopistolle, koska yliopiston laskentakeskus ei vielä silloin ollut päässyt alkuun.

Eivät sen aikaiset tietokoneet mitään ihmeellisiä olleet. Kuitenkin ensimmäinen kone, jonka kanssa olin tekemisissä, Elliott 803, oli aika mielenkiintoinen. Sitä pidettiin ensimmäisenä transistoroituna tietokoneena Suomessa. IBM 1620, joka tuli alkavaan yliopiston laskentakeskukseen ESKO:n jälkeen (ESKO oli suomalainen putkitekniikkaan perustuva kone), oli myös tällainen transistoroitu tietokone. Noiden kahden uuden tietokoneen myötä syntyikin kilpailua IBM:n ja Kaapelitehtaan välillä.

Huomasimme, että tietokoneen käyttö oli kannattavaa sellaisissa vaikeissa tehtävissä, joissa oli paljon laskemista. Yksinkertaisissa töissä, joita normaalisti tulee eteen tilastollista tutkimusta tehtäessä, tietokoneen hyötysuhde oli aika huono. Koneet olivat nimittäin aika vaatimattomia inputin ja outputin suhteen. Käytettävissä oli vain reikänauha. Ei siis ollut magneettikortteja, levy-yksiköistä puhumattakaan. Mistään päätteistä ei voinut myöskään puhua, vaan konetta ohjattiin näppäinpöydän kautta. Kun Elliottin tyyppillinen rekisteri oli 40 bittinä, niin pöydässä oli 40 nappia vastaamassa yhden rekisterin rakennetta.

Käyttäjävälisyydestäkään ei tietenkään voinut tässä vaiheessa vielä puhua. Tavoitteena oli pystyä laskemaan laskut. Vasta myöhemmin, mutta kuitenkin jo parin vuoden sisällä, alkoi kehittyä ajatuksia välleineistä näiden raskaiden laskentatehtävien helpottamiseksi.

Automaattinen monimuuttuja-analyysi

Monimuuttujamenetelmät olivat siihen aikaan muodissa monestakin syystä. Muiden muassa Yrjö Ahmavaaran ja Touko Markkasen synnyttämä innostus faktorianalyysiin ja transformaatioanalyysiin oli suuri. Tein aika paljon yhteistyötä juuri Touko Markkasen kanssa, joka oli Ahmavaaran ohella merkittävä henkilö psykometrisen tutkimuksen alueella.

Myöhemmin näiden menetelmien arvostus laski romahdusmaaisesti. Monimuuttujamenetelmien käyttö muuttui rutiinomaiseksi laskennan tultua helpoksi tietokoneohjelmien avulla. Syntyi valtava reaktio, joka sattui samaan aikaan, kun alettiin puhua positivistisen tieteen kritiikistä.

T&E: Monimuuttujamenetelmien käyttö helpottui siis siinä mielessä, että aineiston haltijan oli mahdollista saada ulos paljon tuloksia tietämättä juuri mitään menetelmänsä tilastollisesta luonteesta ja sisällöstä. Oliko helppous siis tavallaan näennäistä?

S.M.: Kyllä. Tämä on tärkeä kysymys. Tuohon aikaan esitettiin näkemyksiä, että tilastotiedettä ja tutkimusmenetelmiä ei tarvitse enää kunnolla opettaa korkeakouluissa, koska on olemassa nämä hyvät ohjelmat. Ne hoitavat kaiken automaattisesti. Monet käyttivät sellaista puhetaapa, että nämähän ovat jo lopullisesti ratkaistuja kysymyksiä. Vallitsi kaavamainen, sokea näkemys automaattista, jonne syötetään jokin sisään ja ulos tulee kauniita tuloksia.

Tarvittiin kuitenkin merkkien selittäjiä, ja näin laskentakeskukset ajautuivat aikaan liemen joutuessaan tarjoamaan palveluja tulosten tulkitsemisessa. Joillakin aloilla tutkimuksen teko saattoi pirstoutua yleisemminkin. Kaikki työn vaiheet voitiin delegoida: idea saatiin professorilta, materiaalin keräsivät tutkimusapulaiset, ja sitten läjä kannettiin laskentakeskukseen,

jossa joku asiantuntija mietti, mitä seille voisi tehdä. Ajot suoritettiin, minkä jälkeen tutkimuksen tekijä katseli itse tuloksia tarvitte tosin jonkun niitä selittämään. Ajojen tuloksetkin saatiin kirjoittaa valmiiksi auki. Näin tehtiin väitöskirja, jota tarkastettaessa sekä väittelijä että vastaavittaja olivat vaivautuneita, kun tultiin menetelmiä koskevaan kohtaan.

Tätä oli tietenkin oikeutettua vastustaa. Mutta kritiikki oli toisaalta varsin kohtuutonta. Puhtaasti teknisiä menetelmiä lyötiin, mistä seurasi se, ettei niitä enää käytetty järkevissäkään tilanteissa.

Nykyisen SURVO:n tyyppistä ratkaisua olemme voineet kehittää vuodesta 1975 lähtien. Minulla on ollut siihen tähtäviä haaveita ja ideoita jo Kaapelitehtaan ajoilta. Keskustelimme silloin, mikä olisi kiva tapa tehdä tällaisia asioita. Vaikka monet ratkaisut ovat olleet olemassa ideoina jo kauan, niitä ei ole ollut mahdollista toteuttaa silloisilla koneilla. Sitten 70-luvun puolivälissä alkoi ilmaantua laitteita, joissa oli mahdollisuudet tavoiteltuun interaktiiviseen käyttöön. Siinä mielessä sitten lähdettiin uudelleen liikkeelle täällä tilastotieteen laitoksella.

Editoriaalinen käyttötapa

T&E: Siitä on nyt kymmenisen vuotta. Mitä näkemyksesi ovat tänä aikana kehityneet?

S.M.: Aluksi käsitykseni hyvästä interaktiosta oli, että käyttö on keskustelevaa: systeemi esittää kysymyksiä ja tarjoaa valmiita vastausvaihtoehtoja, käyttäjä valitsee jotain, ja päästään eteenpäin seuraavaan vaiheeseen. Mutta tällainen työskentely saattaa osoittautua turhauttavaksi juuri totuneen käyttäjän kohdalla. Ei se, että se siinä ystävällisesti keskustele... se on

ihan kuin piinaava keskustelukumppani, joka haluaa jaaritella yksityiskohtaisesti, selittää kaikki asiat joka kerran. Täytyy siis olla oikoteitä ja suurempia tapoja. SURVO:n kehittämisessä tuli siis sellainen vaihe, jossa huomasin, että tämä ei ole välttämättä ainoa oikea tapa hoitaa näitä juttuja. Kun pohditaan vuorovaikutteisen käytön muotoja, pitäisi lähteä liikkeelle siitä, minkä tyyppisestä työskentelystä ihmiset pitävät. Kaikki eivät pidä samanlaisesta; on oltava vaihtoehtoja. Tämä, mihin minä olen päätenyt ja jonka parissa olen työskennellyt ainakin viimeiset viisi vuotta, on käyttötapa, jota sanon *editoriaaliseksi*.

Sen idea syntyi itse asiassa joululomalla 1979—80, kun kehitelimme poikani kanssa nuottien puhtaaksikirjoitusta tietokoneella. Mieleeni juolahti, että siinä käytetyllä tekniikalla voisi olla yleisempään merkitystä. Muuten, tässä näkyy, miten toisilleen vieraat asiat yhdessä voivat tuottaa uutta. Siksi pitää tehdä useampia asioita, niiden suhde saattaa olla kiinnostava.

Editoriaalinen käyttötapa perustuu puhtaaseen tekstinkäsittelyajatteluun. Vuorovaikutteisuus ilmenee siinä muodossa, että käyttäjä kirjoittaa vapaasti tekstiä kuvaputkelle. Kuvaputkea voidaan pitää liikutelvana ikkunana sitä itseään paljon laajempaan työalueeseen. Se on siis ihan vapaata aluetta, sellainen hiekkalaatikko, jonne voi laittaa mitä tahansa, lähinnä nyt tekstiä ja taulukoita, mutta ei pelkästään tätä käsiteltävää tietoa. Myös kaikki ne ohjeet, joilla vaikutetaan siihen mitä siellä tehdään, sijoitetaan samaan tilaan.

Nythän on muotia, että on moni-ikkunaisia näyttöjä, joilla voi työstää useita asioita samanaikaisesti. Minusta se on jotenkin vähän levotonta; siinä on taas sellaista alkuinnostusta. Olen nähnyt edulliseksi, että kaikki erilaiset asiat, siis käsiteltävät tie-

dot, niihin vaikuttavat operaatiot ja tulokset, jos halutaan, ovat samassa tilassa. Minulle riittää yleensä kaksi ikkunaa.

Yleensä vuorovaikutteiset systeemit on rakennettu sellaisiksi, että ellei käyttö ole keskustelevaa, se on komentotyyppistä: esimerkiksi niin, että vaihtoehtoiset komennot näkyvät alarivillä, josta niitä voi osoittaa kursorilla. Tässä on se ongelma, että kun komennot ovat omassa tilassaan, niitä on aika vaikea palauttaa, täytyy uudelleen valita samalla tavalla. Sen sijaan tälle minun idealleni on leimallista, että komennot ovat kaiken muun tiedon seassa. Ensimmäisellä kerralla ne joutuu jopa kirjoittamaan sinne, mutta koska ne säilyvät tekstin joukossa, ellei käyttäjä niitä pyyhi, niitä voi käyttää aina uudelleen ja uudelleen.

Vuorovaikutteisuus ilmenee editoriaalisessa käyttötavassa sellaisena roolijakona, että käyttäjä kirjoittaa ja tekee työtä niin pitkälle kuin haluaa, mutta jossain vaiheessa hän huomaa tai tietää, että tässä systeemi osaa auttaa, esim. laskea jonkin asian, piirtää jonkin kuvan tai muokata tekstiä tai taulukoita sopivampaan muotoon. Silloin voi pyytää sitä, ja se tapahtuu siten, että kirjoitetaan jotain tähän liittyvää siihen samaan tilaan ja *aktivoidaan* se. Tämä aktiivivaihe on tärkeä. Osoitamme kohdistimella paikan mistä olemme kiinnostuneita, ja sitten painamme jotain nappia, joka herättää systeemin tutkimaan, mitä siinä on. Systeemi puuttuu peliin käyttäjän tarvitessa, tekee halutun toimenpiteen ja palaa sitten jälleen normaalitilaan. Aktiivointi ilmenee minulla siten, että aktivoitu kohta näkyy kirkkaammalla, jolloin käyttäjä huomaa systeemin tekevän jotain. Normaalityöntöön palattaessa valoilmio katoaa ja käyttäjä tietää voivansa jatkaa.

Minun mielestäni käyttäjän ja systeemin repliikit vuorottevat tässä hyvin joustavas-

ti, ja kumpikin voi toivon mukaan käyttää parhaita puoliaan. On tärkeä huomata, että vastuu on aina käyttäjällä. Systeemi ei oma-aloitteisesti tee mitään eikä siis painosta käyttäjänsä. Usein keskustelevat systeemit ovat sellaisia, että ne ottavat helposti aloiteoikeuden itselleen.

SURVO:n uudessa käyttäjä-oppaassa on esimerkiksi esitetty koronkorkolaskelma. Se ei todellakaan ole mikään ohjelma, vaan vapaata tarinaa, jossa on joukossa kaavoja ja muuta tällaista. Jos nyt haluan tietää, mikä on jokin kertynyt pääoma, voin vapaasti korjata näitä lähtöarvoja, asetella niitä toisiin paikkoihinkin tietysti. Kun nyt kuljetan kursorin tuohon ja suoritan aktiivoinnin, kysyn k_n :ää, niin se etsii missä k_n :stä kerrotaan jotain, löytää jostain kaavan, joka riippuu k :sta, hakee k :ta, sijoittaa sen arvon jne.

T&E: Voit siis kirjoittaa tekstin sekaan *eräänlaisia ohjelmia ja suorittaa niitä.*

S.M.: Sanon näitä laskentakaavioiksi, tai työkaavioiksi. Ne ovat asetelmia, joissa työtilaan kirjoitetaan aika vapaamuotoisesti niitä asioita joita halutaan käsitellä. Kaavat täytyy tietysti kirjoittaa oikein, mutta sidesanat saavat olla mitä tahansa. Onko tällainen sitten ohjelmointia? Kyllä kai se uudemman tulkinnan mukaan on — sitä sanotaan objektorientoituneeksi ohjelmointitavaksi. Lähtökohtana on sovellus, ja sen kannalta yritetään miettiä, mikä on kieli jota käytetään. Monesti ajatellaan, että pitäisi rakentaa jokin kieli jotain sovelusta varten, mutta minusta pitäisi enemmän pyrkiä siihen, että voidaan käyttää aika lailla normaalia kieltä. Tässä näin tehdään, vaikka siinä on nyt paljon samaa kuin monissa tekoälysystemeissä, että on aika näennäistä, mitä systeemi asiasta ymmärtää. Se ei todellisuudessa lue kuin nämä kaavakohdat, näiden "=" -merkkien suhteen se on utelias.

Ohjelmoinnista on puhuttu uutena lukutaiti kirjoitustaitona, joka kaikkien pitäisi omaksua. En pidä ohjelmointitaitoa sanan nykyisessä merkityksessä yleisesti kovin tärkeänä. Monet nykyiset kielet tulevat varmasti katoamaan, ja tilalle tulee tietysti juuri näitä käyttäjäläheisempiä tapoja toimia, siis määritellä tehtäviä ja hoitaa niitä. Kai ohjelmoinnille on ominaista se, että siinä tehdään työtä, joka toimii sitten mallina myöhemmille soveltamisille. Minusta objektorientoituneille lähestymistavoille on ominaista, että niissä tehdään malliksi asia kerran kunnolla, ja sen jälkeen malli on käytettävissä itselle ja muille. Lisäksi mallia on helppo muuttaa ja kehittää. Jossain tällaisessa laajemmassa muodossa ohjelmointitaidon hankkiminen on varmasti hyödyllistä.

T&E: Mitä ymmärrät systeemin älykkyydel-
lä?

S.M.: Termi "tekoäly" on sinänsä ihan kaamea — minusta systeemi on älykäs tai ei, "teko"-liite on tavallaan tarpeeton. Loppujen lopuksihan se on ihmisen älyä, jota yritetään ympätä siihen systeemiin. SURVO:n tapaisessa ratkaisussa älykkyytenä voidaan pitää sitä, että systeemi osaa reagoida järkevästi erilaisissa tilanteissa ja käyttäjä tuntee voivansa tehdä eri toimenpiteet miellyttävästi. Jos käyttäjällä syntyy sellainen vaikutelma, että ei-ö tätä nyt voi tehdä yksinkertaisemmin, se on osoitus, että systeemissä on jotain heikkoutta. Omissa töissäni joudun usein keskittämään työskentelyn tehdäkseen pienen muutoksen itse systeemiin. Kohtaan jatkuvasti asioita, joita pidän heikkouksina, ja yritän sitten parantaa niitä.

Sellainen termi kuin "asiantuntijajärjestelmä" halutaan nykyään rajoittaa koskemaan tietyillä "tekoälyyn" liittyvillä tavoilla tehtyjä järjestelmiä. SURVO:n tapai-

nen systeemi on asiantuntijajärjestelmä jossakin toisessa mielessä.

T&E: Onko SURVO:n käyttötavassa *editoriaalisuuden ohella muita erikoispiirteitä?*

S.M.: On esimerkiksi toimintatapa, jota kutsun *tutoriaaliseksi moodiksi*. Suorittamalla tietty etukäteistoimenpide voidaan käyttäjän kaikki näppäinpainallukset panna muistiin. Näin purkitettu työ voidaan koska tahansa toistaa, ja sitä voidaan myös muuttella ja parannella. Tutoria käytettäessä näyttää kuin aavekäyttäjä kirjoitaisi tekstiä ja käyttäisi järjestelmää. Se on ikäänkuin dynaamisessa muodossa oleva oppikirja. Kun tavallisessa kirjassa tai ohjetekstissä voidaan näyttää vain tietyt vaiheet, tässä on koko historia, järjestys mukana. Näin voidaan tuottaa ohjeita systeemin käytöstä, mutta myös esim. joidenkin tilastollisten menetelmien soveltamisesta.

Vuorovaikutteinen järjestelmä edellyttää vuorovaikutteista suunnittelua

T&E: *Kehität tietokonejärjestelmää työväliteiksi muille. Millaista tämä kehitystyö on?*

S.M.: Vasta ihan viime vuosina on havaittu, miten tärkeää on nähdä, *miten* koneiden kanssa työskennellään. Jotain tutkimustietoa on ollut asiasta jo pitkään. Tiedetään esimerkiksi, että näyttöruudulla ei saa olla liikaa asiaa, mikäli haluaa saada käyttäjän tajuamaan sanoman. Tietyt pelisäännöt ovat siis olemassa, ja ne tunnetankin jo osittain, mutta mikään itse työstä irrallaan oleva tutkimus ei pysty selvittämään niitä. Esimerkiksi hyviksi havaittuja vuorovaikutteisia järjestelmiä on vaikea luoda perinteiseen tapaan ylhäältä päin niiden menettämättä joustavia piirteitä. Ennenhän tehtiin näin, ja tuloksena oli varsin jäykkiä toimintatapoja.

Käyttäjäpoloiset mukautuivat kyllä mihin vain ja oppivat pitämään suunnittelijoiden ratkaisuja jopa hyvinä. Se on kai tärkeää, kun joutuu opiskelemaan tällaisen tietosysteemin hallintaa päiviä, viikkoja tai kuukausia. Ihmiset koettavat kokea työskentelyn myönteisenä. He ilmeisesti kehittävät työstä malleja, jotka vahvistavat vallitsevan toimintatavan oikeaksi. Harva suhtautuu kriittisesti tekemäänsä, vaan koettaa mukautua. Tämän takia emme saa selville, miten järjestelmää tulisi kehittää kyselemällä vain käyttäjältä, ovatko valmis järjestelmä ja opittu käyttötapa hyviä. Käyttäjän on sen vuoksi itse rakennettava järjestelmä. Toisin sanoen järjestelmän rakentamistavan on oltava sellainen, että käyttäjä voi osallistua järjestelmän luomiseen.

Vuorovaikutteisuuden tulee siis ilmetä monella tasolla. Se ei ole vain sitä, että järjestelmä on itse vuorovaikutteinen käyttäjän kanssa, vaan että systeemin rakentamisenkin on sitä, kahdellakin tavalla: koneen ja systeemin laatijan välillä sekä sen laatijan ja potentiaalisen käyttäjän välillä, joka ehkä työskentelee jo jonkin koeverSION avulla.

Näin on tapahtunut SURVO:n kehitystyö. Vaikka olenkin itse tehnyt siitä suurimman osan, työssä on ollut mukana paljon sitä aktiivisesti käyttäviä ihmisiä. He ovat kritisoineet ehdotettuja ratkaisuja. Olen itsepäinen enkä halua muuttaa mitään, mutta ilmeisesti osa kritiikistä jää alitajuntaan. Ehdotukset saattavat sitten pulpahtaa esiin sieltä, jopa niin, että joskus koen muiden ehdottamat ratkaisut myöhemmin omina ajatuksinani.

Työssä ei saa antaa periksi, tekijällä täytyy olla itsetuntoa. Ei sekään ole hyvä, että myöntyy kaikkeen, mitä toiset esittävät, sillä silloin yleisnäkemyks ja yhtenäisyys järjestelmän rakenteessa kärsivät.

Yksi käyttäjä käyttää tyypillisesti vain 5 % laajan systeemin kokonaistoiminnoista. Mutta jos hän joskus huomaa tarvitsevan- sa niitä muitakin asioita, niin niiden pitäisi olla tavallaan tuttuja, vaikkei hän olekaan niitä ennen käyttänyt. Sen maiseman, joka systeemistä ihan visuaalisessa mielessäkin avautuu, tulisi olla mahdollisimman yhdenmukainen. Tästä syystä systeemin rakentaminen ei voi olla kovin "demokraattista". Jollakulla täytyy olla päätösvalta. En itse usko ollenkaan, että systeemiä voi tehdä kollektiivisesti siinä mielessä, että kaikki olisivat koko ajan päättämässä, mihin päin sitä viedään. Kun erilaisia näkemyksiä ilmenee, vastuun ja päätösvallan tulee olla yhdellä. Yhden tahdon mukaan tehtynä työ on parempaa kuin jos hyvin erilaiset tahdot vaikuttavat yhtäaikaan.

T&E: *Meille on muodostunut käsitys, että monimutkaisten tuotannollisten järjestelmien automaation suunnittelussa pätevät samanlaiset periaatteet. Työvälineen kehittämisprosessi on samanlainen. Ehkä vain vaikeammin toteutettavissa näin.*

S.M.: Kyllä. Tämä järjestelmä on suhteellisen pieni, mutta kai analogioita täytyy olla.

Ennenhän, ja ehkä paljon vieläkin, kaikessa ohjelmoinnin ja systeemisuunnittelun opetuksessa oli vallalla näkemys, että tämä toiminta on hyvin johdonmukaista työtä, jossa täytyy tehdä eri asteisia kulukaavioita ja systeemikaavioita, jotka testataan huolellisesti kirjoituspöydällä. Vasta tämän jälkeen mennään koneelle ja ruvetaan tekemään lopullista koodia, josta systeemi sitten syntyy.

Tämä on mielestäni kammottava näkemys, ja uudet tuulet puhaltavatkin jo yleisesti. SURVO:n teon yhteydessä olen nähnyt jo pitkään, että suunnittelun pitää tapahtua vaiheittain. Tietysti pitää olla jokin arkkitehtoninen perussuunnitelma, selvät

ideat ja tavoitteet. Mutta rakentamisen pitää koko ajan hengittää. Ratkaisuja pitää olla valmis koko ajan muuttamaan. Suunnittelussa, jossa kaikki piirustukset ovat etukäteen valmiina ja jossa ne viedään läpi sellaisenaan, jo niihin pannun suuren työmäärän velvoittamina, ei kellään ole ottaa lopussa todeta, ettei järjestelmästä tulutkaan sellaista kuin olisi pitänyt.

Minä en ainakaan ole niin viisas, että pystyisin etukäteen näkemään ongelmat, jotka tulevat esiin systeemin toteutusvaiheessa. Siksi on omaksuttava lyhytjännitteisempi linja. Täytyy olla jonkinlaista vaistoa. Sitä on hyvin vaikea selittää. Ainakin tässä vaiheessa systeemin rakentaminen on hyvin epätieteellistä. Spontaanit ja vaistonvaraiset toiminnot ovat siinä hyvin keskeisessä asemassa. Aikaisempi kokemus kasautuu taakse ja sitä pitää tietysti käyttää hyväksi. Mutta ratkaisuja on hyvin vaikea perustella tai selittää muille.

On pakko käyttää intuitiota. Tätä toimintaa voi ehkä verrata säveltämiseen, jossa kontrapunktinen ajattelu on keskeistä. Säveltäjähän ei yleensä tyydy siihen, että on yksi melodia ja vähän jotain säestystä. Vähänkin kehittyneemmässä säveltämisessä on lisäksi mukana kaikenlaisia vastaäänä. Tarvitaan kontrapunktista ajattelua, jotta asiat saadaan sopimaan yhteen. Tätä voi käyttää hyväksi. Eiväthän säveltäjätäkään, minun tietääkseni, ole yleensä määrätietoisia, vaan toimivat melko lailla vaistonsa varassa. Kuitenkin he saavat aikaan harmonisia kokonaisuuksia. Tämä sama pätee systeemisuunnittelun yhteydessä.

Se näkyy mm. tilanteessa, jossa tehdään valintoja usean vaihtoehdon vallitessa. Minulle on syntynyt sellainen käsitys, että pitää valita sellainen vaihtoehto, jolla on heti *muutakin* käyttöä. Tämä tarkoittaa sitä, että systeemi on sellainen kokonaisuus, jolla on ydin ja joka laajenee siitä ulospäin.

Kun laajennukset tukevat toisiaan, ei synny sellaisia ulokkeita, joilla ei olisi tukeaa keskuksesta, vaan struktuuri kasvaa verkkomaisesti. Laajeneminen tapahtuu siis perustellusti.

T&E: *Tarkoitatko siis, että yksittäiset ratkaisut ovat samalla jotenkin yleisiä?*

S.M.: Niin. Tässä pätee sama kuin matematiikassa, että yleistämällä tarkastelutapa saa usein aikaan tehokkaampia ratkaisuja. Tässä kohdataan perinteinen erotte- lu. Jo 20 vuotta sitten keskustelimme siitä, miten eri ihmiset ohjelmoivat. Insinööri-llä oli ominaista tehdä ratkaisunsa juuri tarkoitettuun asiaan. Matemaatikko taas yritti katsoa, mikä on mahdollisesti asian yläpuolella oleva struktuuri. Hänen ratkaisunsa oli yleisempi. Se oli ehkä tehottomampi, mutta saattoi osoittautua hedelmälliseksi jatkoa ajatellen. (Täytyy kyllä sanoa, että jos insinööriä joskus vaivaa pragmatismi, niin matemaatikkoa puolestaan sortuminen haihatteluun.)

Yleensä se, joka kysyy vastausta systeemisuunnittelun ongelmiin, esittää kysymyksensä liian rajoittuneesti. Vaikka haluttuun kohteeseen tehtäisiin ohjelma mittailaustyönä, tilaaja on kuitenkin itse ensimmäisenä ilmoittamassa tyytymättömyytensä siihen. Hän huomaa ohjelmaa käyttäessään, ettei se ole sitä mitä hän pyysi. Mukana olisi pitänyt olla sitä ja tätä. Tämä on juuri yhteydessä siihen, -että systeemi pitäisi tehdä käyttämällä sitä.

T&E: *Voitko sanoa vielä lopuksi, kuinka yleinen edustamasi ajattelutapa on systeemisuunnittelun alalla?*

S.M.: Logiikkani on aika yksinkertainen, mutta ei systeemejä yleensä ole tehty tällaiseksi tai näin. Minä olen tätä puhunut ja esitellyt vuosikaupalla, mutta en ole näissä yhteyksissä huomannut muiden tekevän asiaa ihan samalla tavalla. Mutta en voi tietää. Ongelma on siinä, että ehdo-

tuksia on niin valtavasti, että jos yrittää tutustua niihin, ei itse ehdi tehdä yhtään mitään.

Tietysti pitää pysyä kuulolla alan tapahtumista, mutta luotan enemmän siihen, että ne, jotka ovat yhteistyössä kanssani, kertovat, mitä ovat huomanneet. Oman alani kongresseissa on yleensä käynyt niin, että istun koneeni kanssa ja esittelen systeemiäni ja ihmiset käyvät katsomassa ja kes-

kustelemassa. Minulta jäävät silloin muiden esitykset kuulematta. Yleensä sanon muille suomalaisille, että kertovat, jos jokin mielenkiintoista ilmenee, ja menen sitten sopivan välin tullen katsomaan.

Olen valinnut tämän toimintatavan ihan tarkoituksellisesti. Katson, että joka tilanteessa on parempi yrittää itse tehdä, vaikka tietääkin, että ennen pitkää joku toinen tekee parempia ratkaisuja.

Numeromme kirjoittajat:

Satu Apo, FL, ass., Kulttuurien tutkimuksen laitos, Turun yliopisto.

Raimo Blom, HTT, apul.prof., Sosiologian ja sosiaalipsykologian laitos, Tampereen yliopisto.

Niklas Bruun, OTT, tf. professor, Svenska handelshögskolan.

Karl Johan Donner, FT, Turun yliopisto.

Antti Eskola, YT, prof., Sosiologian ja sosiaalipsykologian laitos, Tampereen yliopisto.

Olli Haapala, dosentti, Turun yliopisto.

Yrjö Haila, FT, Eläintieteen laitos, Helsingin yliopisto.

Antti Hautamäki, FL, tutkija, Suomen Akatemia.

Erkki Kronholm, M.Sci., psykofysiologi, Kansaneläkelaitoksen kuntoutus-tutkimuskeskus.

Heikki Lehtonen, YT, vt. prof., Sosiaalipolitiikan laitos, Tampereen yliopisto.

Pentti Määttä, VTK, FK, Helsingin yliopisto.

Ilkka Norros, FL, ass., Matematiikan laitos, Helsingin yliopisto.

Leena Norros, Dr. rer.nat., erikoistutkija, Sähkötekniikan laboratorio, VTT.

Olli Perheentupa, VTK, yhteiskuntateorian harrastaja.

Hannu Perho, FT, vt. prof., Joensuun yliopisto.

J.P. Roos, apul.prof., Sosiaalipolitiikan laitos, Helsingin yliopisto.

Thomas Wallgren, FK, Vaasan Prosenttiliikkeen toimintaryhmä.