

Systeemityön historia Suomessa

Systeemityö käsittää tietojärjestelmien rakentamisen eri vaiheet ja niissä käytetyt menetelmät, työkalut ja apuvälineet. Systeemityö kattaa koko tietojärjestelmien elinkaaren.

1960-luvulla pääpaino oli käyttää tietotekniikkaa hyväksi kustannussäästöjä tavoitellen. Ajalle ominaista oli toimia kapealla alueella. Järjestelmät olivat eräajojärjestelmiä ja vain operaattorit käyttivät tietokonetta.

1970-luvulla painopistealueeksi alkoivat nousta tietojärjestelmien kehityshankkeet ja projektityö alkoi saada laajalti sijaa. Käyttäjäkunta laajeni käsittämään myös muita asiantuntijoita kuin operaattoreita ja systeemityön- ja tietotyönammattilaisten määrä alkoi kasvaa.

1980-luvulla tietotekniikka alkoi vaikuttaa liiketoimintaan ja sen avulla koettiin saatavan kilpailuetua. Käyttäjäkunta laajeni laite- ja ohjelmistotoimittajiin sekä asiakkaisiin. Tavoitteena yrityksissä oli tukea liiketoimintaa ja integroida erilaisia järjestelmiä yhteen tiedon hallinnan parantamiseksi.

1990-luvulla perustettiin tietohallinto-osastoja tukemaan olemassa olevien järjestelmien ja tulevien järjestelmien kehittämistä yhdessä ohjelmistotalojen kanssa. Monet yritykset myös luopuivat omasta atk-osastostaan kustannus- ja hakiessaan. Syynä oli myös halu saada yhä paremmat tietotekniikkaresurssit käyttöön ulkopuolisten ohjelmistotoimittajien avulla.

2000-luvulla mielenkiinto on kohdistunut liiketoimintaprosessien uudistamiseen, järjestelmien integrointiin toisiinsa ja yrityksen ulkopuolisten järjestelmien käyttöön. Yritysten on ollut pakko verkottua ja ne ovat käyttäneet hyväkseen omaa tietotaitoaan ja osaamistaan yhdistyneenä uudenaikaiseen informaatioteknologiaan. Koko yritys on valjastettu oppimaan uutta ja poisoppimaan vanhaa.

Kehitys 1960-luvulta saakka on muokannut ja asettanut yhä suurempia vaatimuksia tietojärjestelmien kehittämiseksi ja sen mukana systeemityön tekemiseksi ja sen tekijöille. Tietojärjestelmien hankinnat ovat pitkälti investointiprojekteja, jotka vaativat paljon rahaa, aikaa, ihmisresursseja, osaamista ja teknologian tuntemusta. Ammattilaisten on pystyttävä yhä nopeammassa tahdissa oppimaan ja omaksumaan uusia tietoja ja taitoja, mutta myös poisoppimaan entisiä.

Tietojärjestelmien kehittämisestä on tullut strateginen kilpailuetu. Yritys sitoutuu siinä käyttämään jotain tiettyä teknologiaa pitkällä aikavälillä. Siksi oikea panostus ja osaaminen sekä tietojen ja taitojen päivitys ovat elinehto systeemityön ammattilaiselle.

Tämän lehden sisältämien artikkelien on tarkoitus antaa lukijalle käsitys tietojärjestelmien rakentamisen ja systeemityön historiasta Suomessa, sekä systeemityön luonteen muuttumisesta ajan funktiona järjestelmiä rakennettaessa.

*Erja Mustonen-Ollila
tietojärjestelmien lehtori,
tutkija
Sytyke ry:n hallituksen varajäsen,
Lappeenrannan teknillinen yliopisto,
Tietotekniikan osasto,
e-mail: erja.mustonen-ollila@lut.fi*

Spagettia - Raviolia - Lasagnea

Anne Eerola,
Kuopion yliopisto,
Tietojenkäsittelytieteen laitos

2. 1970-LUKU

2.1 Tempupulevy ja ISAC

Avainsanat

Tietojärjestelmän suunnittelu, ohjelmiston mallinnus, tiedonhallinta, ohjelmiston rakenne, oliokeskeisyys, liiketoimintaprosessit, ohjelmistotekniikka, moduulit, komponentit, arkkitehtuuri, integraatio.

1. JOHDANTO

Tarkastelen tässä artikkelissa tietojärjestelmien määrittelyn ja ohjelmistotekniikan kehittymistä Suomessa omien kokemusteni pohjalta. Esitän niitä asioita ja näkemyksiä, jotka ovat tulleet vastaan ja jotka tuntuvat edelleen tärkeiltä. Olen toiminut teollisuuden ohjelmistoammattilaisena, yliopiston opettajana ja tutkijana 1970-luvulta alkaen. Esimerkiksi Lyytinen ja Iivari ovat tarkastelleet tietojärjestelmien kehitystä ja kehitykseen vaikuttaneita suuntauksia Skandinaviassa [8].

Esitettävien asioiden ymmärtämisen helpottamiseksi määritellään muutama keskeinen käsite: *Systeemi* on joukko komponentteja, jotka vaikuttavat toisiinsa ja joilla on tietty yhteinen tarkoitus. *Systeemi* koostuu ihmisistä, koneista sovelluksista, apuvälineistä, alisysteemeistä jne. *Liiketoimintasysteemi* tai *toimialasysteemi* on perustettu toteuttamaan tietty liikeidea tai yleishyödyllinen tehtävä. *Tietojärjestelmä* (*Information System*) ohjaa tietoa, jota tarvitaan systeemissä. *Tietojärjestelmä* on sosiaalinen järjestelmä, jossa ihmiset käyttävät tietotekniikkaa, eli *ohjelmistoja* (*software*), *tietokoneita* (*hardware*) ja tietoliikenneyhteyksiä.

Opiskelin tietojenkäsittelyoppia 1970-luvun alussa Turun yliopistossa. Ohjelmointia harjoiteltiin Fortran-, Cobol- ja Assmbler-kieliä käyttäen. Ohjelmat lävistettiin *reikäkortteille*. Korttipinot laitettiin laatikkoon, josta operaattorit ajoivat ne. Parhaimmillaan ohjelman sai päivän aikana ajoon pari kertaa. Tästä oli se etu, että ohjelman logiikkaa ehti miettiä rauhassa ja *lohkokaavioita* (*flow chart*) ohjelman toiminnoista ja haarautumisista kannatti piirtää. Tähän käytettiin tempupulevyä. Ohjelmoinnin lisäksi opetuksessa keskeisellä sijalla olivat tietorakenteet, kuten pinot ja linkitetty listat, algoritmit, käyttöjärjestelmät ja kääntäjät. Codd'in (1970) esittämät *relaatiotietokantojen normaali-muodot* [4] ja *kyselyjen optimointi* tulivat tutkimuksessa tutuiksi. Sivuaineena 1970-luvun alussa opiskeltiin mm. kauppätiedettä (kauppakorkeakoulusta), matematiikkaa ja fysiikkaa.

Systemointia opeteltiin Kerolan ja Järvisen laatiman luentomonisteen avulla. Ensimmäinen ja antoisa kontakti käytännön työelämään tuli suunnitteluprojektin muodossa. Mieleen jäi yritysedustajan neuvo:

Jos tehdään kaikki, mitä käyttäjä haluaa, saadaan aikaiseksi ns. halusysteemi, jota ei halua enää kuukaan.

Systemointitaitoja syvennettiin tutustumalla ISAC menetelmään (*Information Systems work and Analysis of Change*). Menetelmä oli Suomessa suosittu 70 ja 80-luvulla. Menetelmä oli toimintopainotteinen. Mallintaminen aloitettiin kuvaamalla nyky-

tila. Mukaan tarkasteluun otettiin sekä *manuaaliset että automaattiset toiminnot*.

Menetelmässä kuvattiin (kts. Kuva 1):

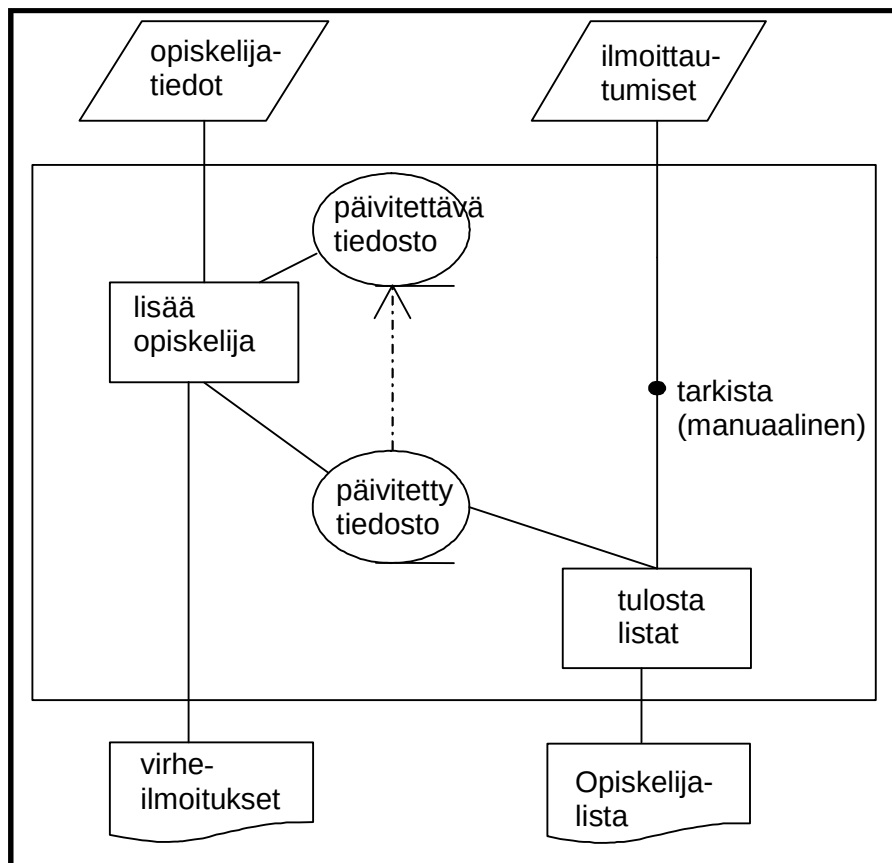
- toimintoihin tulevat syöttötiedot tai materiaalijoukot,
- tuloksena saatavat tulostiedot ja materiaalijoukot ja
- ylläpidetyt tiedostot.

Paljon huolta ja vaivaa aiheuttivat magneettinauhatiedostojen päivitykset, jotka piti suunnitella tarkasti, jotta päivitys ei kestänyt liian kauan. Nykytilan mallintamisen jälkeen suunniteltiin kehittämistavoitteet ja johdettiin tavoitetila, joka kuvattiin samalla notaatiolla kuin nykytila.

2.2 Yrityksen sisäinen atk-osasto

1970-luvulla oli tyypillistä, että isoilla yrityksillä oli omat atk-osastot, jotka ylläpitivät ja kehittivät *rääätälöintiperiaatteella* ohjelmia juuri kyseisen yrityksen tarpeisiin. Työskentelin tällaisessa organisaatiossa. Kehittämishankkeet organisoitiin projekteiksi, mutta *projektiseuranta oli löyhää*. Projektiin saatettiin lisätä tehtäviä ja työntekijöitä arvioimatta kannattavuutta uudelleen. Projektin aikataulua ja päättymistä ei seurattu, vaan projektit joko kasvoivat tai loppuivat pikkuhiljaa. Yleisesti atk-projekteja pidettiin kannattamattomina. Elinkaarimallista ei puhuttu, mutta *noudatettiin vesiputous mallia* hyvin ankaralla tavalla:

- Ensin määriteltiin asiakkaiden kanssa yhteistyössä lähinnä keskustellen ja haastattelujen avulla, mitä rakennettavan järjestelmän tulee tehdä ja mitä tietoja käsitellään. Tämä vaihe saattoi kestää 3-9 kuukautta.



Kuva 1. ISAC KAAVIO

- Kun tarpeet oli määritelty, ohjelmistoammattilaiset alkoivat toteuttaa sovellusta. Toteutusvaihe saattoi kestää 1-3 vuotta. Tänä aikana oltiin hyvin vähän yhteyksissä asiakkaisiin.

Kun ohjelma saatiin valmiiksi, se ei yleensä ollut ihan sellainen, mitä käyttäjä olisi tarvinnut.

Elettiin aikaa, jolloin atk-ammattilaisen pöydällä ei ollut tietokonetta. Asiakasvaatimuksista kirjoitettiin tekstidokumentteja, joita sihteerit kirjoittivat puhtaaksi. Lisäksi ohjelmia dokumentoitiin lohkokaaavioiden ja kommenttirivien avulla. Kaikki koneet olivat konehuoneessa, jonne piti mennä kääntämään ja ajamaan ohjelmansa. Tietokoneiden muisti ja suorituskapasiteetti oli murto-osa nykyisestä. Niinpä yhtenä isona työvaiheena oli suunnitella ohjelmien segmentointi siten, että linkitetyn ohjelman osien heittoa levyille (swapping) tapahtui mahdol-

lisimman vähän. Tavoitteena oli, että suoritus aika ei kasvaa kohtuuttomaksi ja keskusmuistiin mahtuu samanaikaisesti tarvittavat segmentit.

Tuotannossa tapahtumat kirjattiin ylös syöttölomakkeille, joilta tiedot lävistettiin keskitetyssä lävistyksessä reikäkortteille, jotka syötettiin konehuoneessa eräajo-ohjelmille.

Asiakkaiden kommentteista ja ohjeista tärkeimpänä jäi mieleen:

Tärkeintä ei ole tehdä kaikkia asioita oikein vaan tehdä oikeita asioita.

3. 1980-LUKU

3.1 Osituskäyttö ja päätteet yleistyvät

Äitiyslomalle jäädessäni 1980 käytettiin yleisesti reikäkortteja. Lomalta palatessani niitä ei ollut enää

missään. Yrityksessä siirryttiin vähitellen osituskäyttöön ja tapahtumia alettiin syöttää järjestelmään suoraan päätteiden avulla. Paljon keskusteltiin siitä, onko kyseinen menettelytapa riittävän luotettava, kun tapahtumista ei jäänyt mitään syöttölomaketta tositteeksi. Tästä syystä kehitettiin tapahtumien varmistukseen automatoituja menetelmiä.

3.2 Jokainen tekee ohjelmansa itse

Käyttäjät ja asiakkaat alkoivat olla tyytymättömiä pitkiin toimitusaikeisiin, ohjelmistotuotannon tuottavuuteen ja siihen, että sovellus ei ollut valmistuessaan sellainen kuin käyttäjä oli ajatellut. Samanaikaisesti yleistyivät mikrotietokoneet ja ns. neljännen sukupolven välineet. Ensimmäistä kertaa käyttäjät pystyivät rakentamaan itse sovelluksiaan. Välineet olivat helppoja oppia ja niiden avulla saattoi ratkaista spontaaneja tietotarpeita riittävän edullisesti ja nopeasti.

Atk-ammattilaisena tehtäväkseni jäi opettaa välineen käyttöä, suunnitella ratkaisu karkealla tasolla ja kirjoittaa ohjelma, joka poimi yrityksen tiedostoista käyttäjän tarvitsemat tiedot. Näitä käyttäjä sitten muokkasi itsenäisesti:

- valitsemalla tietojoukosta spontaanisti määritellyt ehdot täyttävät tiedot,
- lajittelemalla, ryhmittelemällä ja suorittamalla laskutoimituksia.

Tällainen väline oli esim. Mapper. Myös taulukkolaskentaa hyödynnettiin.

Menettelytapa mahdollisti ”mitä jos” analyysien suorittamisen ja päätöksentekoa tukeviin tietojärjestelmiin (esim. asiantuntijajärjestelmät), alettiin kiinnittää huomiota järeiden tuotannon suunnittelua ja tuotan-

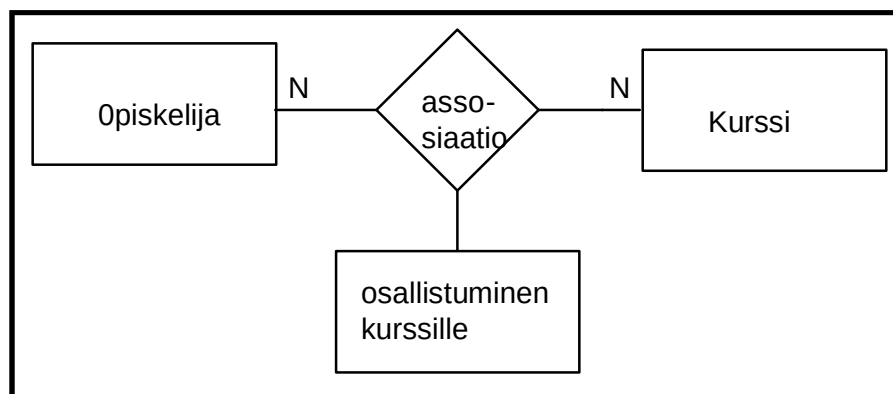
toa tukevien järjestelmien (esim. CAD/CAM) rinnalla.

Yleisesti arveltiin, että kehitys jatkuu samaan suuntaan. Vähän kerrassaan jokainen tekee ohjelmansa itse ja atk-ammattilaiset käyvät tarpeettomiksi. Asia ei kuitenkaan ollut aivan näin yksinkertainen. Mikrot ja neljännen sukupolven välineet johtivat siihen, että *tietojen redundanssi* kasvoi hallitsemattomasti. Tiedot eivät olleet enää *yhdenmukaisia* ja *ristiriidattomia*. Esimerkiksi varaston rahallinen arvo saattoi olla eri riippuen siitä laskettiin se materiaalihallinnan mikrolla vai talousosaston tietojärjestelmällä. Lisäksi suurten tietomäärien hallinnassa tuli tehokkuusongelmia. Analyysiä varten tarvittiin useita satoja tuhansia rivejä ja niitä ei pystytty analysoimaan ja lajittelemaan riittävän nopeasti.

1980-luvulla yleistyi ohjelmien prototyyppien rakentaminen. Markkinoilla oli välineitä, joiden avulla voitiin nopeasti rakentaa sovelluksesta prototyyppi, joka annettiin asiakkaalle kokeiltavaksi. Prototyyppien tärkein tehtävä oli varmistaa vaatimusten oikeellisuus ja antaa käyttäjälle mahdollisuus kokeilla sovellusta etukäteen. *Prototyypit tulivat jäädäkseen*. Niitä käytetään edelleen, joskin tehokkuus-tyistä lopullinen koodi joudutaan usein ohjelmoimaan tehokkaammalla ohjelmointiympäristöllä ja -kielellä.

3.3 Tieto yrityksen keskeinen resurssi

Vähitellen tieto alettiin ymmärtää yrityksen keskeiseksi voimavaraksi. Käsitteitä ja tietoja määriteltiin ja mallinnettiin ER-kaavioiden avulla (kts. Kuva 2) [3]. Tiedostopohjaiset järjestelmät korvautuivat vähitellen *tiedonhallintaohjelmistoilla*, joissa oli tuettuna *tietojen samanaikainen päivitys*.



Kuva 2. Yksinkertainen ER-malli

3.4 Ohjelmistokehitys asiakasryhmissä

1980-luvun puolella välissä muutin Kuopioon ja aloin ohjelmistotalo-vaihe.

Ohjelmistoa ei enää räätälöity yhdelle asiakkaalle, vaan sovelluksia kehitettiin useiden asiakkaiden toiveiden ja *prioriteettien* mukaisesti asiakasta kootun *ohjausryhmän* tuella. Apuna käytettiin muun muassa *seinätaulutekniikkaa*, jossa seinälle tai taululle hahmoteltiin kuvaavien symbolien avulla nykytilaa tai tavoitetilaa. Näitä seinätauluja päivitettiin yhdessä asiakkaiden kanssa *aivoriihime*nettelyä käyttäen. Luovuutta eristävät keinot ja toimintatavat olivat kahvipöytäkeskustelujen aiheena. Tyypillistä oli, että asiakasryhmät kokoontuivat puolenvuoden välein yhden sovelluksen ympärille. Tuotteistuksesta ei vielä puhuttu. Isompiin kehityssponnistuksiin joku asiakas lähti yleensä *pilottiyri*tykseksi osallistuen ohjelmiston määrittelyyn ja testaukseen tiiviimmin kuin muut yritykset. Yritys sai näin äänensä paremmin kuuluville ja ohjelmistot ensimmäisenä käyttöön.

*Parametrio*hjatut ohjelmistot mahdollistivat sen, että asiakkaalle ei rakennettu sovellusta alusta alkaen, vaan parametrien ja skripti-kielen avulla sovitetiin ohjelmisto asiakkaan käyttöön sopivaksi.

Projektityömenetelmiin alettiin kiinnittää entistä enemmän huomiota. Projektisuunnitelmat tehtiin tekstinkäsittelyohjelmalla ja niissä myös yritettiin pysyä. Taulukkolaskennalla hoidettiin projektiseuranta. Uusinta uutta olivat *työasemat*.

Ohjelmien rakenteeseen, dokumentointiin ja testaukseen ei kiinnitetty paljoa huomiota. Sovellukset olivat edelleen *monoliittisia* ja koodi oli yleensä ainoa dokumentti. *Uudelleen*-käytössä yleisin *menettely* oli ”copy and paste” Tämä aiheutti sen, että ohjelmarivien määrä kasvoi nopeasti. Koska kopioitua ohjelmakoodia ei yleensä kokonaan ymmärretty, siitä ei uskallettu ottaa mitään pois. Niinpä tehtiin lisää if-lauseita, joiden taakse koodattiin tarvittava lisätoiminnallisuus. Yleinen vitsi oli:

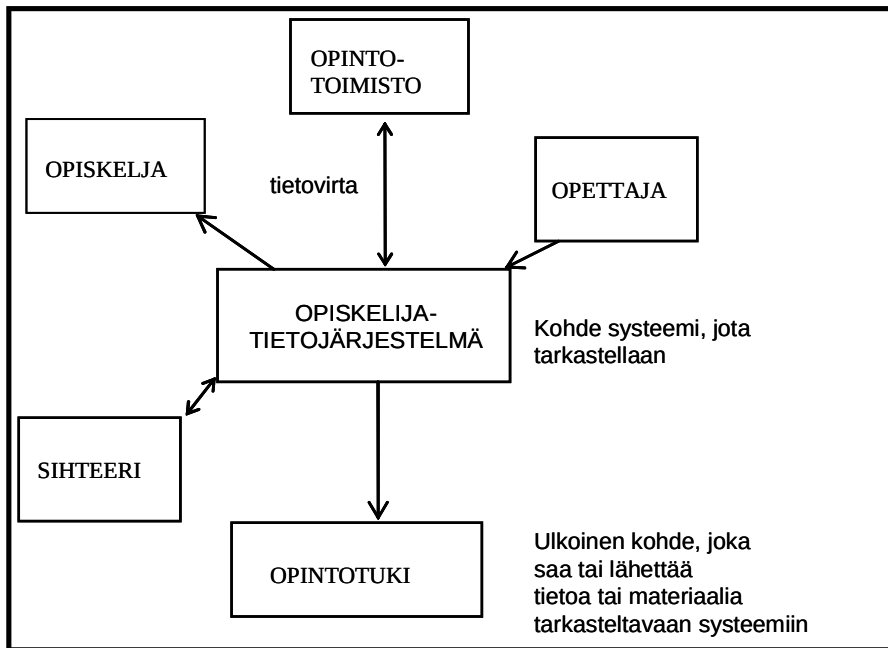
Koskaan ei ole aikaa tehdä ohjelmaa kunnolla, mutta aina on aikaa tehdä se uudelleen.

Tyypillistä oli, että ohjelmiston uudistamisen yhteydessä piti uusia kaikki laitteistosta ja tietokanta-ohjelmistosta lähtien. Jos yrityksellä oli aiemmin ollut käytössä hierarkkinen tietokanta (esim. IMS) tai verkostopohjainen ratkaisu, harkittiin olisiko syytä siirtyä *relaatiotietokantaan*. Alettiin myös keskustella *tietokantojen vaihdettavuudesta*. Nähtiin, että olisi etu, jos asiakkaalle voitaisiin myydä sovel-

4.1 Suutarin lapselle kengät

1990-luvulla huomattiin, että tietojenkäsittelyn ammattilaisetkin tarvitsivat avukseen tietotekniikkaa. CASE (Computer-Aided Software Engineering) välineet alkoivat yleistyä. Niiden avulla pystyttiin mallintamaan rakennettavaa sovellusta tietokonetta hyväksikäyttäen. Alettiin suunnitella, mitä *notaatioita* tarvitaan ja minkälaista *prosessia* tulisi käyttää ohjelmiston suunnittelussa. Tavoitteena oli, että *ohjelmistojen dokumentaatio* kehittyy ja että ohjelmia pystytään *generoimaan automaattisesti* laadittujen mallien ja kaavioiden perusteella. CASE-välineen olennaisena osana oli ja on edelleen *tietohakemisto* (*repository*), jonne projektin alusta lähtien kerätään *tietoa tiedosta*. Markkinoilla oli saatavana useita CASE-välineitä - myös suomalaisia (esim. Prosa).

Huomiota kiinnitettiin myös *laatu- ja järjestelmiin*. Yritykset ja ohjelmistotalot kuvasivat toimintansa laatukäsikirjoihin, toimintaa optimoitiin ja toiminnalle hankittiin *sertifikaatteja*.



Kuva 3. Yhteys kaavio (context diagram)

lus siten, että tietokantaratkaisu olisi asiakkaan valittavissa. Tähän ajanjaksoon liittyy yksi työurani moka. Hankin tiedonhallintaohjelmiston, jossa ei ollut samanaikaisen päivittämisen tukea. Ei tullut mieleeni varmistaa selvältä tuntuva asiaa.

3.5 Rakenteellinen analyysi

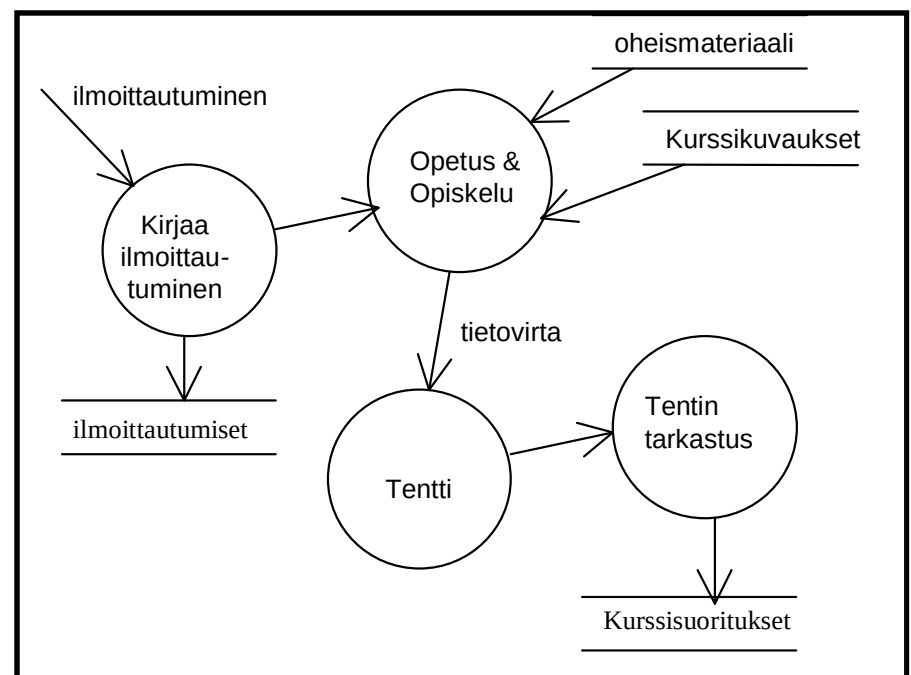
1989 siirryin opettajaksi ja tutkijaksi Kuopion yliopistoon tietojenkäsittelyopin ja matematiikan laitokselle. Käsitemallin rinnalla alettiin kiinnittää huomiota ohjelmiston toiminnallisuuden ja rakenteen kehittämiseen:

Ohjelma ei saa olla spaghetti-koodia, jossa "go-to" -käskyjä käyttäen hypitään paikasta toiseen.

Rakenteellisessa analyysissä hahmoteltiin tietojärjestelmän rajaus (Kuva 3) ja yleiskuva tietojärjestelmän toiminnasta (Kuva 4) [9]. Myös ihmisten työskentelyä tarkasteltiin (manuaaliset toiminnot).

Jälkiviisaasti ajatellen rakenteellinen analyysi on käyttökelpoinen organisaatioyksiköiden välisen *tiedonkulun* (data flow) kuvaamisessa ja

järjestelmän yleiskuvan muodostamisessa. Myös ohjelmiston rajaus ja ulkoisten yhteyksien määrittely ovat hyödyllisiä. Alemman tason kaaviot menivät helposti liian yksityiskohtaisiksi ja ne kannatti usein korvata *päättötaululla, -puulla tai rakenteellisella kielellä*. Rakenteellinen analyysi sopi huonosti tietokantapohjaisten ohjelmistojen suunnitteluun.



Kuva 4. Yleiskaavio (overflow diagram)

Organisaation prosesseja määriteltiin, kehitettiin ja uudistettiin käyttäen prosessikaavioita. *Liiketoimintaprosessien uudistamisessa (reengineering)* edellytyksenä on että kaikki toiminta tuottaa lisäarvoa. Kehittäminen fokusoitui avainprosesseihin ja organisaatiot madaltuivat (*baskerimalli*, eli kaikki hikinauhassa) [2]. Ohjelmien tarkastusmenettelyt, *parityöskentely ohjelmoinnissa ja ohjelmien testaus* alkoivat saada huomiota ohjelmistoteollisuudessa.

4.2 Objects are there, just pick them up

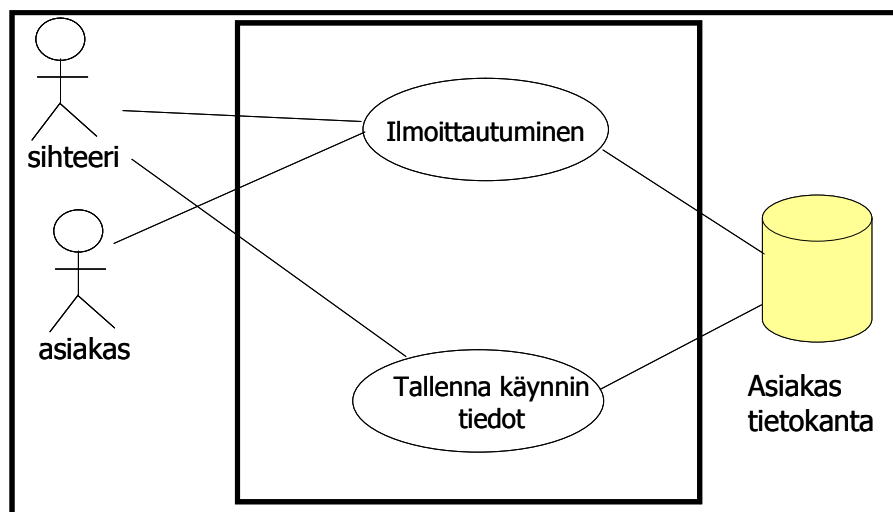
1990 luvun alkupuolella keskeiseksi systeemityön ongelmaksi tunnistettiin se että:

- Käsitemallit painotti tiedonhallintaa unohtaen toiminnot.
- Rakenteellinen analyysi keskittyi toimintoihin unohtaen tiedot

Ei oltu enää tyytyväisiä siihen, että tiedot ja toiminnot olivat erikseen. Alettiin puhua *vastuista (responsibility)* ja *oliokeskeiset (object oriented)* menetelmät alkoivat yleistyä ja niiden tutkimus ja käyttö teollisuudessa lisääntyi:

Spagetti mallista siirryttiin ravioliin [7].

Olio kapseloi tiedot (raviolissa liha) ja toiminnot (raviolissa taikina), siten, että tietoihin päästiin käsiksi vain toimintojen avulla. Alun perin 1970-luvulla kehitetty Smalltalk-ohjelmointikieli pääsi uudelleen tarkastelun ja sen rinnalla C++ yleistyi C-kielen laajenuksena. Perusideana oliokeskeisessä näkökulmassa oli reaali maailman käsitteiden ymmärtäminen olioiksi, joilla oli ominaisuutena tiedot ja näihin liittyvät toiminnot. Järjestelmän toiminnallisuus saatiin aikaan *olioiden* välisen *viestinvälityksen* avulla. Oliot kuvattiin *luokkahierarkiassa*, jossa alaluokka peri yläluokan ominaisuudet.



Kuva 5. Käyttötapaus ilman ihmisten aktiviteetteja

Rakenteellinen analyysi ei tietenkään soveltunut oliokeskeisen järjestelmän suunnittelumenetelmäksi. Paremman lähtökohdan antoi käsiteanalyysi, jonka pohjalta syntyi *useita oliokeskeisiä suunnittelumenetelmiä* 1990 luvun alussa. Olioehdokkaista olivat substantiivit, kuten käsitteet, tiedot, tapahtumat, henkilöt, organisaatiot, systeemit ja koneet. Verbit olivat metodiehdokkaista ja ne pyrittiin sijoittamaan oikeaan olioon. Tämä ei ollut aina helppoa.

Monet oliot esiintyivät sekä reaali maailmassa (esim. kirjaston asiakas) ja ohjelmistossa (asiakastiedot). Jälkiviisaasti ajatellen tuntuu siltä, että "object oriented" -sana suomennettiin harhaanjohtavasti: Olio sanan käyttö (asiakas olio, potilas olio jne.) tuntui hölmöltä ihmisistä puhuttaessa. Olio viittasi johonkin elävään. Parempia käännöksiä sanalle olio olisivat ehkä olleet objekti, entiteetti, kohde tai subjekti.

Moniperinnöllisyys (luokalla enemmän kuin yksi yläluokka) aiheutti tutkijoille erityisen paljon päänvaivaa. Oliokeskeisyydellä tavoiteltiin parempaa modulaarisuutta [12]:

- vähemmän riippuvuuksia ohjelman osien välillä (*coupling*) ja

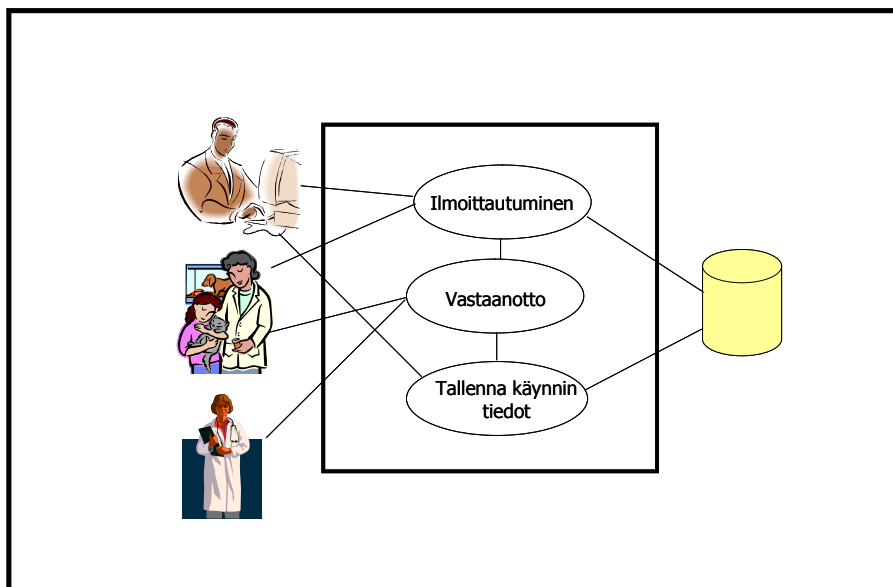
- parempi *koossapysyvyys (cohesion)*, eli olio muodostaa tiiviin kokonaisuuden.

Luokkahierarkian avulla tavoiteltiin *uudelleenkäytön* lisääntymistä. Käytännössä asiat eivät edenneet ihan tavoitteiden mukaisesti, Wilde ja Huit kirjoittivat artikkelissaan [16]: "Oliokeskeisessä ohjelmassa on perinteiseen ohjelmaan verrattuna huomattavasti enemmän ja erityyppisiä riippuvuuksia. Lisäksi olion toiminnan ymmärtämiseksi joutuu tutkimaan yläluokkia ja alaluokkia vuorotellen (*jojo-ilmiö*)".

Onkohan niin, että kun ohjelmistotalan ammattilaiset alkoivat 1990-luvulla puhua olioista, samalla etäisyys ohjelmistojen hyödyntäjiin kasvoi. Asiaa ei auttanut se, että tietojärjestelmätutkimus ja tietojenkäsittelytiede etäännyivät toisistaan.

4.3 Unified Modelling Language

UML (Unified Modelling Language) kielen käyttö ohjelmien mallinnuksessa yleistyi. Näin ohjelmistotalan ammattilaiset eri puolilla maailmaa pystyivät ymmärtämään toisiaan. UML:ää tukemaan kehitettiin RUP – prosessikuvaus (Rational Unified Pro-



Kuva 6. Käyttötapaus, jossa on mukana ihmisten työ

cess) [10]. Pienien ja keskisuurten yritysten käyttöön RUP on liian raskas ja laaja. Sitä pitää supistaa. Tähän on tuki olemassa.

Käyttötapauskaaviot skenaarioineen tulivat ohjelmistoteollisuudessa suosituksi kuvaustavaksi. Määritelmän mukaan käyttötapaus on tuokio-kuva ihmisen ja tietokoneen vuorovaikutuksesta (kts. Kuva 5). Tässä on heikkoutena se, että ihmisten toimintaan ilman tietokonetta ei kiinnitetä huomiota. Kaavioista ei selviä tarkasteltava toimintokokonaisuus eikä sitä näin ollen voida kehittää. On hyödyllistä kuvata käyttötapauskaavioon myös ihmisten toiminta, eli *toimintatapaukset* (*action case*), jotta voidaan ottaa huomioon ihmisten työ (kts. Kuva 6). Samalla ohjelmistojen *käytettävyyttä parane*e, kuvauksista tulee ymmärrettävämpiä ja niiden järjestyksen pystyy tarkistamaan.

UML:ssä käytetään prosessien kuvaukseen *aktiviteettikaaviota* (*activity diagram*), jonka yhdenmukaisuus 1970-luvun lohkokaavion kanssa on ilmeinen. Uutena asiana on organisaatioyksiköjä kuvaavat kanavat ja prosessien tahdistuksen kuvaaminen. Näiden avulla reaali maailmassa luonnollinen rinnakkaisuus pystytään mallintamaan. *Luokkakaaviot* (*class*

diagrams) ovat ohjelmistoammattilaiselle hyödyllinen väline. Ongelmana on, että kaavioon tulee helposti liikaa asiaa. Tilannetta voi helpottaa kuvaamalla erikseen määrittely-, käyttöliittymä- ja toteutustason luokkakaaviot [5].

Kokemusten mukaan ohjelmistojen hyödyntäjät eivät kovin hyvin ymmärrä luokkakaavioita ja niissä esitettyjä asioita. Asiakkaiden kanssa kommunikointi sujuu prosessikaavioita, käyttötapauskaavioita ja skenaario-kaavioita käyttäen. Kannattaa muistaa, että ennen kuin aletaan määritellä ja suunnitella ohjelmistoa pitää tarkastella työkokonaisuutta, jota ollaan tukemassa.

UML:n heikkoutena on, että se antaa kovin vähän tukea arkkitehtuurin ja komponenttien kuvaamiseen. Tämä on ikävä asia, koska arkkitehtuurien kehittäminen on tänä päivänä keskeistä niin käytännön tietojärjestelmäsuunnittelijoiden kuin tutkijoidenkin keskuudessa.

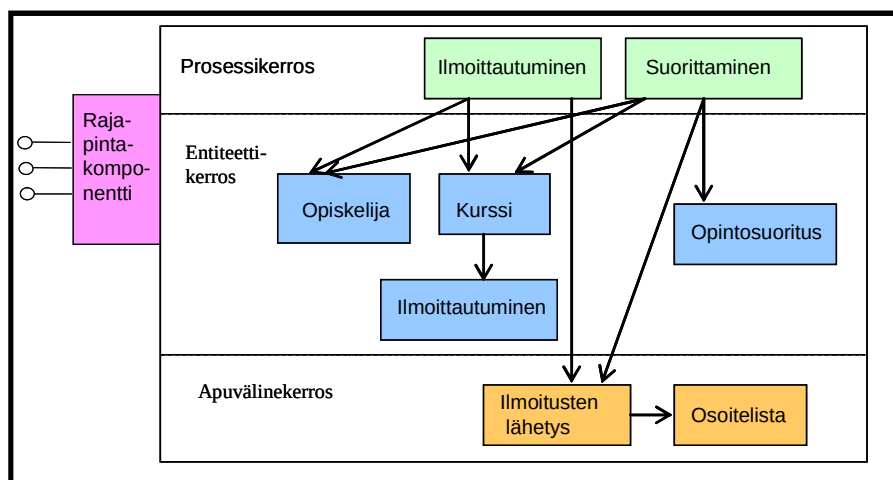
4.4 Komponentit ovat eri kokoisia

1990-luvulla yritykset alkoivat muodostaa yhteistoimintaverkkoja. Tänä päivänä on yleistä, että yritys

toimii hajautetusti, tietojärjestelmät ovat hajautettuja ja jopa tiimit toimivat hajautetusti. Ohjelmistoyrityksen kilpailutekijäksi on noussut tehokkuus, toimitusaika, laatu ja luotettavuus. Uudelleen käytön pitäisi kehittyä ja monimutkaisuus pitää hallita. Samanaikaisesti sovelluskehitys on vaikeutunut. Ei ole helppo kehittää ohjelmistoja, jotka mahdollistavat uuden teknologian, kuten hajautus, internet/intranet, tietoliikenne, mobiliteknologia. Ohjelmistotuotanto on siis, toisin kuin 1980-luvulla uskottiin, tullut entistä vaativammaksi. Kukaan alan ammattilainenkaan ei hallitse kaikkia tekniikoita. Tarvitaan erikoistumista, mutta pitää myös muistaa yhteisen kielen muodostuminen ja säilyttäminen.

Komponenteissa on modulaarisuus saatu ensi kertaa toteutettua kohdittuun hyvin: Komponentilla on *rajapinta* (*interface*), jonka kautta sen tarjoamia palveluja kutsutaan. Komponentti tarvitsee toimiakseen yleensä toisia komponentteja ja infrastruktuuria. Myös *nämä riippuvuudet* (*dependency*) määritellään selkeiden rajapintojen avulla. Komponentit *ovat eri kokoisia* (*granularity*) lueteltuna pienimmästä suurimpaan [7]:

- *Hajautettu komponentti* (*Distributed Component*) toteuttaa ja kätkee hajautuksen. Sillä on yksi tai useampi hyvin määritelty rajapinta, johon voidaan viitata verkosta käsin. Nimi hajautettu komponentti on hiukan harhaanjohtava, koska komponentin ei tarvitse olla hajautettu. Tyypillisesti tänä päivänä hajautettu komponentti voi olla EJB, COM, CORBA jne. teknologialla toteutettu.
- *Toimialakomponentti* (*Business Component*) koostuu hajautetuista komponenteista. Toimialakomponentin rajapinta muodostuu sen hajautettujen komponenttien rajapinnoista. Toimialakomponentti noudattaa *kerrosarkkitehtuuria* ja sisäl-



Kuva 7: Komponenttisysteemi

tää tarpeesta riippuen käyttöliittymä-, työtila-, toimintalogiikka- ja resurssikerroksen. Näin Italian keittiön *lasagne korvaa raviolin*.

- *Komponenttisysteemi (Business Component System)* koostuu toimialakomponenteista, jotka ovat tyypiltään esim. prosessi-, käsite-, apu- tai rajapintakomponentteja (kts. Kuva 7). Komponenttisysteemin rajapinta muodostuu sen toimialakomponenttien rajapinnoista.

Ohjelmien oikeellisuudelle asetetaan entistä kovempia vaatimuksia, joten testausmenetelmiä on kehitettävä [17].

4.5 Ohjelmistotekniikan koulutusta

1990-luvun lopulla useat tahot havahtuivat siihen, että ohjelmistoteollisuus tarvitsee palvelukseensa lisää tietojenkäsittelytieteen maistereita. Tähän liittyen Kuopion kaupunki lahjoitti Kuopion yliopistolle ohjelmistotekniikan professorin viran (5 vuotta). Alkoi *ohjelmistotekniikan opetuksen ja tutkimuksen kehittäminen*. Tietojenkäsittelyn opetusta laajennettiin käytännönläheiseen suuntaan huomioiden suurten, usein hajautettujen, tietojärjestelmien rakentamisen vaike-

us. Opetusta lisättiin seuraavilla aihealueilla:

- Ohjelmistotuotanto, tuotteenhallinta.
- Mallinnusmenetelmät, ml. liiketoimintaprosessien ja ihmisten toimintakokonaisuuksien mallintaminen ja kehittäminen.
- Ohjelmiston rakenne: komponentit, arkkitehtuurit, suunnittelumallit.
- Ohjelmiston testaus.
- Käytettävyyden suunnittelu.
- Hajautetut järjestelmät.

Näkökulmana ovat olleet ohjelmiston hyväksikäyttäjän tarpeet, ohjelmistoyrityksen toiminta, ja ohjelmistotoimittajien työ. Ohjelmistotekniikan tutkimuksessa tarkastellaan samoja asioita kuin opetuksessa.

5. 2000-LUKU

Komponenttien kehittämisen *riinnalla arkkitehtuurin suunnittelu* on nousut keskeiseksi kehittämisasiaksi niin ohjelmistoteollisuudessa kuin tutkimuksessakin. Arkkitehtuurin suunnittelu, kuvaaminen ja arviointi ovat tärkeitä, koska näin voidaan etukäteen (ennen ohjelman koodaamista) arvioida sovelluksen toiminnallisten ja laadullisten ominaisuuksien toteutumismahdollisuuksia [1]. Tämä säästää kalliilta virheinvestoinneilta. Lisäksi

arkkitehtuurikuvaukset mahdollistavat ajatustenvaihdon, neuvottelun ja yhteisymmärryksen muodostumisen sovelluksen ominaisuuksista komponentin toimittajan, komponentin integroijan ja sovelluksen hyväksikäyttäjän välillä [15].

Ilahduttavaa on, että alunperin Gamm:an julkistamat *suunnittelumallit (design pattern)* [6] kiinnostavat ohjelmistotuottajia, opiskelijoita ja tutkijoita. Suunnittelumallien avulla tehdään *piilossa oleva suunnittelutietämys näkyväksi*. Omaa tietoutta voi dokumentoida tai voi käyttää hyödyksi muiden kehittämiä malleja. Suunnittelumallien avulla helpotetaan hajautetun sovelluksen rakentamista (esim. välitin ja edustaja/korvike), sovitetaan ohjelmistoja (adapters) tai kääritään perinnejärjestelmiä (wrappers).

Tänä päivänä sovelluksilta edellytetään *yhteistoiminnallisuutta (interoperability)*. Integraatiota toteutetaan työpöytä-, toimintalogiikka- ja tiedostotasolla. Tärkeäksi mahdollisuudeksi yritysten ja yritystenvälisen toiminnan tehokkuuden ja laadun nostajana on noussut prosessi-integraatio [11]. Integrointitekniikoiden lisäksi integrointiprosessi on tarkastelun ja kehittämisen kohteena [13],[14].

Terveystieteiden sovellusintegroatiota tutkitaan ja kehitetään Mikko Korpelan vetämässä PlugIT-hankkeessa, jonka rahoittajina ovat TEKES sekä useat ohjelmistotalot ja sairaanhoitopiirit. PlugIT tuottaa avoimia ohjelmistorajapintojen määräyksiä sekä niihin liittyviä menetelmiä ja osaamista terveydenhuollon ohjelmistoyrityksille ja niiden asiakkaille. Hankkeen tavoitteena on tukea terveydenhuollon palvelutoimintaa ohjelmistotuotannon palveluketjun kautta, paremmin integroituvien ohjelmistokokonaisuuksien avulla. Hankkeessa on eri ammattiryhmien aitoa yhteistyötä. Muka-

na on kolme Kuopion yliopiston ja yksi Pohjois-Savon Ammattikorkeakoulun tutkimusyksikkö (kts. <http://www.uku.fi/atkk/plugin>).

Uusimpana tutkimusaiheena on *käytettävyyden kehittäminen*. Ohjelmiston arkkitehtuuria ja rakennetta parannetaan huomioiden käyttäjien toimintaprosessit, ongelmanratkaisutavat ja oppimisen. Jos olet havainnut käytettävyysongelman, lähetä siitä lyhyt kuvaus Mirja Immoselle (Mirja.Immonen@cs.uku.fi). Kiitos!

6. Loppuyhteenveto

Tätä historiaa kirjoittaessa tuli mukava olo. Paljon me ohjelmistomattilaiset olemme vuosien aikana oppineet. Olkoonkin, että pyörä on keksitty moneen kertaan, on silti päästy eteenpäin. Tekniikkaa osataan. Entistä parempiin tuloksiin varmasti päästään, jos ohjelmiston toimittajan, integroijan ja asiakkaan yhteistoimintaa kehitetään entisestään. Suomen kaltaisessa pienessä maassa ei kannata sortua vääränlaiseen kilpailuun. Yhteistyö ja korkeatasoinen osaaminen mahdollistavat kansainvälisen ohjelmistoliiketoiminnan lisääntymisen.

Ihmiset tekevät ohjelmistot, ihmiset hyödyntävät ohjelmistoja - ei unohdeta ihmistä.

LÄHDELUETTELO

- [1] Bosch J.: *Design & Use of Software Architectures, Adopting and evolving a product-line approach*, Addison-Wesley, 2000.
- [2] Berztiss A.: *Software Methods for Business Reengineering*, Springer-Verlag, 1996.

- [3] Chen P. P.: *The Entity-Relationship Model – Toward a Unified View of Data*, ACM Transactions on Database Systems, Vol. 1, No. 1, March 1976, pp. 9-36.
- [4] Codd E.: *A Relational Model of Data for Large Shared Data Banks*, Communications of the ACM, Vol. 13, No. 6, 1970, 377-387.
- [5] Fowler M., Scott K.: *UML Distilled, Applying the Standard Object Modelling Language*, Addison-Wesley, 1997.
- [6] Gamma E., Helm R., Johnson R., Vlissides J.: *Design Patterns, Elements of Reusable Object-Oriented Software*, Addison-Wesley, 1995.
- [7] Herzum P., Sims O.: *Business Component Factory*, Wiley Computer Publishing, New York, 2000.
- [8] Iivari J., Lyytinen K.: *Research on information systems development in Scandinavia – unity in plurality*, Scandinavian Journal of Information Systems 1998: 10 (1&2):135-185.
- [9] Kendall A.: *Introduction to Systems Analysis and Design, A Structured Approach*, Elsevier Science Publishers B.V. (North-Holland), ERI 1989.
- [10] Kruchten, P.: *The Rational Unified process, an introduction*, Addison-Wesley, 2001.
- [11] Linthicum D. S.: *B2B Application Integration e-Business-Enable Your Enterprise*. Addison-Wesley Information Technology Series, USA, 2001
- [12] Meyer B.: *Object-oriented Software Construction*, Prentice Hall, 1988
- [13] Mykkänen J., Porrasmaa J., Korpela M.: A process for specifying integration for multi-tier applications in healthcare. In: Health Data in the Information Age. Proceedings of the 17th International Congress of the European Federation of Medical Informatics MIE 2002 (25-29 August 2002, Budapest). Surján G, Engelbrecht R, McNair P, eds. p. 691-696. IOS Press, Amsterdam 2002.
- [14] Mykkänen J., Tikkanen T., Rannanheimo J., Eerola A., Korpela M.: *Integration of Health Information Systems – Specification Levels and Collaborative Definition*, MIE 2003.
- [15] Smolander K.: *On the Role of Architecture in Systems Development*, Academic dissertation, Lappeenrannan teknillinen yliopisto, 2003.
- [16] Wilde N., Huitt R., 1992: *Maintenance Support for Object-Oriented Programs*, IEEE Transactions on Software Engineering, Vol. 18, No. 12, Dec 1992, pp. 1038-1044.
- [17] Toroi, T., Eerola, A., Mykkänen, J.: *Testing Business Component Systems*, raportti, Kuopion yliopisto, 2002.

Kirjoittaja työskentelee ohjelmistotekniikan professorina Kuopion yliopiston Tietojenkäsittelytieteen laitoksella.

e-mail: Anne.Eerola@cs.uku.fi

Ohjelmistotuotanto 2000-luvulle

*Jorma Sajaniemi,
Joensuun yliopisto,
Tietojenkäsittelytieteen laitos*

Ohjelmistojen rakentaminen on tunnetusti kallista ja tulosten laatu usein huono. Seuraavassa tarkastellaan tähän johtaneita syitä sekä tutkimuksen haasteita 2000-luvulle siirryttäessä.

Ohjelmistojen rakentamisessa ilmenevät ongelmat eivät ole satunnainen ja irrallinen ilmiö, vaan niillä on selvä yhteys tietojenkäsittelyssä tapahtuneeseen kehitykseen. Kukin aikakausi on pyrkinyt kehittämään hyviä työskentelytapoja, mutta aiemmista tavoista luopuminen on aina ollut vaikeaa. Jotta ymmärtäisimme nykyiset ongelmat paremmin, on syytä katsoa mistä lähtökohdista eri vuosikymmeninä on ponnisteltu ja kuinka ne ovat vaikuttaneet työmenetelmiin sekä ohjelmistotyön kehittämiseen.

Yksinkertainen 1970-luku

Reikäkorttien aikaan 1970-luvulla tietokoneet olivat kalliita ja niiden käytön kannattavuus oli laskettava tarkkaan ennen järjestelmän suunnitteluun ryhtymistä. Järjestelmän rakentamisen aikanakin laitteiden käyttömahdollisuudet olivat vähäisiä, jonka johdosta ohjelmien suunnittelu ja kirjoittaminen piti tehdä mahdollisimman pitkälle ilman koneiden apua. Kynä ja paperi olivat suunnittelulle ja jopa ohjelmoinnille tärkeämpiä apuvälineitä kuin tietokoneet. Jopa saman valmistajan laitteissa oli erilaiset, toistensa kanssa yhteensopimattomat käyttöjärjestelmät.

Rakennettavat järjestelmät olivat tuolloin hyvin yksinkertaisia ja niillä oli yhteyksiä toisiin järjestelmiin vain hyvin niukasti. Rakentamisen jälkeen tarvittava ylläpito oli vähäistä, koska laitteisto pysyi samana, ympäröivät jär-

jestelmät pysyivät muuttumattomina ja muutenkin ulkoa ei juurikaan tullut paineita muutoksiin. Järjestelmien suunnittelusta ja toteutuksesta vastasi tuolloin yleensä muutaman hengen ryhmä, joka koostui yrityksen omasta väestä - ja jollei omaa atk-väkeä ollut, niin se oli sitten hankittava.

Näillä lähtökohdilla 1970-luvun ohjelmistoprosessi koostui tyypillisesti seuraavista vaiheista:

- Esitutkimus: sen selvittäminen, voiko atk ylipäättään tuoda säästöjä
- Systeemin suunnittelu: järjestelmän yleissuunnittelu ja konetyypin valinta
- Suunnittelu: järjestelmän yksityiskohtien ja ohjelmien logiikan suunnittelu
- Ohjelmointi: ohjelmien kirjoittaminen, pöytätestaus, siirto koneelle ja konetestaaminen
- Instalointi: järjestelmän asennus lopulliseen käyttökuntoon ja käyttöönotto
- Laajentaminen: uusien mahdollisuuksien etsiminen käytön aikana

Tällainen prosessi oli omalle ajalleen varsin sopiva: kun järjestelmät olivat yksinkertaisia ja muutokset vähäisiä, ei monimutkaisempaa rakentamisprosessia tarvittu.

Ohjelmistokriisi

Ensimmäiset vakavat ongelmat alkoivat ilmaantua 1980-luvulla ja 1990-luvulle siirryttäessä ne olivat jo kasaantuneet ohjelmistokriisiksi asti. Ulospäin tilanne näkyi ohjelmistojen heikkona laatuna: järjestelmissä oli paljon virheitä eivätkä ne vastanneet käyttäjien tarpeita. Myös järjestelmien tekemisessä oli pahoja ongelmia: projektit venyivät ja järjestelmät myöhästyi, tekeminen olikin huomatta-

vasti suunniteltua kalliimpaa ja hankkeita lopetettiin, kun niistä ei syntynyt tulosta.

Mutta näitä ulos näkyviä syitä tärkeämpää on ymmärtää se, mistä nämä ongelmat oikeastaan juontuivat.

Ohjelmistokriisin olennainen syy oli monimutkaisuuden kasvu. Järjestelmien koko kasvoi, jonka johdosta niitä ei pystytty enää käsittelemään ja ymmärtämään ilman apuvälineitä. Siirtyminen eräajoista suorakäyttöisyyteen monimutkaisti ohjelmien käyttöliittymää merkittävästi. Ohjelmaversioiden tiheä ilmestymistarve pakotti tekemään uutta koodia jo ennen edellisen version valmistumista ja tiedostojen versioinnin hallitseminen muuttui hankalaksi. Lisäksi järjestelmien käyttäjä- ja asennusmäärät kasvoivat, jolloin asennusten ylläpito vaikeutui ja muutosvaatimuksia alkoi tulla satamalla.

Myös järjestelmien sisäinen monimutkaisuus kasvoi. Järjestelmän eriosat sijoitettiin eri laitteille, jolloin erilaisten laiteyhdistelmien määrä kasvoi voimakkaasti. Järjestelmistä tehtiin erilaisia variaatioita eri alustoille, jolloin halutun yhdistelmän kokoaminen muuttui hankalaksi. Ja suuret järjestelmät vaativat lisäksi paljon tekijöitä, jotka saattoivat jopa työskennellä eri paikkakunnilla, jolloin kommunikointi ja tiedostojen päivittäminen muuttuivat ongelmallisiksi.

Ohjelmistokriisin perimmäinen syy oli siis yksinkertaiseen maailmaan sinänsä hyvin sopivien menetelmien riittämättömyys uudessa, monimutkaisessa maailmassa.

Monimutkaisuus hallintaan

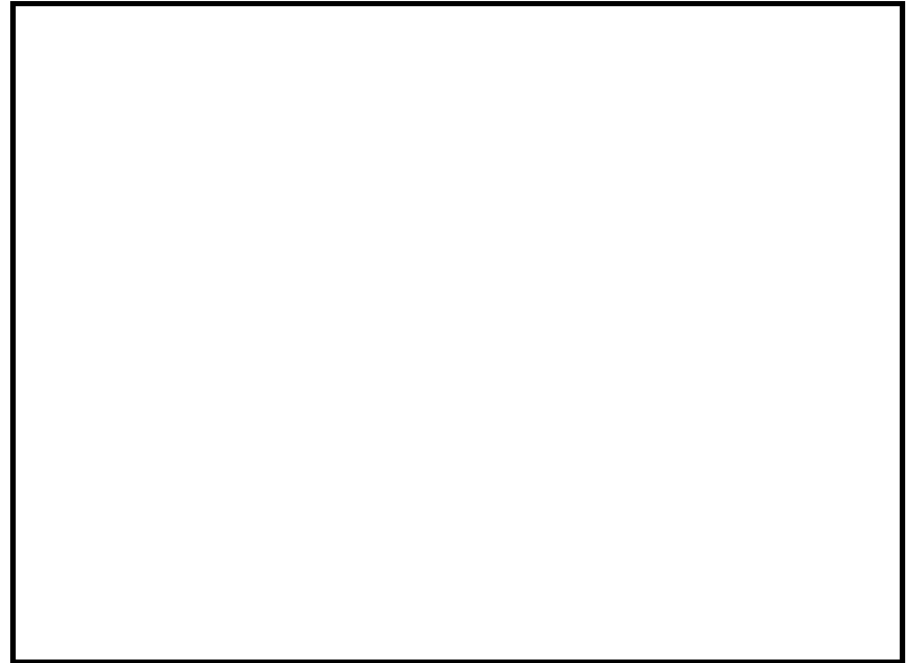
Ohjelmistokriisi synnytti tietenkin myös ratkaisuja tilanteen parantamiseksi. Nämä ratkaisut voidaan jakaa

karkeasti kolmeen ryhmään: uudet elinkaarimallit, tukitoimintojen kehittäminen sekä yksittäisten vaiheiden tehostaminen uusilla menetelmillä ja välineillä.

1970-luvun yksinkertaisen mallin tilalle kehitettiin ohjelmistotyön ohjaamiseen sellaisia malleja kuin vesiputousmalli, spiraalimalli, prototyyppi-
en käytön malli ja vähittäisen kehittämisen malli. Näille on yhteistä määrittysten tiukka kiinnittäminen ja muutosten salliminen vain harvoissa kohdissa ja myös ajallisesti vain harvoin. Mallien avulla etsittiin ratkaisuja muutosten, epävarmojen määritysten ja suurten tekijäjoukkojen hallintaan.

1990-luvulla ympäristössä tapahtuvat muutokset kiihtyivät entisestään ja jopa järjestelmän pohjaksi valittuja perusteknologioita voitiin joutua vaihtamaan muutaman kuukauden välein. Perinteiset elinkaarimallit osoittautuivat liian jäykiksi tällaisiin tilanteisiin, ja niiden tilalle kehitettiin kokonaan uusia lähestymistapoja. Tällaisia ovat esimerkiksi XP (extreme programming) ja Scrum, jotka painottavat nopeaa reagointia muuttuviin olosuhteisiin ja notkeutta suuriinkin muutoksiin. Ne painottavat vaatimusten tarkkaa (mutta nopeaa) hallintaa, ohjelmistotyön turhien toimintojen karsimista sekä jäljellejääneiden tärkeiden toimintojen huolellista suorittamista. Tällaiset menetelmät auttavat toimimaan tilanteessa, jossa nopea tekninen kehitys muuttaa järjestelmälle asetettuja vaatimuksia jo järjestelmän kehitystyön aikana, mutta lisääntyneeseen monimutkaisuuteen nekin eivät tuo olennaista helpotusta.

Monimutkaisuuden hallintaan kehitettiin ohjelmistotyön eri vaiheisiin erilaisia menetelmiä ja apuvälineitä. Tällaisten käyttö esimerkiksi suunnitteluvaiheessa edistää hyvään lopputulokseen pääsemistä. Testauksen apuvälineet puolestaan voivat vaikkapa automatisoida taantumatestauksen



Kuva: Uusi, monimutkainen maailma teki vanhoista menetelmistä käytökelvottomia ja johti ohjelmistokriisiin.

suorittamisen, jolloin tämä toimenpide tulee tehtyä riittävän tiheästi ja ohjelmakoodiin vahingossa pujahtaneet virheet löytyvät ennen järjestelmän toimittamista eteenpäin. Käytökelpoisuudestaan huolimatta menetelmät ja apuvälineet tuovat kuitenkin parannuksia vain ohjelmistotyön yksittäisiin tehtäviin.

Ratkaisu monimutkaisuuteen ei siis löytynyt pelkästään elinkaarimallien ja apuvälineiden kehittämisestä, vaan lisäksi tarvittiin myös ohjelmistotyön tukitoimintojen kehittämistä. Tukitoiminnot eivät ole itse järjestelmän tekemistä vaan niitä oheistoimintoja, jotka tarvitaan tekemisen hallittavuuden aikaansaamiseksi. Tärkeimmät näistä ovat tuotteen hallinta, versionhallinta, muutosten hallinta, laadun hallinta ja projektinhallinta.

Kun rakennettava järjestelmä on monimutkainen ja siitä on tuotettava tiheässä tahdissa erilaisia versioita erilaisiin ympäristöihin, antavat tuotteen hallinta ja versionhallinta eväät tiedostokaaoksen pitämiseen kurissa. Kun ympäräivä maailma ja asiakkaan tarpeet sekä mieltymykset vaihtuvat tiuhaan tahtiin, muutosten hallinta an-

taa kehittäjille työrauhan ja estää turhat muutokset jopa kokonaan. Kun monimutkaisuus uhkaa järjestelmän luotettavuutta, laadun hallinta pyrkii estämään virheiden synnyn ja auttaa havaitsemaan virheet nopeasti ennen kuin ne aiheuttavat jatkovirheitä. Ja kun tiukat aikataulut ja niukat budjetit uhkaavat kaataa koko kehitystyön, antaa projektinhallinta eväät realististen suunnitelmien tekoon ja niiden toteuttamiseen.

Askel 2000-luvulle

2000-luvulle siirryttäessä monimutkaisuus jatkaa yhä kasvuaan. Vaikka laitteistojen ja käyttöjärjestelmien osalta onkin tapahtumassa kirjavuuden vähenemistä, jatkuu välikerroksen tuotekehitys nopeana ja lisäksi markkinoiden globalisaatio kasvat-
taa esimerkiksi tuotteiden lokalisoinnin tarvetta. Onkin ilmeistä, että ohjelmistotyön nykyiset menetelmät eivät tule riittämään yhä uusissa tilanteissa, vaan jatkossa tarvitaan parempia ja voimakkaampia menetelmiä. Tällaiset menetelmät eivät kuitenkaan synny itsestään vaan niiden kehittämiseen tarvitaan tutkimusta. Ja mitä vaativampaan tilanteeseen niitä tarvi-

taan, sitä korkeatasoisempaa tämän tutkimuksen on oltava.

Tutkimuksen keskeisiä kysymyksiä tulevat jatkossa olemaan tukitoimintojen parempi ymmärtäminen sekä niiden kehittäminen ja automatisoiminen tämän ymmärryksen pohjalta. Esimerkiksi laadunhallintaan liittyvien tarkastusmenettelyjen automaatioaste on nykyisin varsin vaatimaton ja Joensuun yliopistossa onkin panostettu sen kehittämiseen. Toisaalta nykyistenkin tekniikoiden hyväksikäyttö on varsinkin pienissä ja keskisuurissa yrityksissä yllättävän vähäistä. Niin-

pä on myös panostettava sellaisten menetelmien kehittämiseen, jotka voidaan ottaa helposti käyttöön ja jotka siitä huolimatta takaavat pääosan raskeampien menetelmien hyödyistä. Esimerkiksi Lappeenrannan teknillisessä yliopistossa on kehitetty tällainen menetelmä vaatimusten hallintaan.

Uusien menetelmien ja välineiden kehittämisessä on myös huomioitava ihmisen kognitiiviset rajoitukset ja kognitiivisen ergonomian vaatimukset: ei ole järkevää kehittää välineitä, jotka edustavat tekniikan uusimpia mahdollisuuksia, jos ohjelmistotyön

tekijät eivät todellisuudessa pysty niitä käyttämään. Niinpä uudet tekniikat on testattava empiirisin menetelmin, mikä asettaa aivan uudenlaisia haasteita alan tutkimukselle. Mutta ilman tutkimusta menetelmät jähmettyvät ja maailman jatkuva monimutkaistuminen juoksee niiden ohi!

*Jorma Sajaniemi,
Joensuun yliopisto,
Tietojenkäsittelytieteen laitos,
PL 111,
80101 Joensuu,
Jorma.Sajaniemi@joensuu.fi*

Älytekniikan foorumi

15.5.2003

Seinäjoki Areena,
Kirkkokatu 23, 60220 Seinäjoki

Seinäjoen seudun
aluekeskusohjelma



08.30 – 09.00	Ilmoittautuminen ja kahvi, näyttelyyn tutustuminen	13.30 – 14.00	Kahvi, näyttelyyn tutustuminen
09.00 – 09.05	Avaussanat	14.00 – 14.30	Ohjausjärjestelmät vaativissa olosuhteissa
	Puheenjohtaja, Professori Markku Kivikoski		Teknologiajohtaja Eero Yli-Ojanperä
	Tampereen teknillinen yliopisto, Elektroniikan laitos	14.30 – 15.00	Sulautetun järjestelmän vikasietoisuus
09.05 – 09.50	Viestintätekniikan mahdollisuuksia ja haasteita		Professori Hannu-Matti Järvinen
	Dr, Senior Research Manager Jukka Rantala	15.00 – 15.15	Tampereen teknillinen yliopisto, Ohjelmistotekniikan laitos
	Nokia Research Center, Electronics Laboratory	15.15 – 15.45	Tauko, näyttelyyn tutustuminen
09.50 – 11.00	Higher Layer Protocols		EMC - sähköstaattinen purkaus
	Managing Director Holger Zeltwanger		TkT Lauri Palva
	CAN in Automation (CiA), Erlangen, Germany	15.45 – 16.15	Teknillinen korkeakoulu, Sovelletun elektroniikan laboratorio, Espoo
11.00 – 12.00	Lounas, näyttelyyn tutustuminen		Käytettävyyden ohjelmistoprojektin eri vaiheissa
12.00 – 12.30	Etäanturitekniikka		DI, Käytettävyytutkija Inka Vilpola
	TkT Timo Varpula		Tampereen teknillinen yliopisto, Ohjelmistotekniikan laitos
	VTT Tietotekniikka, Espoo	16.15 – 16.30	Loppusanat
12.30 – 13.00	Automaatiojärjestelmien hajautusta palvelevat arkkitehtuurit		Puheenjohtaja, Professori Markku Kivikoski
	DI, Tutkija Jukka Peltola		Tampereen teknillinen yliopisto, Elektroniikan laitos
	Teknillinen korkeakoulu, Automaation tietotekniikka, Espoo	16.30 – 18.00	Seinäjoen kaupungin vastaanotto
13.00 – 13.30	Koneiden välinen tiedonsiirto, M2M		Seinäjoki Areena
	FM, Tutkija Leila Rannanjärvi		
	VTT Elektroniikka, Oulu		

Seminaarin hinta: 150 € (alv 0%) sisältäen luennot, materiaalin, lounaan, kahvit ja kaupungin vastaanoton.
Ilmoittautuminen: 06-4166 284, assistentti Jaana Paavola, jaana.paavola@seinajoki.fi.
Lisätietoja: 050-557 5547, ICT-koordinaattori Jari Rajamäki, jari.rajamaki@seinajoki.fi.
WWW: www.seinajoki.fi/teknologiakeskus/

Tietosysteemin rakentamismallista Tietojenkäsittelyn kehittämismalliin

*Terttu Honkasaari,
Tietotekniikan koulutus,
Helia*

Tietotekniikan kehittämisskeskus käynnisti syksyllä 1982 ATK-instituutin aloitteesta projektin, jonka tehtävänä oli kehittää menetelmä- ja apuvälineriippumaton tietosysteemin rakentamisen kehikko. Projektin tehtäväksianto perustui Atk-alan neuvottelukunnan laatimaan tietotekniikan tutkimusohjelmaan. Projektin tavoitteena oli uusia ATK-instituutin rakentamismalli, joka 1970-luvun puolivälissä oli tuotettu ammatillisen atk-koulutuksen yhdeksi runkomateriaaliksi ja samalla myös pohjaksi yritysten systeemityön ohjeistoille. Rakentamismalli oli julkaistu Tietojenkäsittelyliiton kirjasarjassa ja sai kansikuvansa mukaan lempinimekseen ”naula- ja vasarakirja”. 70-luvulla tämän rakentamismallin pohjalta kehitettiin useiden yritysten systeemityöoppaita. Tunnetuin niistä lienee Valtion tietokonekeskuksen opas, joka julkaistiin myös kirjana. Myös uuden 80-luvun mallin piti palvella sekä koulutusta että yritysmaailmaa. Mallin tuli antaa systeemityötä tekeville kehikko, jolla jäsentää tietojenkäsittelyn kehittämistyötä ja myös tämän kehittämistyön kohdetta, tietojenkäsittelyä.

RAKE-projektista TIKEksi

Projektin vetäjänä oli ATK-instituutin tutkimuspäällikkö Terttu Hon-

kasaari ja projektiryhmän jäsenenä tutkijaopettaja Seija Halme ATK-instituutista, erikoistutkija Markku Kopu Valtion tietokonekeskuksesta, suunnittelija Marjatta Maula Kansallis-Osake-Pankista, osastopäällikkö Arto Sainio ATK-instituutista ja tutkija Leena Tikkanen Tietotekniikan kehittämisskeskus ry:stä. Projekti sai nimilyhenteen RAKE, koska tuloksen työnimenä oli RAKentamisKEhikko. Projektin johtoryhmään kuuluivat rehtori Lauri Fontell ATK-instituutista, osastopäällikkö Raimo Anttila Valtion tietokonekeskuksesta, toiminnanjohtaja Henry Haglund Tietotekniikan kehittämisskeskuksesta ja osastopäällikkö Matti Reunanen Kansallis-Osake-Pankista. Projektille koottiin lisäksi tukiryhmä, johon kuului yli 20 toimintayksikön edustajia, projektin eri vaiheissa vähän eri yrityksiä ja eri henkilöitä: ATK-instituutti, A-vakuutus, Cypernetics Oy, DEC, Enso-Gutzeit Oy, Ericsson Programatic Oy, Imatran Voima, Kansallis-Osake-Pankki, Kansa-yhtiöt, OTK, Outokumpu Oy, Posti- ja Telelaitos, Pääkaupunkiseudun Tietokeskus, Rautatiehallitus, Sinebrykoff Oy, SOK, Tampereen yliopisto, Tietojyvä Oy, Tietosavo Oy, Tietotehdas Oy, Tietotekniikan kehittämisskeskus ry, Tilastokeskus, Valtion tietokonekeskus sekä Widenius, Sederholm & Someri.

Projektin työn tuloksena julkaistiin helmikuussa 1984 kirja Tietojenkäsittelyn kehittämien, julkaisijana Tietojenkäsittelyliitto ry. Kirjaan liittyi

myös opetusmateriaali sekä systeemityön peruskäsitteiden että systeemityön tekemisen opettamiseen. Lisäksi julkaistiin moniste Johdatus systeemityön ja siihen liittyvä opetusmateriaali. Heti vuoden 1984 alussa käynnistyi jatko projekti tietojenkäsittelyn kehittämissmallin soveltamisesta sekä vuotta myöhemmin tietojenkäsittelyn kuvausmallista ja kuvaamisen ohjeistosta.

Vuoden 1986 keväällä uudistettiin tietojenkäsittelyn kehittämissmallin peruskirja saatujen kokemusten ja kommenttien perusteella: Tietojenkäsittelyn kehittämissmalli, Käsikirja. Kirjoittajina olivat tässä vaiheessa Seija Halme, Terttu Honkasaari ja Arto Sainio. Tästä kirjasta tuli yllättävän pitkäikäinen ja vielä vuosituhanen vaiheen tietämilläkin siitä on otettu pieniä uusintapainoksia koulutuskäyttöön ja monissa opinnäytteissä se näyttää olevan edelleenkin lähde teoksena. Tämä johtunee ehkä siitä, että kehittämissmalli oli menetelmä-, apuväline- ja kuvauskieliriippumaton ja myös siitä, että kirja määrittelee myös systeemityön peruskäsitteet, jotka ovat kuitenkin olleet melko pysyviä.

Työpaperit
työskentelymuotona:
välitulokset näkyviksi

Työryhmä kirjoitti työnsä taustaksi useita työpapereita vuoden 1983 aikana. Osa työpapereista kirjasi sys-

teemityön nykykäytäntöjen ongelmia, osa sisälsi idearaakileita, osa keräsi tulevaisuuden näkemyksiä, osa oli yhteisten keskustelujen tuloksia kirjavia, osa oli nykykäytäntöjä analysoivia, osa hahmotti projektiryhmässä sovittuja työtapoja. Periaatteena oli kirjata rohkeasti keskustelun pohjaksi kaikki erilaiset näkemykset ja ajatukset. Työpapereita käsiteltiin myös yhdessä tukiryhmän kanssa työseminaarimuotoisesti työskennellen.

Työpapereita kirjoitettiin mm. seuraavilla otsikoilla:

- Rakentamiskehikon soveltamis-ympäristö – skenaario vuosille 1987 – 1992
- Mallien arviointikehikko
- Alustava tarve- ja ongelmakartoitus (nykyisessä rakentamismallissa)
- Rakentamiskehikolle asetettavat tavoitteet ja rajaukset
- Tietosysteemi, sen ympäristö ja sen kehittäminen
- Tuotteiden kuvaukset
- Ongelmia nykyisessä rakentamiskäytännössä
- Uudelle rakentamiskehikolle asetettavia vaatimuksia
- Systeemityön menetelmien perustutkimuksen tilan selvitys Suomessa (yliopistot)
- Lyhyt analyysi CRIS-kartoituksen menetelmien teoreettisesta perustasta ja menetelmäkehityksen suuntaamisesta lähivuosina (IFIP:n menetelmäkilpailun tulokset)
- Eri menetelmien arvioita: MBI, SAK, TISMA, ETHICS, REMORA, CIM, SIDOS, JSD, IE, EDM, SUM, PSC/PIOCO jne.
- Tietosysteemin käsitteistö
- Rakentamiskehikon ja sen käyttöympäristön suuntaviivoja
- Kokonaistutkimus
- Systeemityön apuvälineet

- Apuvälineiden vaikutus systeemityöprosessiin, erilaisia näkemyksiä
- Apuvälineiden vaikutusarvioita
- Rakentamispuitteiden ominaisuuksia
- Näkemyksiä rakentamiskehikon teoriapohjaksi
- Ehdotus käytettävästä käsitteistöstä
- Rakentamiskehikon tehtäväluetelo
- Rakentamiskehikon kuvaus

Tausta kehittämismallille

Näkemyks tietojenkäsittelyn kehittämistyöstä oli ollut 1970-luvun loppuun asti hyvin tietokonekeskeinen ja mielenkiinto malleissa oli keskittynyt atk-systeemien rakentamiseen ja tietokoneen rooliin tietoja käsittelevänä koneena. 1970-luvun lopulla ja 1980-luvun alussa oli eteenkin Pohjoismaissa pyritty lähestymään tietojenkäsittelyn kehittämistyötä myös toiminnan kehittämisen ja ihmisen tietojenkäsittelyn kehittämisen näkökulmista. Kehittämistyö nähtiin tällöin luonteeltaan iteratiivisena ongelmanratkaisu-, oppimis- ja päätöksentekoprosessina. Kehittämistyö tuotti päätöksentekoinformaatiota tietojenkäsittelytyöstä ja kehittämisen vaihtoehtoista. Mukana olevat kehittäjät oppivat myös koko ajan: kehittämistyön alussa mutkikas kohdealue ei ole näkyvässä kokonaan. Siitä opitaan koko ajan lisää työn aikana. Näin ongelmaa ei voida muuttaa kerralla ratkaisuksi. Niinpä kehittämistyö etenee vuoroin määrityksiä täsmentäen, vuoroin osaratkaisuja tehden. Tietojenkäsittelyn kehittämisen lähtökohtana nähtiin nyt olevan organisaation oman toiminnan lisäksi eri sidosryhmien toiminnan kehittäminen. Todettiin, että asiakkaidenkin toiminnan kehittäminen pitää olla yhtenä lähtökohtana ja asiakkaat haluttiin nähdä ja ottaa huomioon myös organisaati-

on tietojärjestelmien suorina käyttäjinä. Tuohon aikaan ei vielä osattu arvata Internetin kehitystä ja sitä myötä asiakkaan suora käyttäjäroolia, mutta ryhmä piti kuitenkin tärkeänä asiakaskäyttäjän roolin mukaan ottamisesta.

Tietojenkäsittelyn ja sen kehittämistyön jäsennys

Tietojenkäsittelyn kehittämismallissa tietojenkäsittelyn kehittäminen on jäsennetty tapahtuvaksi 7 eri tarkastelukulmasta:

- | | |
|---|---|
| K | Toimintayksikön tietojenkäsittelyn kokonaiskehittäminen |
| T | Kohdealueen toiminnan kehittäminen |
| I | Ihmisen toiminnan kehittäminen |
| Y | Tietoyhteyksien kehittäminen |
| V | Tietovarastojen kehittäminen |
| S | Atk-tehtävien sisällön kehittäminen |
| R | Atk-ratkaisun kehittäminen |

Jokaisessa tarkastelukulmassa on eri kehittämistyön kohde. Yksi tarkastelukulmista kohdistuu toimintayksikön koko tietojenkäsittelyyn ja siihen kuuluvat kehittämistehtävät keskittyvät siten organisaation tietojenkäsittelyn strategiseen ja pitkän aikavälin kehittämiseen. Muut tarkastelukulmat kohdistuvatkin sitten rajatumpaan toiminnan ja tietojenkäsittelyn osaan. Tarkastelukulmilla ei ole mitään kiinteää soveltamisjärjestystä toimintaa ja tietojenkäsittelyä kehitettäessä. Niitä kaikkia ei välttämättä edes tarvita kaikissa eri tietojenkäsittelyn kehittämis-tilanteissa. Kehittämistarpeet ja lähtökohdat ovat eri hankkeissa erilaiset, joten tilanne määrää, missä järjestyksessä ja mitä tarkastelukulmia tarvitaan. Jokaiseen tarkastelukulmaan kuuluu kehittämistehtäviä. Jokaisessa kehittämishankkeessa pitää kehittämis-tilanteesta lähtien arvioida, mitä tarkastelukulmia ja mitä tarkastelukulman kehittämistehtäviä tarvitaan sekä

missä järjestyksessä tarkastelukulmia sovelletaan ja missä järjestyksessä kehittämistehtäviä tehdään. Eri tarkastelukulmien tehtäviä voidaan tarpeen mukaan myös yhdistellä, joten yhdessä ”kootussa” tehtävässä voidaan tarvittaessa soveltaa samaan aikaan myös eri tarkastelukulmia.

Tarkastelukulman ja kehittämistehtävän kuvausmalli

Kaikki tarkastelukulmat kuvattiin mallissa yhdenmukaisella tavalla: Kehittämistehtävien lisäksi esitettiin ko. tarkastelukulman kehittämistyön kohde, tavoite, lähtökohta sekä kuvailtiin tarkastelukulman taustaideaa. Kaikissa tarkastelukulmissa oli mukana eri tyyppisiä tehtäviä: kohdealueen rajaamista, selvittämistä, määrittämistä, ratkaisun suunnittelua, ratkaisun toteuttamista, turvaamisen suunnittelua, testaamista. Jokaisen tarkastelukulman jokaisesta tehtävästä kuvattiin tehtävän kohde, lähtökohta, sisältö, mahdolliset osatehtävät ja lopputulos.

Kehittämismalli oli menetelmä-, apuväline- ja kuvauskieliriippumaton, joten näitä ei kehittämistehtävissä ole kuvattu.

Mitä uutta kehittämismallissa

Aiempaan tietosysteemin rakentamismalliin verrattuna uudessa tietojenkäsittelyn kehittämismallissa oli paljon uutta. Rakentamismalli oli katkautunut lähinnä S- ja R-tarkastelukulmien tehtävät. Toiminnan kehittäminen oli siinä rajattu vain kohteena olevaan tietosysteemiin. Nykyisessä toiminnassa olevat kehittämistarpeet tai ongelmat voivat olla lähtökohta toiminnan kehittämiselle, mutta nyt tietotekniikan nähtiin mahdollistavan myös kokonaan uusia toimintamahdollisuuksia. Kokonaan uutta oli ihmisen toiminnan kehittämisen laaja mukaan ottaminen: 70-luvun rakentamismallissa sitä edusti lähinnä yksi tehtävä,

manuaalitoimien suunnittelu. Nyt lähtökohtakin I-tarkastelukulmassa oli toinen: ihminen kokonaisuutena sekä mielekkäät työkokonaisuudet kohdealueella työskenteleville ja hyvät toimintatavat asiakaskäyttäjille - huomioon ottaen inhimilliset tietojenkäsittelyvalmiudet. Tietoyhteyksien kehittämisen tarkastelukulman tehtävistä osa liittyi myös läheisesti ihmisen toiminnan kehittämiseen. Siinä oli mukana myös tietoliikenteen kehittämiseen liittyvät tehtävät. Rakentamismallissa ei ollut vielä näkyvissä yhteiskäyttöisten tietokantojen vaikutusta systeemityöhön; käsiteanalyysin ja tietöanalyysin tuomat näkemykset puuttuivat siis siitä. Rakentamismalli lähti vielä tilanteesta, jossa toimintaa tukemaan kehitettiin ensimmäistä tietosysteemiä, automatisoitiin toimintaa. Rakentamismallin aikaan ei ollut vielä käytössä kehittämiä ja prototyyppien tekeminen ei siten sisältynyt sen maailmaan.

Kehittämismallin käyttö

Kehittämismallin käyttöä eri tilanteissa pyrittiin myös kuvamaan. Kirjaan otettiin mukaan tyyppitilanteina joitakin hyvin eri tyyppisiä tietojenkäsittelyn kehittämistilanteita organisaatiossa, esimerkiksi valmissysteemin hankinta, yhteiskäyttöisen tietokannan kehittäminen, toimiston tietojenkäsittelyn kehittäminen jne. Nämä valitut esimerkkitilanteet kuvattiin sekä selostettiin lyhyesti, mitä kehittämismallin tarkastelukulmia tarvittiin ja mitä kehittämistehtäviä eri tarkastelukulmista sekä missä järjestyksessä ja miten yhdistellen niitä tulisi toteuttaa.

Varsinaisesti systeemityön ohjauksen tehtävät oli selkeästi erotettu kehittämistehtävistä ja kehittämismallin näkemys systeemityön ohjauksesta ja ohjaustehtävät esitettiin liitteessä. Ohjaus jäsennettiin kolmelle eritasolle: systeemityön kokonaisohjaus, kehittämisshanketasoinen ohjaus ja kehittämistehtävätasoinen ohjaus.

Koska kirja tarkoitettiin systeemityön ammattilaisten koulutuksen perusteokseksi, kirjan alkuosa keskittyi systeemityön ja systeemityön kohteena olevan tietojenkäsittelyn käsitteiden määrittelyyn ja käsitteiden välisten yhteyksien selvittämiseen ja pohtimiseen. Käsiteosuus kattoi koko kehittämismallin alueen, myös ihmisen tarkastelun tietojenkäsittelijänä. Samoin systeemityön ohjaukseen liittyvät käsitteet oli mukana.

Miten malli syntyi

Näin vuosien jälkeen on vaikea muistaa tarkasti, mikä ratkaisevimmin vaikutti kehittämismallin muodostumiseen sellaiseksi, mikä siitä tuli.

Taustalla oli joka tapauksessa monien silloisten menetelmistöjen selvitys. Piirteitä poimimme monista lähteistä ja kokemuksi ja näkemyksiä keräsimme useista organisaatioista. Jokainen työssä mukana ollut vaikutti omilla taustoillaan lopputulokseen.

Keskusteluissamme emme pysyneet vain tietojärjestelmätieteen, systemoinnin, systeemiteorian alueilla. Selvitimme myös keskusteluin systeemityöhön vaikuttavia muita tieteen aloja, teorioita ja niiden näkemyksiä. Työpapereista löytyy mm. yhteenve-toja seuraavien alojen vaikutuksista tietojenkäsittelyyn, systeemityöhön ja sen ratkaisuihin: psykologia, kielitieteet, fysiologia, ergonomia, työtieteet, työpsykologia, filosofia, logiikka, etiikka, semiotiikka, hallintotieteet, valtiotieteet, politiikka, sosiologia, sosiaalipsykologia, didaktiikka, viestintä, taloustieteet, insinööritieteet.

Mallien analysointikehikon loimme, jotta pystyimme helpommin vertaamaan erilaisia malleja. Jokaisen tutkimamme mallin systeemityönäkemysmuodostamiseksi kokosimme luettelon rakentamistehtävistä, laadimme yhteysmatriisin, jossa näkyi raken-

tamistehtävät ja niiden tuottamat tulokset ja kuvaukset sekä piirsimme prosessikuvauksen systeemityön etenemisestä matriisin pohjalta. Prosessikuvauksesta ei saatu aina täydellistä, sillä mallin selostus ei välttämättä esimerkiksi kertonut kaikkien syntyneiden kuvausten käyttöä jatkotyössä. Mallin näkemys systeemityön kohdesta muodostettiin kuvaamalla kohdesysteemin käsitelmä: luetteloidimme kaikki tietosysteemiä ja sen kohdesysteemiä kuvaavat käsitteet, paikallistimme ja kuvasimme käsitteiden väliset riippuvuudet ja yritimme tältä pohjalta koota yhtenäisen käsitelmän ”tervettä järkeä käyttäen”. Muita tarkkailtavia seikkoja olivat mm. käyttäjän rooli, kanta sovellusalueeseen, kanta ohjaustehtäviin, viittaukset työmenetelmiin, rajausten toimintayksikön toiminnan analysointiin, kanta käyttöönoton suunnitteluun. Jokaista mallia pyrimme myös vertaamaan Tietosysteemin rakentamismalliin. Noita työpapereita nyt katsoessa tuo analysointi näyttää perusteelliselta ja varmasti se myös opetti meille paljon.

Systeemityön menetelmien perustutkimusta kartoitimme yliopistoille tehdyllä kyselyllä, johon vastasivat Turun, Oulun, Helsingin, ja Jyväskylän yliopistot, Teknillinen korkeakoulu sekä Helsingin ja Turun kaupunkorakentamiskoulut.

Yliopistojen tutkimuskohteet liittyivät tuohon aikaan mm. ihmiskeskeiseen systemointiin, automaattisiin apuvälineisiin, menetelmien vertailuun ja valintaan, toimistoautomaatioon, tietohakemistoihin, systeemityön liittämiseen osaksi organisaation toimintaa, integroitumiseen organisaatioteoriaan. Yleisesti useat vastaajat totesivat empiirisen tutkimusotteen lisääntyvän. Tästä tilanneselvityksestä emme juurikaan saaneet tukea kehitettävälle mallille.

IFIP:n menetelmäkilpailun (CRIS) tulokset julkaistiin sopivasti juuri projektimme käynnistyessä. Kilpailuun osallistuvien piti kuvata menetelmällään yksi annettu tietosysteemi. Se helpotti menetelmien vertaamista. Arviointia ryhmällemme oli laatiassa menetelmäkilpailun tuomaristotyöhön osallistunut Hannu Kangasalo Tampereen yliopistosta. Kilpailuun osallistuneiden menetelmien lähtökohtana todettiin olevan jokin tai useampi seuraavista: Software Engineering, systeemiteoria, psykologia, tiedonhallinta, kielitiede, tietojenkäsittelytiede (Computer Science), tekoäly, filosofia, tietoteoria, kommunikaatioteoria. Menetelmällä voitiin pyrkiä hallitsemaan erityisesti jotakin seikkaa, kuten käyttäjien osallistumista, tietokonetukea, evolutionääristä suunnittelua, käsitteellistä mallintamista, prototyyppikehittämisen käyttöä, tiedonhallintaan sopeutumista, toimistoautomaation integrointia, systeemi-dynamiikan kuvaamista jne. Teoreettisen perustan analyysissä kiinnitettiin huomio lopulta neljään seikkaan: Soveltaako menetelmä yhtä vai useampaa ajatusmallia tai teoriaa? Missä suunnittelun vaiheessa tai vaiheissa ajatusmallia tai teoriaa sovelletaan? Mikä kunkin sovelletun ajatusmallin tai teorian kehitysaste on? Millaisia ajatusmallit ovat rakenteeltaan, ts. mitä käsitteitä ja käsitteiden välisiä suhteita ne sisältävät ja millaisia säännönmukaisuuksia niihin sisältyy?

Tämän selvityksen perusteella pystyimme tekemään yhteenvedon kansainvälisen menetelmäkehityksen yleissuunnista.

Projektiryhmä ja tukiryhmä edustivat erilaisia taustoja ja organisaatioita. Tämä oli työssämme rikkaus. Yhteistyö oli avointa, keskustelevaa ja yhteistä tulosta jatkuvasti rakentavaa. Keskustelimme paljon, kirjoitimme paljon ja piirsimme paljon. Koimme

työilmapiirimme turvalliseksi, joten ideointi toimi. Ideat myös käsiteltiin. Ongelmat uskalsimme myös nostaa esille, kirjoittaa työpapereihimme. Uskalsimme kyseenalaistaa vallitsevia käsityksiämme, heittäytyä uusien ideoiden mukaan. Tapamme työskennellä ja hyvä työilmapiirimme vaikuttivat varmasti hyvään lopputulokseemme. Yhteistyö oli tässä projektissa todellakin voimaa. Näin vuosien perästäkin haluan vielä kiittää kaikkia mukana olleita yhteistyöstä projektissamme, joka tuotti halutut tulokset ja ajallaan.

Lähteet

Halme, Honkasaari, Kopu, Maula, Sainio, Tikkanen: Tietojenkäsittelyn kehittäminen, Tietojenkäsittelyliitto ry. 1984.

Halme, Honkasaari, Sainio: Tietojenkäsittelyn kehittäminen, Tietojenkäsittelyn kehittämissä, Käsikirja. Tietotekniikan liitto ry. 1987.

Halme: Johdatus systeemityöhön. ATK-instituutti – Valtion painatuskeskus. 1986.

Honkasaari, Huuskonen, Ripatti, Roukala, Soini, Tikkanen: Tietojenkäsittelyn kehittäminen, Tietojenkäsittelyn kuvausjärjestelmä, Tietotekniikan kehittämissä ry. 1986.

Kopu, Kuivalahti, Maula: Tietojenkäsittelyn kehittäminen, Kehittämishankkeen suunnittelu. Tietotekniikan kehittämissä ry. 1985.

ATK-instituutin työryhmä: Tietosysteemin rakentamismalli. Tietojenkäsittelyliitto ry. 1975.

*Terttu Honkasaari,
Johtaja,
Tietotekniikan koulutus,
Helia*

Systeemityön käsitteen ja sisällön varhaishistoriasta

Eero Kostamo

Systeemityöhön liittyvät tehtävät alkoivat joskus vuoden 1957 aikana. Toimin silloin Helsingin kaupungin hallituksen järjestelytoimiston toimistotyöntutkijana. Organisointi- ja rationalisointiselvitysten ohella tehtäviini alkoi yhä useammin tulla laskentatoimien järjestelmien koneellistamista kirjapitokoneilla. Ei siinä puhuttu systeemityöstä, mutta tosiasiallisesti suunniteltiin tietojenkäsittelyn systeemejä – sen ajan laitteita hyödyntäen.

Oivalsin melko pian, että tietojenkäsittelyn tulevaisuus tulee rakentumaan tietokoneiden varaan. Tosin silloin puhuttiin elektronisista tietojenkäsittelykoneista tai matematiikkakoneista. Median taholla uutta ilmiötä nimitettiin ”sähköaivoiksi”. Ammatti-lehtien lisäksi tietoutta näistä uusista laitteista levitti lähinnä IBM, joka järjesti ohjelmointikurssejakin. Ne jouduttiin käymään ”kuivaharjoitteluna” niin kauan kuin IBM:lläkään ei ollut yhtään tietokonetta Suomessa.

Varmistaakseni mukaanpääsyn näköpiirissä olevaan tietojenkäsittelyn uuteen aikakauteen hain ASLA-stipendiä jatko-opintoihin USA:ssa. Sain sellaisen sekä virkavapautta Helsingin kaupungilta ja niin ilmoittauduin syyslukukauden 1958 alkaessa University of Michiganin Business Schooliin. Sieltä löytyi oikea tietokonekin, IBM 650, ja pääsin nyt testaamaan tekemiäni ohjelmia muutenkin kuin kuivaharjoitteluna.

Yliopiston lukuvuoden päätyttyä tein kesällä 1959 opintomatkan USA:sta löytyviin julkisten laitosten tietojenkäsittelykeskuksiin. Arkistostani löytyneen matkakertomuksen mukaan kävin 14 keskuksessa. Matkakertomuksessa on pätkä tekstiä, josta ilmenee silloinen käsitykseni systeemityöstä:

”Suunnittelu- ja ohjelmointityössä voidaan erottaa kolme eri tehtäväryhmää seuraavasti:

1. Systeemin suunnittelu
2. Ohjelmointi
3. Koodaaminen

Systeemin suunnitteluun kuuluu nykyisen menetelmän yksityiskohtainen tutkiminen, uudesta järjestelmästä otettavien lopputulosten asiallisen tarpeen ja sisällön määrittely, integrointimahdollisuuksien selvittäminen, kontrollisen luominen, eri konejärjestelmiä koskeva vaihtoehtojen järjestelmien valinta ja kannattavuuslaskelmat sekä lopullisen systeemin laatiminen yhdistämällä aineiston käsittelyyn osallistuvien henkilöiden, osastojen ja koneiden työ saumattomaksi kokonaisuudeksi.

Ohjelmointi käsittää päätetyn systeemin mukaisesti EDP-koneelle kuuluvan osan analysoimisen ja konetyön suunnittelun koneen työskentelylogiikan edellyttämällä tavalla. Ohjelma esitetään ohjelmakaaviona (block-diagram).

Koodaaminen on em. ohjelmakaavion ohjelma-askeleiden kääntämistä tietyn EDP-koneen konekielelle. (Myös tätä vaihetta kutsutaan usein ohjelmoimiseksi.)”

Tavoitteeni päästä tietotekniikatoihin toteutuikin pian kotimaahan paluuni jälkeen. Minut nimitettiin järjestelytoimistossa EDP-suunnittelijaksi. Seuraavana vuonna (1960) perustettiin Helsingin kaupungin tietojenkäsittelykeskus ja sain nimityksen sen suunnittelupäälliköksi. Ryhdyin innolla hoitamaan ensimmäistä esimiestehtävääni ja kun mitään valmiita toimintamalleja ei ollut, laadin toimintaamme varten *suunnitteluosaston toimintaohjeet*. Olikin hauska kokemus löytää nyt vanhoista papereistaan tällainen kesäkuulle 1961 päivätty dokumentti.

Yleisluontoisten organisaatio- ja vastuunjakotekstien jälkeen olin jäsenellyt toimintaohjeet suunnitteluvaiheeseen, koeajovaiheeseen, toteuttamisvaiheeseen, systeemien huoltoon ja dokumenttien arkistointiin. Suunnittelun sisältöä kuvaa seuraava tekstioite:

”...Lisäksi on suunnittelupäällikölle esiteltävä sekä luonnosvaiheessa että valmiina:

- 1) Systeemiselostukset mahdollisine työnkulkukaavioineen
- 2) Reikäkorttikaaviot
- 3) Kentällä käytettävät lomakkeet
- 4) Tietojenkäsittelykeskuksessa valmistettavat lomakkeet
- 5) Ohjelman yleiskaavio.”

Nuorempi polvi saattaa ihmetellä, että mitä ovat reikäkorttikaaviot? Ajoja reikäkorttikoneilla tarvittiin, koska tk-keskuksen ensimmäisessä tie-

tokoneessa, IBM 1401, ei ollut tiedostojen tallettamista varten magneettinauhoja tai levymuistia. Tiedostot oli talletettava reikäkortteille. Samoin tahtumatiedot syötettiin koneelle reikäkortteina.

Toimintaohjeiden eräänä liitteenä oli *työmäärien arviointilomake*. Siinä oli eritelty noin 30 työvaihetta tai kohdetta. Esimerkiksi ohjelmoinnin työmääräarvio oli laskettava seuraavalla erittelyllä:

- 1) Pääkaavio
- 2) Detaljikaaviot
- 3) Koodaus
- 4) Koeaineiston laatiminen ja laskeminen
- 5) Pöytä tarkastus
- 6) Testaus

Arkistostani löytyi myös suunnitteluosaston *toimintaohjeiden lisäys*, joka on päivätty n. kolme kuukautta edellä selostetun dokumentin tekemisen jälkeen. Sen parista ensimmäisestä kappaleesta heijastuu silloisen systeemyön oppimisprosessia, joten liitän nämä kappaleet tähän sellaisinaan:

”Ennenkuin koneosasto voi ottaa jatkuvasti suoritettavaksi tietojenkäsittelytehtäviä, täytyy näistä olla täydelliset työohjeet. Erikseen mainittavia poikkeuksia lukuunottamatta nämä ohjeet laatii suunnitteluosasto. Työohjeiden laatiminen kuuluu osana suunnitteluosaston dokumentointityöhön, joka kokonaisuudessaan käsittää seuraavat alueet:

- 1) suunnittelun oma dokumentointi
- 2) koneosaston työohjeet
- 3) valvontaryhmän työohjeet
- 4) kenttähenkilökunnan työohjeet.

Aikaisemmin annetut ohjeet ja dokumentointiohjeet tullaan yhdistämään

yhtenäisiksi toimintaohjeiksi. Kiireellisenä annetaan vielä erikseen ohjeet koneosaston työohjeiden laatimisesta.”

Käsite *systeemyö* on tullut tietoisuuteeni ilmeisesti joskus vuosina 1962 tai -63. Päättelen tämän siitä, että tuo käsite esiintyy, tosin muodossa ’systemarbete’, ensimmäisen kerran omissa teksteissäni vuonna 1963 ilmestyneessä kirjassa *Automaattisten tietojenkäsittelysysteemien suunnittelu*. Olin tuon kirjan toimittaja ja kirjoitin sen alkuun käsitteiden määritelmät. Systeemin suunnittelun tarkasteluun käytin pari sivua. Seuraava ote kertoo sen aikaisen ymmärrykseni systeemyöstä.

”...Huomaamme siis, että [systeemi]analyysi on täysin riittämätön kattamaan yksinään kaikki systeemin aikaansaamiseen tarvittavat toimenpiteet. Toisaalta ei myöskään systeemin suunnittelu kuvaa oikealla tavalla niitä tehtävien osia, joissa pääpaino on tutkimisella ja erittelyllä. Täsmällisyyden saavuttamiseksi pitäisi puhua systeemianalyysistä ja –synteesistä, mikäli näitä sanoja halutaan käyttää, tai systeemin tutkimisesta ja suunnittelusta.

Ruotsinkielessä tämä ongelma on ratkaistu ainakin eräässä teoksessa (Matematikmaskinnämnden: *ADB-kurs, Systemarbete*, 1962) käyttämällä nimitystä ’systemarbete’, johon on katsottu sisältyvän sekä ’systemanalys’ että ’systemutformning’. Samoin on amerikkalaisilla termi ’systems engineering’, joka sisältää sekä käsitteet ’systems analysis’ ja ’systems design’.

Koska suomenkielessä ei ole sopivaa yhteisnimitystä ja koska toisaalta kolmesta sanasta muodostettu nimitys jatkuvassa kielenkäytössä on hankala, on

tämän julkaisun piirissä päätetty käyttää sanaa systeemin suunnittelu sekä systeemin analysointi- että kokoonpanovaiheista.”

Käsite *systeemyö* tuli osaltani käyttöön, kun olin ollut jo jonkin aikaa valtion tietokonekeskuksen palveluksessa vetämässä sen suunnittelupuolta. Suunnitteluosaston johtotiimissä pohdimme tietenkin menetelmiä ja toiminnan systematisoimista mukaan lukien peruskäsitteet, joita käytetään. Systeemyö käsitettä ajoi erityisesti edesmennyt fil.tri Yrjö Seppälä. Hän perusteli sitä sillä, että systeemin aikaansaamiseen sisältyi niin monenlaisia töitä, että käsitteen tulisi olla mahdollisimman yleinen. Seppälän perustelut hyväksyttiin ja niin alettiin vtikk:n suunnitteluosastolla tehdä systeemitöitä.

Kun 1973 siirryin johtamaan pankkien yhteistä on-line maksuliikenneprojektia (POLM), huomasin eri pankkien käyttävän systeemyöstä erilaisia ilmaisuja. Pankkien yhteishankkeissa koetimme käyttää täsmällisiä termejä, mutta en muista, että systeemyö-käsitteestä olisi keskusteltu. Puhuimme ehkä useimmiten *järjestelmäkehityksestä*.

Termit ovat muuttuvaisia. Perusilmiön juurikaan muuttumatta olen huomannut, lyhenteinä ilmaisten, tietotekniikassa seuraavan kehityskaaren: EDP, ETK, ATK, Atk, atk, ICT, IT, It, it. Kaikkien niiden toteuttaminen on vaatinut systeemyötä.

Eero Kostamo

SRP-menetelmistö

Veikko Roukala

Menetelmistön synty

SRM-menetelmistö sai alkunsa kesällä 1986, kun menetelmän perustajat, kirjoittaja, Soini ja Mikkonen lähtivät IBM:stä ja perustivat Tietosuunnittelu-SRM Oy:n. Kaupparekisteriin ei hyväksytty nimeä ”Tietosuunnittelu Oy”, koska käänteisnimi oli olemassa, vaan Kaupparekisteri ehdotti irtokirjainten lisäämistä nimen perään. SRM tarkoittaakin ”Systeemien RakentamisMalli” tai ”Systems Resource Management” tai ”Soini-Roukala-Mikkonen”.

Menetelmistön mukaisia asioita oli toki olemassa aikaisemminkin. Jo 70-luvulla Markku Mikkonen konsultoi IBM:n asiakkaita järjestämällä ”BSP” (Business Systems Planning)-projekteja. Ne saivat liiketoiminnan ja tietohallinnon johdon edustajat innostumaan ja projektit toteutettiin johdonmukaisesti. Niiden tulokset jäivät kuitenkin useimmiten pölyyntymään hyllyihin. Tuloksissa ei ollut konkreettisia hankesuunnitelmia jatkotyön järjestämiseksi.

Tapani Soini tuli mukaan kuvioihin 1980-luvun alkupuolella ja toi projekteihin kohdemallin laatimistekniikan. Aikaisempien tietoluokkien sijaan ristiintaulukoitiin relaatiomallin mukaiset tietokohteet (kohderyhmät) yrityksen toimintomallin kanssa. Näin saatiin määritettyä tietojärjestelmät, jotka priorisoitiin. Tärkeimmälle tehtiin ns. toteutuksen suunnittelu, jonka tuloksena saatiin konkreettiset jatkoprojektit. Soini kirjoitti myös kirjansa ”Tietöanalyysi” vuonna 1984 (W&G).

Kahdeksankymmentäluvun alussa Englannissa professori Eason kuvasi systeemityömenetelmien kehittymisen ja tavoitetilan kolmena vai-

heena. Hänen mukaansa kuusikymmentäluvulla ja myöhemminkin käytettiin atk-keskeistä systeemityömenetelmää. Siinä atk-ammattilaiset haastattelivat käyttäjiä ja laativat saamansa tiedon perusteella atk-ohjelmat, joihin käyttäjät ottivat kantaa vasta testaamisen yhteydessä: ”Enhän minä tätä tarkoittanut” (vaihe 1).

Seitsemänkymmentäluvun lopussa syntyi käyttäjäkeskeinen systeemityö. Siinä suunnitteluprojektiin ”otettiin mukaan” käyttäjiä. Enemmistö tai vähintään puolet projektijäsenistä oli kuitenkin atk-ammattilaisia. Kehittämistapa oli edelleenkin atk-lähtöinen ja projektissa puhuttu kieli atk-slangia – käyttäjät eivät päässeet vieläkaan vaikuttamaan haluamallaan tavalla. Yhä tehtiin määrittelyä ja suunnittelua (vaihe 2).

Tavoitteena Eason kuitenkin piti oppimiseen perustuvaa toimintaa kehittävää systeemityötapaa: kehittäminen menetelmä ”puhuu” toiminnan tuntioiden (käyttäjien) kieltä ja projekteissa suunnitellaan (liike)toiminnan kehittämistä (vaihe 3). Projektissa on mukana on yksi systeemisuunnittelija, joka oppii toiminnan ja kuvaa asiat myös tietotekniikkaratkaisun edellyttämällä tavalla. Erään asiakkaan pro-

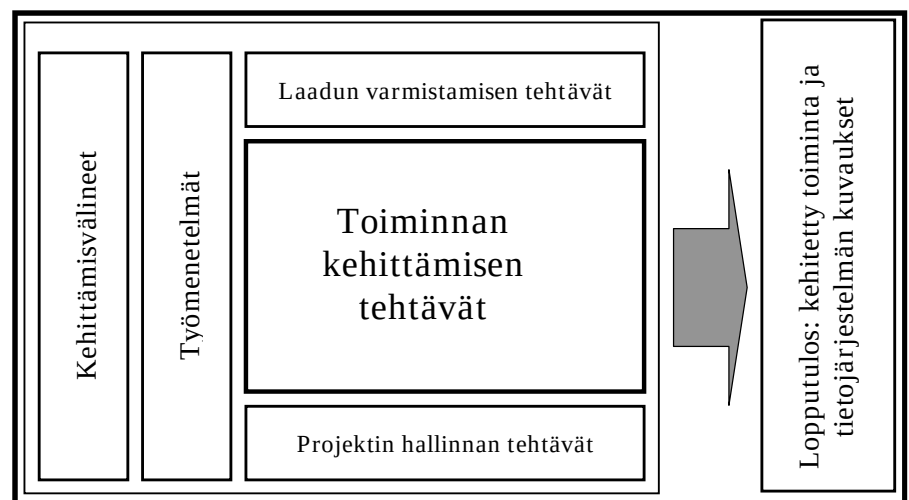
jektipäällikkö, jolla oli kokemusta ”vanhoista” kehittämistavoista, totesikin riemastuneena, että nyt tuo atk-ammattilainen saa kuunnella meidän ”slangiamme”.

Uuteen yritykseen tuli mukaan myös allekirjoittanut. Kirjani ”Kehittämiprojektien laadun varmistaminen” oli juuri tullut painosta (1986, W&G). Kirja sisälsi projektin hallinnan, laatukatselmuksia, projektien tarkastamisen ja toimintopisteanalyysin (FPA).

Menetelmistön käytön kasvu ja vakiintuminen

Kahdeksankymmentäluvun lopulla menetelmä ohjeistettiin tietojärjestelmätutkimuksesta systeemityöhön, atk-suunnitteluun, Case-välineen käyttöön (IEW, ADW) ja tietojärjestelmän käyttöönottoon. Ohjeiden yhteenlaskettu sivumäärä oli noin tuhat. Yhdessä projektissa toki tarvittiin yleensä kuitenkin vain yhtä suppeaa ohjetta.

Juhani Ellä ja Timo Malkki tulivat mukaan yhtiöön konsultoimaan, kouluttamaan ja kehittämään menetelmistöä edelleen. Jouko Ikäheimonen valittiin toimitusjohtajaksi ja vastaa-



Kuva 1

maan myynnistä. Markku Mikkonen ja Tapani Soini kirjoittivat kirjan ”Yrityksen tietosuunnittelu” 1986 (W&G) ja allekirjoittanut Soinin kanssa kirjan ”Tietojärjestelmien rakentaminen: toimintaa kehittävä systeemityö” 1988 (W&G). Tässä vaiheessa yritys hankki myös uudet koulutus- ja toimitilat Pieni Robertinkatu 11:sta pankin innoittamana ja puoleksi rahoittamana.

Menetelmistö koostuu:

1. Kehittämisen tehtävistä (systeemityömallista), joka sisältää
 - strategisen ja taktisen kehittämisen
 - toiminnan kehittämisen (systeemis suunnittelu)
 - atk-suunnittelun ja toteutuksen
 - valmisohjelmiston valinnan ja toiminnan muutoksen toteutuksen (tietojärjestelmän käyttöönotto).
2. Projektinhallinnan tehtävistä
3. Laadun varmistamisen tehtävistä
4. Edellisiin käytettävistä työmenetelmistä sekä
5. Tehtäviin ja työmenetelmiin soveltuvista kehittämisvälineistä.

Kehittämisen tehtäviä kutsutaan usein malliksi. On myös projektityömalleja ja laadun varmistamisen malleja. Työmenetelmät puolestaan ovat riippumattomia malleista. Ne kehittyvät ja muuttuvat omaan tahtiinsa, samoin kuin välineet. Esimerkiksi toimintaa voidaan kuvata hierarkkisena mallina, prosessina eri tavoin, vuokaaviona tai lävistäjämatriisina. Tämä tarkoittaa, että ohjeistuksessa on menetelmistön osat pidettävä erillään. Vieläkin törmää sellaisiin malleihin joissa ainakin projekti- ja kehittämistyö sekä työmenetelmät ovat sekaisin.

Menetelmistö –termiä on kuvannut myös Anders G. Nilsson. Häntä mukaillen SRM-menetelmistön sisältö ja edut ovat:

1. Liiketoimintalähtöisyys, joka sisältää päämäärät, tarkoituksen ja arvot: kehittämishankkeiden tulee lähteä toiminnan tarpeista, unohtamatta, että tietotekniikasta voi myös lähteä kehittämisimpulsseja.
2. Teorioihin ja malleihin perustuminen: SRM pohjautuu relaatio-teoriaan (se on lahja yritysjohdolle liiketoiminnan kohteiden kuvaamiseksi ja ymmärtämiseksi; johto tosin ei ole mitään lahjaa huomannut saaneensa), talusteorioiden ja laatujohtamisen malleihin.
3. Kehittämistyön jakaminen vaiheisiin (malli ja työmenetelmät; kuvan 1 mukaisesti – koko elämänpäivä).
4. Ei jälkidokumentointia: työskentelytapa on johdettu seinätaulumenettely. Suurlehtiöitä ja viestilappuja käyttäen pienen huoneen seinät ja ikkunat peittyvät muutamassa viikossa kuvauksiin, jotka projektijäsenet tekevät innostuneina yhdessä mallin ohjeiden mukaisesti, kukaan ei jää passiiviseksi. Projektin atk-jäsen voi tallettaa tietoja jo projektin aikana tai heti sen jälkeen sopivalle välineelle.
5. Välineiden käyttö (suutarin lapsen kengät): mallin ja työmenetelmien mukaisten kuvausten tekeminen ohjeistettiin IEW ja ADW –case –välineille. Myös muita välineitä voidaan käyttää.

Mielestäni kuitenkin menetelmistön oleellisin tarkoitus on toimia yhteydenpitovälineenä: saadaan aikaan yhteinen näkemys ja virheetön lopputulos (mitä myöhemmin kehittämiskäskäsen aikana virhe havaitaan, sitä kalliimmaksi tulee sen korjaaminen) innostavasti ja ideoivasti.

SRM-menetelmistön kaikkia tai joitakin osia on sovellettu monissa eri yrityksissä usealla toimialalla vuosina 1986 lähtien: teollisuus ja jakelu, pank-

ki, vakuutus, eläkevakuutus, terästeollisuus, kauppa, julkinen hallinto, kunnallishallinto, monialateollisuus, ilmainen, atk-palvelu, meriliikenne, raideliikenne, tele ja puhelin, öljy, metsäteollisuus, varastointi ja radiotoiminta.

Sattumuksia ja tilanteita

Kuudentoista vuoden aikana sattui monia mieleen jääviä tilanteita. Tässä pari esimerkkiä parhaasta mahdollisesta kiitoksesta konsultille.

Erään Espoon kaupungin projektin päätyttyä tuli kaksi projektijäsentä sanomaan minulle, että olin ensimmäinen heidän tapaamansa konsultti, jota ei tarvinnut pelätä.

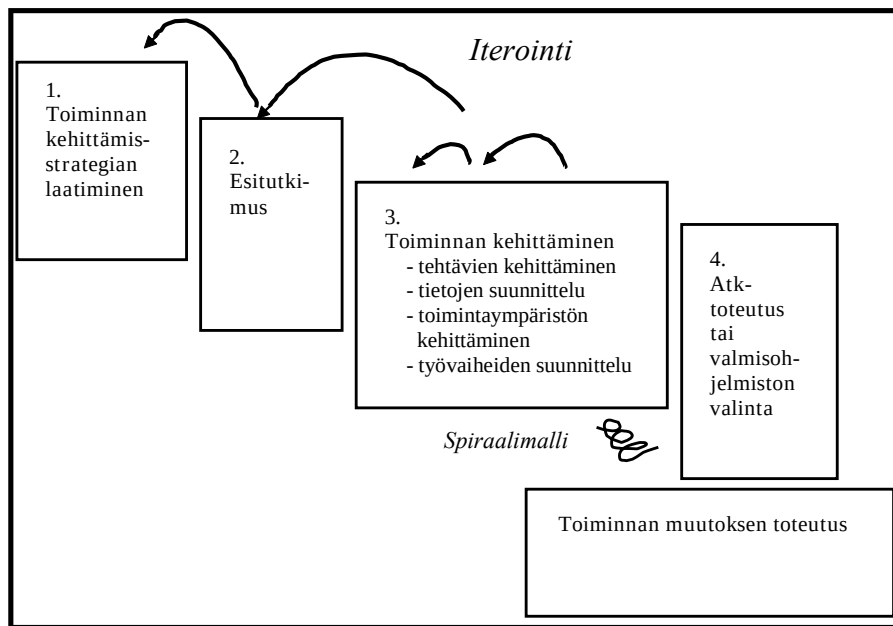
Eräs tietohallintopäällikkö puolestaan totesi minulle tietojärjestelmätutkimuksen päätyttyä oppineensa työnantajansa toiminnasta viimeisen kolmen kuukauden aikana enemmän kuin aiempien kolmen vuoden aikana yhteensä.

Kanssani työskennellyt asiakasni systeemis suunnittelija meni töihin suureen atk-taloon. Puolen vuoden kuluttua hän soitti minulle ja kertoi, että menetelmäni oli paljon parempi kuin se, mitä hän nyt joutui käyttämään.

Toiminnan taantuminen

Yhdeksänkymmentäluvun alussa koko Suomen joutuessa lamaan, eivät yritykset enää investoineet ulkopuoliseen konsultointiin ja koulutukseen. Vuonna 1992 pankkituki oli jo olemassa, eikä Pankki suostunut vain korkojen maksuun, vaan vaati myös lyhennykset.

Yritys hajosi. Soini oli jo muutamaa vuotta aikaisemmin perustanut oman yrityksensä. Vielä nelisen vuotta sitten hän vaikutti Uudessa Seelannissa. Osa liittyi Ernst & Young Consulting:iin ja allekirjoittanut perusti



Kuva 2.

SRM-kehitys Oy:n. Jatkoin siinä entistä työtäni vuoteen 2002 asti.

Kuva 2 esittää SRM-menetelmistön työvaiheita. Laatuguru Deming sanoi, että yritys on systeemi, jonka jokainen komponentti ja niiden väliset riippuvuudet on tunnettava. Toiminnan kehittämisstrategian tuloksena on mm. malli koko yrityksen toiminnasta ja sen kohteista sekä niiden välisistä riippuvuuksista. Ne toiminnot jotka luovat ja päivittävät samoja liiketoiminnan kohteita muodostavat yrityksen pääprosessit. Osa niistä voidaan nimetä ydinprosesseiksi. Toimintojen tarvitsemat resurssit ja vastuut ovat kolmas komponentti, joka osoittaa prosessien, tietojen ja yhteyksien omistajuudet.

Esitutkimus puolestaan tehdään tärkeimmäksi kehittämiskohteeksi asetetulle pääprosessille. Se tarkentaa toimintoja, liiketoiminnan kohteita ja vastuita sekä osoittaa osaproessit. Sen tuloksena laaditaan myös projektisopimukset kehittämistyön lähtökohdiksi. Näin syntyneet projektit ovat kooltaan pieniä: suunnittelu ja toteutus yleensä alle puoli vuotta.

Toiminnan kehittämisen tekevät sen tuntijat (eivät käyttäjät; vrt. am-

mattinimikkeet vs. alkoholin ja huumeiden sekä atk:n käyttäjät). Systeemi-suunnittelija tuo IT-tietämyksen projektiin.

Toiminnan muutoksen toteutus (entinen tietojärjestelmän käyttöönotto) varmistaa muutoksen toteutuksen. Sen aikaa vievin osuus on yleensä testaaminen ja koulutus. Kirjoitin aiheesta kirjan vuonna 1998 (Satku).

Muutoksen hallinta tarkoittaa mm. johdon toimia, joilla edelliset asiat tehdään – johto antaa resurssit ja aikaa kehittää omaa toimintaansa ja omaksua uudet toimintatavat.

Yhdeksänkymmentäluvulla SRM:n selkeä tavoite oli yhdistää toiminnan, tietojärjestelmien ja laadun kehittäminen. Eräässä 80-luvun lopun tietojärjestelmien kehittämisen strategiaprojektissa tehtiin havainto, että tapahtui päällekkäistä työtä, kun tämä asiakas kehitti samanaikaisesti laatu-järjestelmää ja eräälle liiketoiminnan osa-alueelle toimintaa ilman atk-järjestelmän mukanaoloa. Henkilö, joka oli mukana kaikissa kolmessa projektissa totesi, että tietojärjestelmä tutkimuksen tulokset olivat selkeämpiä ja tarkempia kuin muiden projektien tulok-

set: pitää kuvata tarkkaan, koska tietokone on idiootti: ymmärtää vain nollan ja ykkösen.

Allekirjoittanut on jatkanut menetelmistön koulutusta ja konsultointia 2000 –luvulle. Opetan Heliassa yhtenä opettajana kolmesta opintojaksoa ”Tietotekniikka liiketoiminnan tukena”, aiheenani toiminnan kehittämisstrategian laatiminen sekä ”Tietojärjestelmän käyttöönotto” ja ”Valmisohjelmiston hankintaa” muiden yritysten kouluttajana.

Nykytilanne

Säästötoimet ja toiminnan ”amerikkalaistuminen” (pörssi, lyhytjänteisyys) sekä ulkoistaminen ovat mitätoineet monessa yrityksessä menetelmistöjen merkityksen. Jatkuva kiire saa aikaan sen, että enää ei ole kunnon projekteja: asioita tehdään oman toimen ohella silloin kuin ehditään. Paras asiantuntija ei ehdi lainkaan kehittämään yrityksen toimintaa.

On palattu Easonin vaiheisiin 1 ja 2. Ulkoistettu tietohallinto on toimitaja muiden joukossa. Emoyhtiöihin on kuitenkin syntynyt uusi osittain ehkä turhakin joukko – osastojen sisäiset it-yksiköt. Tiedon kulku ja tekeminen ei välttämättä ole jouhevaa – ei ole aina määritetty yhteistä prosessia: toimitaja – sisäinen atk - asiakas.

Ratkaisuja tilataan monelta toimittajalta, joka on lyhytnäköistä kilpailuttamista. Ei ole enää toiminnan kehittämisen menetelmistöä. Ohjelmistotaloilla on omat vaatimusmäärittelynsä, kuvaustapansa ja muut atk-keskeiset näkökulmansa. Yritysten toiminnan ja tietojärjestelmien kehittämistavat voivat olla monenlaisia ja keskenään erilaisia, jopa ristiriitaisia. Tätä ristiriitaisuutta voidaan kutsua epäsi-kiömalliksi.

Veikko Roukala,
veikko.roukala@my.helia.fi

Tietoyrityksen johtaminen - Case Tietotehdas

*Tapani Seppänen,
HM&V Research Oy*

Tietotehtaan kasvu ja toiminnan muutokset 1970- ja 1980-luvuilla kuvaavat hyvin tietojenkäsittelyyn ja systeemityöhön keskittyneen atk-palveluyrityksen kehitystä Suomessa.

Tietotehdas perustettiin Yhdyspankin ja sen lähipiiriin kuuluvien suuryritysten atk-osastoista vuonna 1968. Voimavarojen keskittämisellä pyrittiin parantamaan silloin niukkojen atk-resurssien hyväksikäyttöä sekä luomaan paremmat edellytykset tietoteknisen osaamisen kehittymiselle. Alusta lähtien oli myös tavoitteena luoda Tietotehtaasta liiketoimintaperiaatteilla toimiva atk-yritys, joka myy palvelujaan myös osakaskuntansa ulkopuolelle.

Seuraava kuvaus perustuu syksyllä 1990 kirjoitettuun tekstiin, joka oli pohjana HM&V Research Oy:n julkaisemassa kirjassa Tietoyrityksen strategiat, johtaminen ja menestyminen 1990-luvulla. Kuvaus ei sisällä yhtiön kehitystä ja toiminnan muutoksia 1990-luvulla.

Asiakkaan sitominen

Osakkaiden osuus Tietotehtaan liikevaihdosta oli luonnollisista syistä 1970-luvun alkupuolella lähes 100%. Yhdyspankki oli suurin asiakas sekä tietokonekeskukselle että systeemityötä tekeville yksiköille. Muut osakkaat käyttivät myös laajasti tietokonekeskuksen palveluja. Systeemityöpalvelujen käyttö jäi niiden kohdalla kuitenkin satunnaiseksi.

Merkittävä muutos tapahtui vuonna 1977, kun Vakuutusyhtiö Pohjola kiinnostui Tietotehtaan palveluista. Pohjolan tietojärjestelmien perusteellinen uusiminen oli juuri tuolloin käynnistynyt, ja sovelluskehitys vaati suuren määrän lisää henkilökuntaa. Yhtiössä arvioitiin lisähenkilöstön tarve kuitenkin tilapäiseksi, jonka vuoksi neuvottelut käynnistettiin Tietotehtaan kanssa. Mallina neuvotteluille oli Tietotehtaan ja Yhdyspankin välinen systeemityön runkosopimus. Syntyneen sopimuksen perusteella Tietotehtaalta siirtyi 20-30 atk-ammattilaisia Pohjolan tiloihin tekemään sovelluksia Pohjolan projektipäälliköiden vetämiin projekteihin.

Systeemityön runkosopimuksen lisäksi Pohjola halusi päästä Tietotehtaan osakkaaksi. Osakkuudesta sovittiin Tietotehtaan hallituksessa vuonna 1978. Vähän tämän jälkeen myös Vakuutusyhtiö Fennia teki vastaavan sopimuksen Tietotehtaan kanssa. Myös Fennialle Tietotehtaan osakkuus oli tärkeä.

Pohjolan ja Fennian sopimukset olivat alku Tietotehtaan toiminnalle vakuutusyhtiöiden järjestelmäkehityshankkeissa. Toiminta laajeni myöhemmin yhteistyöhön myös Kansan, Sammon, Teollisuusvakuutuksen, Tapiola-Yhtiöiden ja paikallisten vakuutusyhdistysten kanssa.

Pohjolan ja Fennian sopimukset olivat samalla alku Tietotehtaan politiikalle kytkeä suuryritysassiakuuteen yhtiön osakkuus. STS-Pankki tuli omasta aloitteestaan systeemityöasiakkaaksi ja osakkaaksi vuonna 1983. Paljolti samoilla perusteilla OKO tuli vuonna 1986 systeemityöasiakkaaksi ja osakkaaksi. Molempien tavoitteena oli päästä mukaan myös Yhdyspankin vetämään pankkijärjestelmien kehitystyöhön.

Orgaaninen kasvu sisäisellä koulutuksella

Tietotehdas kasvoi nopeasti 1970-luvun alussa. Uusia henkilöitä palkattiin kuin tehtaalte liukuhihnan ääreen. Laatu kärsi erityisesti systeemityössä, mistä johtuen monet asiakassuhteet vaarannettiin noina ensi vuosina.

Kun systeemityön voimakas buumi alkoi uudelleen 1980-luvun alussa, opittiin vähitellen kokemuksen kautta näkemään luonnollisen kasvun rajat. Aluksi rekrytoitiin kokenutta henkilökuntaa kilpailijoilta, käyttäjäorganisaatioista ja erityisesti julkisesta hallinnosta.

Kilpailu johti kuitenkin hyvin nopeasti siihen, että kokeneita atk-henkilöitä saatiin korkeintaan yhtä paljon kuin vaihtuvuus harvensi omia rivejä. Tietotehdas oli pakotettu palaamaan nuorten vastavalmistuneiden rekrytointiin ja sisäiseen koulutukseen.

Vastavalmistuneiden rekrytointi tapahtui 1980-luvun vaihteessa yhteishaussa kaksi kertaa vuodessa joulun alla ja toukokuun lopussa. Yksiköt valitsivat itse omat harjoittelijansa noin 150-200 hakijan joukosta. Valitut sijoitettiin heti alkuun yhteiseen harjoit-

telijakoulutukseen, jonka kurssimainen jakso kesti noin kolme kuukautta. Tämän jälkeen harjoittelijat siirtyivät asiakasprojekteihin kukin omaan yksikkönsä. Opettajina harjoittelijakoulutuksessa toimivat kokeneet henkilöt oman työnsä ohella. Vain kurssien vetäjä oli tehtävässään kokopäivätoiminen.

Jos systeemityöosasto halusi kasvattaa henkilömääräänsä nopeasti, se saattoi kuhunkin jaokseen palkata yleensä korkeintaan yhden harjoittelijan kummastakin harjoittelijahastusta. Tällä vauhdilla ja normaalilla 100:n vaihtuvuudella osasto kasvatti henkilöstöään noin 10% vuodessa. Kokeneiden henkilöiden työaikaa ei voinut ajatella käytettäväksi suuremman lisähenkilöstön sisäänajo. Tilanne ei paljoa muuttunut, vaikka uusia henkilöitä saatiin rekrytoitua myös muista yrityksistä.

Kasvu yritysostojen kautta

Yritysostot merkitsivät Tietotehtaalte toiminnan laajentamista uusille markkina- tai tuotealueille. Finnsystemsint ostolla käynnistettiin kansainvälisten ohjelmistotuotteiden markkinointi. Avonius oli Tietotehtaan kannalta kiinnostava, koska se oli markkinajohtaja IBM-pientietokoneissa. Kansainvälistymisen strategiassa yrityskauppa nähtiin lähes pakonomaiseksi toimenpiteeksi, sillä aikaisemmat yritykset olivat jääneet ponnottomiksi. Yrityskaupat eivät kuitenkaan olleet helppoja. Lähes kaikkiin Tietotehtaan yritysostoihin 1980-luvulla liittyikin vaikeuksia.

Tietotehdas sai Suomessa myös toimialajärjestelijän tehtävän, sillä jatkuvasti sille tarjottiin ostettavaksi huonosti meneviä atk-yrityksiä.

Tietotekniikka oli 1980-luvulla toimialana vasta kypsymässä, mistä

johtuen syntyi jatkuvasti uusia yrityksiä ja vanhoja kuoli kannattamattomana pois. Tässä tilanteessa toimialajärjestelijän rooli oli epäkiitollinen. Yrityksissä ei myöskään ollut myyntikelpoista substanssia, ja tuottoarvo oli enemmän ihmisissä kuin organisaatioissa, menelmissä tai kiinteissä asiakassuhteissa.

Systeemityön organisointi

Tietotehtaalla oltiin 1970-luvulla innostuneita kokeilemaan uusia ajatuksia ja suuntauksia erityisesti henkilöstöhallinnon alueella. Erilaisten organisaatiomuotojen kokeilu oli myös jatkuvaa.

Näistä eriskummallisista oli systeemityön osalta 1970-luvun alussa käyttöön otettu matriisiorganisaatio. Siinä henkilöt sijoitettiin osastoihin ammattitaidon mukaan ja asiakastyöt tehtiin projekteissa tuotanto-osastolla. Jokaisella henkilöllä oli hallinnollinen esimies, joka päätti esimerkiksi palkan korotuksista ja toinen esimies projektioorganisaatioissa, joka vastasi työnjohdosta ja henkilöiden sijoittelusta projekteihin.

Kokemukset toimintokohtaisesta organisaatiosta olivat myös huonot. Erillinen myyntiyksikkö ei osannut myydä itsenäisesti systeemityöprojekteja. Liian pitkälle eriytetty työnjako johti osaamispuutteisiin, tarjousvalmistelun kustannusten nousemiseen ja heikkoon asiakaspalveluun. Tarjousvalmisteluun piti aina kytkeä sekä myyjä että projektipäällikkö. Tällöinkin työstä vastuuseen joutuva projektipäällikkö oli usein helisemässä, kun oma myyjä oli asiakkaan vaatimuksista liian valmis myönnetyksiin.

Eräs onnistuneimpia päätöksiä oli siirtyminen vuonna 1980 tulosityksikköorganisaatioon. Sitä oli edeltänyt joukko erillisratkaisuja kuten pientietoko-

nehenkilöstön kokoaminen yhteen osastoon, laskentapalveluohjelmistojen myynnin ja kehittämisen keskittäminen yhteen yksikköön. Konsultointi ja koulutus siirrettiin Finnsystemsint, joka oli tullut konserniin yrityskaupan kautta.

Tulosityksiköt muodostettiin asiakastoimialapohjalla. Tulosityksikön saivat ensin pankkitoiminta, teollisuus, palvelutoimialat (kauppa, vakuutus ja muut palvelutoimialat) sekä yleisjärjestelmiä käyttävät pienet ja keskisuuret asiakkaat. Varsin pian myös tietokonekeskuksesta tehtiin tulosityksikkö.

Vuonna 1983 aloitettu mikrojen myynti eriytettiin vaiheittain omaksi tulosityksiköksi ja lopulta erilliseksi tytäryhtiöksi. Toimialajakoa yritettiin pitää puhtaana vaikka tulosityksiköitä perustettiin myös tuote- ja palvelupohjalla esimerkkeinä tietokonekeskus, mikromyynti ja koulutus. Yritysostojen kautta konserniin tulneiden Avoniuksen ja Tietojyvän liiketoimintoja järjestettiin myös uudelleen toimialapohjalla.

Tulosityksikköorganisaatio vapautti Tietotehtaan esimieskunnan kilpailemaan vapaasti markkinoilla. Tuloksena oli yksiköiden liikevaihdon voimakas kasvu. Tähän 1980-luvun alussa vallinnut systeemityön ja sovel-lusohjelmistojen voimakas kysyntä antoi hyvät edellytykset. Esimerkiksi vuonna 1978 aloitettu systeemityön myynti vakuutusyhtiöille kasvoi nollasta yli 100 hengen tulosityksiköksi kuudessa vuodessa. Kaupan ja palvelutoimialojen yksikkö kasvoi 20 hengen osastosta samoin yli sadan hengen tulosityksiköksi 1980-luvun ensi puoliskolla. Mikromyynti kasvoi nollasta 100 miljoonan markan liikevaihtoon neljässä vuodessa.

Vaikka Tietotehdas kärsi 1980-luvun alussa henkilöstöpulasta siinä missä sen asiakkaatkin, yhtiö oli pu-

laan paremmin varautunut. Se jopa hyötyi ammattitaitoisen henkilöstön heikosta saatavuudesta. Oma peruskoulutus ja monia asiakasyrityksiä joustavampi palkkapolitiikka antoivat selvän kilpailuedun vallitsevassa tilanteessa.

Asiakkaiksi saatiin myös yrityksiä, jotka siihen asti eivät olleet käyttäneet ulkopuolisia systeemyöpalveluja kehittämishankkeissaan. 1980-luvun alussa aikaisemmin vallinneet ennakkoluulot murtuivat lopullisesti ulkoa ostettavilta atk-palveluilta, ja se johti myöhemmin muun muassa suuryritysten atk-osastojen yhtiöittämisiin.

Toisaalta Tietotehdas oli monille asiakkailleen myös oivallinen rekrytoinnin kohde. Jatkuva asiakassuhde loi pysyvää ja kiinteää henkilösuhteen, joka monissa tapauksissa johti siirtymisiin asiakkaan puolelle. Yhdyspankkiin siirtyi vuosien varrella kymmeniä tietotehtaalaisia esimies- ja asiantuntijatehtäviin. Vakuutusyhtiöiden kohdalla sama siirtymä oli myös hyvin yleinen. Rekrytointi kohdistui yleensä kokeneempaan henkilöstöön. Sitä yritettiin estää myös sopimussakotilla, jotka puettiin koulutuskulukorvauksen muotoon, mutta menettelyä ei pystytty noudattamaan kuin muutamiin harvoihin tapauksiin.

Organisaation optimikoko

Tietotehtaalla tulosityksikön kooksi päätettiin eräällä strategisen suunnittelun kierroksella suositella noin 100 hengen suuruista yksikköä. Tällöin ajateltiin erityisesti systeemyötä räätälintyönä tekevää yksikköä. Tietokonekeskuksen uskottiin vaativan suurempaa kokoa, ja toisaalta fokuoituneet tuote- tai asiakastarvepohjalle perustetut tulosityksiköt olivat luonnostaan paljon pienempiä. Ruotsalaisen Enato-

rin vastaavana tavoitteena oli korkeintaan 50 hengen suuruiset yksiköt.

Tulosityksikkökoon rajoittaminen 50 tai 100 henkeen perustui ajatukselle, että yksikön piti olla niin pieni kuin mahdollista, mutta kuitenkin riittävän suuri toimimaan omillaan ja itsenäisenä ja ilman konsernin sisäistä laskutusta. Tulosityksiköt jakautuivat Tietotehtaalla osastoihin, joiden koko oli 20-30 henkeä. Osasto oli hallinnollinen kokonaisuus.

Osastot jaettiin edelleen 5-8 hengen jaoksiin, joissa asiakastyöt tehtiin. Vain poikkeustapauksissa jaospäällikkö miellettiin hallinnolliseksi esimieheksi. Osastopäällikkö päätti palkan- korotuksista, ammattinimikemuutoksista ja henkilöstön kannalta tärkeistä asioista. Osastopäällikön työajan veivät hallinnolliset tehtävät ja myyntityö. Valmiit sopimukset siirrettiin jaospäälliköiden vastuulle. Jaospäällikkö vastasi asiakkaalle päin projektien miehituksesta ja työn etenemisestä ja valmistumisesta sopimusten mukaan. Tosin eräät jaospäälliköt käsittivät tehtävänsä väärin ja keskittyivät liiaksi hallinnollisiin asioihin. Tätä projektien katteet eivät kestäneet. Jaospäälliköiden laskutustavoite oli noin 90 veloitettavaa tuntia kuukaudessa, kun muilla henkilöillä tavoite oli 120 tuntia.

Toimialapohjaisissa systeemyöyksiköissä ei juuri ollut erillistä myyntihenkilöstöä. Osastopäälliköt vastasivat pääasiassa myyntityöstä. Osastoilla oli oma asiakasreviiri, esimerkiksi tietty asiakas tai asiakastoimiala.

Uuden liiketoiminnan luonti

Tietokonekeskus oli Tietotehtaan rahasampo 1980-luvun lopulle asti. Ohjelmistokehitysprojektit olivat 1970-luvulla vain harvoin erikseen kannat-

tavia. Tuotekehityksen tavoitteena oli 1970-luvulla saada uusia asiakkaita ja lisäkuormitusta tietokonekeskukseen. Systeemyötä tehtiin aluksi vain asiakkaille, joiden käyttötoiminta hoidettiin Tietotehtaalla.

Tuotekehityksessä paneuduttiin aluksi rakentamaan sovellusohjelmistoja pienten ja keskusuurten yritysten laskentapalveluja varten. Esikuvia vastaavalle liiketoiminnalle oli ulkomailla, mutta hyvä kilpailija löytyi myös esimerkiksi kotimaisesta Unicista, jonka koko toiminta perustui laskentapalvelujen myyntiin.

Tietotehtaalle tietokonekeskuksen käyttötoiminta suuryrityksille loi kannattavuuden perustan. Pienyritysten laskentapalveluista ei koskaan saatu syntymään kannattavaa liiketoimintaa.

Räätälöidystä systeemyöstä kehittyi 1980-luvulle tultaessa Tietotehtaan toinen tukijalka tietokonekeskuksen rinnalle.

Liiketoiminnan painopisteen siirtymiset olivat Tietotehtaalla suhteellisen hitaita. Mistään ei jouduttu luopumaan niin nopeasti, että tilalle ei olisi saatu korvaavaa liiketoimintaa. Uudet mahdollisuudet tutkittiin perusteellisesti ja niitä ei hylätty liialla dogmaattisuudella. Tästä hyvä esimerkki oli lähteminen jo vuonna 1975 mukaan pientietokonetoimintaan, vaikka se nähtiin selvästi laskentakeskuspalvelujen kilpailijaksi.

Rekrytointi

Tietotekniikan ammattilaisista oli 1970-luvulla niin määrällinen kuin laadullinenkin pula. Monet tehtävät eivät olleet vielä selkiytyneitä ja koulutus oli puutteellista. Tietotehtaalla ongelmaan

reagoitiin organisoimalla laaja sisäinen kurssitoiminta.

Pidettiin itsestään selvänä, että esimerkiksi korkeakouluista ei saada valmiita henkilöitä ilman intensiivistä jatkokoulutusta. Jo varhaisessa vaiheessa kurssipaikkoja tarjottiin myös asiakkaille maksua vastaan.

Joka vuosi harjoittelijakoulutukseen palkattiin kymmeniä uusia vastavalmistuneita tai opintojensa loppuvaiheessa olevia henkilöitä. Riitti kun oli saanut esimerkiksi ohjelmoinnin tai suunnittelun perustiedot. Monelle tulevalle atk-ammattilaiselle Tietotehdas oli korkeakoulun välitön jatke. Koulutus oli teknistä, käytännönläheistä ja kurssimaisen tiiviisti ohjattua.

Tietoyrityksellä piti olla myös unelmia, joihin henkilöstö halusi sitoutua. Esimiehen rooli oli tärkeä. Uskotava ei ollut esimies, joka ei pystynyt olemaan myös ammattilaisena esikuvaa alaisilleen. Ulkopuolelta tullut "ammattijohtaja" ei menestynyt tietoyrityksessä paitsi riittävän ylhäällä täysin hallinnollisissa tehtävissä tai saneeraajana. Vahva esimies pienessä tietoyrityksessä veti henkilöstön mukanaan. Hän oli paras myyjä, paras asiantuntija ja samalla esikuva henkilöstölle.

Rekrytointi oli eräs tärkeimpiä tietoyrityksen menestymiseen vaikuttavista tekijöistä. Onnistunutta rekrytointia seurasi määrätietoinen koulutus ja urakehityksen ohjaus. Taitava esimies antoi alaisilleen haasteita, jotka olivat kyllin vaativia mutta eivät ylivoimaisia. Jatkuvat onnistumiset työtehtävissä ruokkivat oppimista ja positiivista itseään vahvistavaa kierrettä.

Asiantuntija haluaa koetella voimiaan ja näyttää muille. Tietoyrityksissä kerrotaan mielellään maineteois-

ta ja "sankaripelastajista" sekä hiki-sistä loppuristituksista, joissa projektiryhmän resurssit joutuivat äärimmäisen kovalle koetukselle. Menestyneet urheilijat menestyvät myös tietoyrityksessä. Tästä on useita esimerkkejä. Suoritustehokkuus ja voimakas aikaansaamisen tahto, kilpailuvietti ja halu panna voimansa koetukselle ovat tyypillisiä piirteitä urheilijoille ja näitä ominaisuuksia tarvitaan myös tietoyrityksessä.

Rekrytointipolitiikka löi Tietotehtaalla selvästi leimansa henkilöstön urakehitykseen. Atk-ammattitaidosta tuli arvostuksen mitta. Uraputkeen oli erittäin vaikea tulla sivusta. Esimerkiksi tuotekokemuksen perusteella muualta palkatulla myyntihenkilöstöllä oli sopeutumisvaikeuksia. Asiakasyrityksessä hankittu toimialakokemus tai laskentatoimen tuntemus ei korvanut sitä ammattitaitoa ja arvostusta, joka saavutettiin käymällä läpi Tietotehtaan oma harjoittelijakoulutus ja sitä seurannut sisäänajovaihe asiakasprojekteissa.

Tietotehtaalla henkilöstön palkkaaminen asiakkailta ei muodostunut perustamisen jälkeen enää merkittäväksi. Tästä kärsi varmasti projektityötä tekevien yksiköiden ohjelmistokehitys. Toisaalta asiakasprojekteissa oppi vähitellen näkemään asiakkaan liiketoiminnasta ne piirteet, jotka vaikuttivat tietojärjestelmiin. Myös useamman eri asiakkaan kanssa työskentely avarsi näkemystä usein enemmän kuin, jos koko ikänsä oli vain yhden työnantajan palveluksessa.

Urakehitys

Tietotehdas Oy oli 1970-luvulla tunnettu moniportaisesta ja -mutkaisesta ammattinimikkeistöstään. Aluksi oli kyse ennen kaikkea hinnoittelusta. Yhtiöitetyn atk-osaston työn hinnoittelu oli uusi ja vaikea asia 1960-luvun

lopussa. Insinööritaidolla ja neuvotteiluilla osakasasiakkaiden miehittämisessä työryhmissä päädyttiin lopulta "oikeudenmukaisiin" veloitusperusteisiin. Tuloksena oli monimutkaiset veloitusperusteet, jotka lähtivät tiukasti tuotantokustannuksista aiheuttamisperiaatteen mukaan. Erityisesti konekäsittelyn veloitusperusteet olivat hyvin yksityiskohtaiset ja hinnastot monikymmensivuiset.

Henkilötyön veloitusperusteet olivat aluksi myös hyvin moniportaiset. Konsultoinnille, systeemisuunnittelulle, tekniselle suunnittelulle ja ohjelmoinnille luotiin tiheäportainen hinnasto. Systeemityölle löytyi hinnastossa yli kymmenen erilaista yksikköhintaa.

Tiheä ammattinimikkeistö muodostui vähitellen myös urakehityksen mittariksi. Sisäisessä puhelinluettelossa jokaisen henkilön nimen perässä oli hänen tuntihintaansa kuvaava ammatinimike.

Korkeakoulusta harjoittelijakoulutukseen tuleva henkilö oli ensimmäiset kuusi kuukautta nimikkeeltään suunnittelija-harjoittelija (S/H). Tämän jälkeen kiinteällä palkankorotuksella palkittuna hänet ylennettiin nuoremaksi suunnittelijaksi (S/N). Tässä ammattinimikkeessä jatkui pääasiasa ohjelmointipainotteinen työ seuraavat puolitoista vuotta, jonka jälkeen hänet ylennettiin suunnittelijaksi. Suunnittelijavaiheen jälkeen edessä oli kaksi uravaihtoehtoa.

Tekniikasta innostuneet valitsivat linjan, joka johti ensin erikoissuunnittelijaksi (S/E), erikoisasiantuntijaksi (E) ja lopulta vanhemmaksi erikoisasiantuntijaksi (E/V). Toinen esimiestehtäviin johtava uraputki oli projektinvetäjälinja. Suunnittelijasta edettiin vanhemmaksi suunnittelijaksi (S/V), edelleen nuoremaksi projektipäälliköksi

(P/N), projektipäälliköksi (P) ja lopulta vanhemmaksi projektipäälliköksi (P/V). Systeemityöjaosten vetäjät olivat yleensä vanhempia projektipäälliköitä.

Jaospäälliköstä edettiin osastopäälliköksi ja mahdollisesti yksikön johtajaksi. Useimmat systeemityöosastojen päälliköistä ja jotkut tulosityksikön johtajista olivat 1980-luvun puoliväliin mennessä käyneet läpi tämän uraputken suunnittelijaharjoittelijasta lähtien.

Harjoittelijasta eteneminen projektipäälliköksi kesti yleensä 4-6 vuotta. Kaikille se ei kuitenkaan onnistunut. Toisaalta monia kiinnostivat myös tekniset asiantuntijatehtävät.

Asiakashinnoittelussa luovuttiin tiheästä ammattinimikkeistöstä 1980-luvun alussa. Jäljelle jäi tuntihinta ohjelmoinnille, tekniselle suunnittelulle, systeemisuunnittelulle, projektin vedolle ja konsultoinnille. Samalla yritettiin siirtä henkilöriippumattomaan työtehtävien mukaiseen laskutukseen. Tässä ei kuitenkaan onnistuttu. Niin sitkeässä oli tekijän ammattinimikkeeseen sidottu tuntihinta.

Hinnoittelun muutoksesta huolimatta aikaisempi tiheä ammattinimikkeistö jätettiin - osittain myös henkilöstön toivomuksesta - kuvaamaan urakehitystä ja henkilökohtaisten palkankorotusten pohjaksi.

Monet ulkopuoliset pitivät Tietotehtaan ammattinimikepolitiikkaa omituisena, ja kieltämättä siihen sisältyi ylilyönnin piirteitä. Se oli kuitenkin nuorelle atk-ammattilaiselle selkeä uraputki. Koulussa sisäistetty suora-viivainen kehityspolku - luokka ja vuosi - jatkui vielä työelämän ensimmäiset sisäänaajovuodet. Nimikemuutokset olivat myös julkisia tunnustuksia ja merkki uralla etenemisestä. Projektipäällikkö tai erikoissuunnittelija Tieto-

tehtaalla oli alalla arvostettu henkilö. Takana oli paljon teknisiä taitoja kehittävä koulutusta, harjoittelua ja työtä sekä sisäänajo systeemityön vaikeasti opittavaan maailmaan.

Menetelmät

Liiketoiminnan perinteiden ja kokemuksen puuttuminen pyrittiin Tietotehtaalla korvaamaan 1970-luvun alussa koulutuksen lisäksi voimakkaalla menetelmäkehityksellä. Patenttiratkaisu kaikkii ongelmiin oli perustaa projekti. Tällä tavalla syntyi vähitellen kansioittain ohjeistoja, joita kutsuttiin standardeiksi. Menetelmäkehityksen avulla yritettiin ratkaista myös markkinoinnin ongelmia.

Tilanne olisi ihanteellinen, jos henkilöstön kokemattomuus ja ammatitaidon puutteet voitaisiin korvata hyvillä menetelmillä ja oppailla. Tietotekniikan alueella erilaisilla ohjekirjoilla ja manuaaleilla on aina ollut suuri merkitys. Menetelmäkehitys on ollut hyvin laajasti myös yliopistotutkimuksen ja kaupallisen tuotekehityksen kohteena.

Menetelmäkehityksen tuottamat ohjeistot olivat Tietotehtaalla laajan sisäisen koulutustoiminnan pohja. Ohjeistoja luotiin myös muun muassa asiakasyhteyksiin. Tietotehdas vaati, että asiakkaat toimivat siten, kuin ohjeissa edellytettiin. Vastaavaa toimintamallia näkee yleisesti monissa julkisen hallinnon palveluissa.

Projektitöissä käytettiin standardoitua vaihejakomallia. Jokaiselle projektille laadittiin yhtenäisten ohjeiden mukainen projektsuunnitelma. Tehtävät käskytettiin suunnittelijoille ja ohjelmoijille valmiiksi laadituilla lomakkeilla. Periaatteena oli pilkkoa suuremmat kokonaisuudet mahdollisimman pieniksi osatehtäviksi. Pilkkomi-

nessa pyrittiin yleistettyihin standardiratkaisuihin, jolloin kokemuseräistä tietoa voitiin kerätä erilaisten tehtävien vaatimista työmääristä. Tietoa käytettiin sitten projektien työmääräarvioinnin kehittämiseen.

Hyviä kokemuksia saatiin erityisesti strukturoitujen menetelmien käytöstä ohjelmoinnissa. Samantyyppistä mallia etsittiin myös systeemisuunnittelun tueksi. Yhteiseksi suunnittelumenetelmäksi valittiin aikanaan ruotsalainen ISAC. Siinä tavoitteena oli tietyllä kaavamaisella tekniikalla kuvata määriteltävä tietojenkäsittelytehtävä sellaiseen muotoon, joka olisi havainnollisesti kuvannut tehtävän ja josta olisi ollut helppo jatkaa ohjelmointityö.

Käytännössä ISAC-menetelmä tuotti suunnattomat määrät lomakkeita. Asiakkaat eivät niitä ymmärtäneet, ja suunnittelijakin ymmärsi ehkä tekniikan mutta ei välttämättä määriteltävää ongelmaa. Hyvin sekavaan paperinippuun pyydettiin sitten hämmennyneen asiakkaan puumerkki hyväksymisen merkiksi. Kun ongelmat lopulta testausvaiheessa selvisivät, vedottiin kuvauksiin ja asiakkaan puumerkkiin, ja vaadittiin lisätöille ja muutoksille uutta kustannusarviota ja jatkoa toimitusajalle.

Usko menetelmien tehoon alkoi horjua 1970-luvun lopulla. Kaavamaisiin malleihin innostuttiin kuitenkin uudelleen 1980-luvulla, kun tietokantapohjaisten järjestelmien kehittäminen ja case-välineiden käyttö tuli muotiin.

*Tapani Seppänen,
HM&V Research Oy,
Itälahdenkatu 22 A,
00210 Helsinki,
puh: 0400-444 701,
e-mail: tapani.seppanen@hmv.fi*

Muisteluksia systeemin suunnittelun alkuvuosilta

Miikka Jahnukainen

Aluksi se oli systeemin suunnittelua

Käsite ja termi systeemyö on myöhäsyntyinen, sillä se tuli käsityksen mukaan käyttöön vasta 60-luvun jälkimmäisellä puoliskolla. Vuoden 1969 Tietojenkäsittelyn sanastossa (Pietarinen 1969) sana ei vielä esiinny. Toisaalta ATK-systeemin suunnitteluprosessia tutkiva työryhmä, jonka raportti painettiin vuonna 1968 (Jahnukainen ym. 1968) käytti sanaa systeemyö jo raporttinsa otsikossa. Eero Kostamo on käyttänyt termiä systems work vuonna 1965 (Kostamo 1965/2).

Aluksi tietokonetta vain ohjelmoitiin ja koulutuskin alkoi ohjelmointikurssituksella. Kaupallishallinnollisen soveltamisen alueella ohjelmien tekemisen ja halutun kokonaistuloksen väliin jäi kuitenkin laaja työkenttä, joka oli luonteeltaan loogista ja käsitteellistä organisaation toimintatapojen, informaationkulun ja tietojenkäsittelyn suunnittelua. Siitä alettiin 1950/60-lukujen vaihteessa käyttää termiä 'systeemin suunnittelu'. Näen siis, että systeemyön historia oli aluksi systeemin suunnittelun historiaa.

Systeemin suunnittelijasta tulikin mielestäni sillanrakentaja organisaation kehitettävien toimintojen ja sen ihmisten sekä tietokoneen hyväksikäytön välille. Toimivan kokonaisuuden kannalta katsottuna systeemin suunnittelijat ja ohjelmoijat (ohjelmoitsijat)

toimivat eri osapuolten välisen yhteistoiminnan järjestelyn, järjeistämisen ja automaation avulla tehostamisen asiantuntijoina ja järjestelmien rakentajina ja toteuttajina. Siksi onkin luonnollista, että useiden atk-organisaatioiden kehto on ollut rationalisointiosasto.

Systeemin suunnittelija uuden työnsä äärellä

Ensikosketukseni systeemin suunnitteluun sain helmikuusta 1960 alkaen Helsingin kaupungin tietojenkäsittelykeskuksessa, osin jo ennen sen perustamista. Ensimmäinen systeemin suunnittelutehtäväni oli varastokirjanpitojärjestelmän suunnittelu rakennusvirastolle.

Huomaan muistavani työstä varsin vähän. Mieleissäni on muistikuva, jossa istun lukuisia kertoja varastokirjanpitäjän kanssa asiaa läpi käyden ja hänen työtään seuraten hänen pöytänsä vieressä olevalla tuolilla. Paneuduin huolellisesti ja innokkaasti hänen näkökulmiinsa. Sen muistan, että hän suhtautui varsin positiivisesti tulevaan systeemiin. Tästä varmaan lähti käsitykseni muutosvastarinnasta; että se enemmänkin synnytetään kuin on sinänsä olemassa. Kyseinen henkilö oli todellakin avainhenkilö asenteiden suhteen, esimieheni Otto Karttunen ja Eero Kostamo olivat tietenkin jo aiemmin "myyneet" systeemin rakennusviraston johdolle.

Alkuaikoina systeemin suunnittelun metodiikka muodostui lähes 'tyh-

jästä', kokeile ja erehdy –periaatteella käytännön tekemisen myötä. Elinkin systeemiä suunnitellessani kädestä suuhun. Se tarkoitti sitä, että siihen tuli muutoksia monia kertoja. Minulla oli työparinani Jouni Pulkkinen ohjelmoijana. Hän aloitti työnsä varsin aikaisessa vaiheessa. Jouni oli varmaan monta kertaa menettämässä hermonsä kanssani. Sellainen asia kuin systeemin suunnitelman jäädyttäminen ennen ohjelmoinnin aloittamista oli täysin tuntematon lähestymistapa.

Yksi tehtävä oli varastonimikkeiden koodittaminen. Mitään mallia ei sitä varten ollut. Tämä tarkoitti, että olit arvostettu koodiasiantuntija tehtyäsi ensimmäisen koodituksen. Muistaakseni tulos oli ns. luokitteleva koodi, joka siis identifioinnin lisäksi sisälsi raportteja varten tarvittavia ja muistia helpottavia alakoodeja. Koodin piti tietenkin muistitilan säästämiseksi olla mahdollisimman lyhyt. Jokin vuosi myöhemmin päädyimme siihen, että järkevin koodityyppi on sittenkin ns. juoksevan numeron koodi joustavuutensa takia.

Systeemin suunnittelun systematisointi

Ensimmäiset kokemukset systeemin suunnittelusta herättivät tarpeen systematisoida ja alkaa kehittää suunnittelutyön periaatteita ja menetelmiä. Eero Kostamo alkoikin hahmottaa systeemin suunnittelulle ns. vaihejakoa jo 60-luvun alkuvuosina. Olin tässä menetelmäkehittämissä

mukana. V.1963 ilmestyneestä Kostamon toimittamasta kurssikirjasta (Kostamo 1963) saa käsityksen tuolloisesta metodiikan tilanteesta. Muis- taakseni ainakin Valtion tietokonekes- kuksen alkuaikoina oli jo käytössä oma vaihejakonsa.

Toinen lähestymistapa on SOP (IBM 1962). Tuo vihkonen on selväs- ti kulunut käytössäni, ilmeisesti kuiten- kin vasta muutama vuosi myöhemmin. Siinä käytetään sanaa Business Sys- tem. Lähestymistapaa kuvaa ”true goals are identified in terms of what the business contributes to its environ- ment.”. Puhutaan aktiviteeteista, toi- minnoista. Mielestäni siinä näkyy jo piirteitä prosessilähestymistavasta. Tuolloin SOP ei kuitenkaan minulla edennyt käytännön ymmärtämisen ja soveltamisen tasolle.

Tietokoneyhdistyksen puitteissa aloitti 1966/67 työryhmä tehtävänään koota ehdotus ATK-systeemin suun- nitteluprosessin rakenteeksi ja sisällök- si. Raportin (Jahnukainen ym. 1968) esipuheessa todetaan, että rakenne- malleina on käytetty Valtion tietoko- nekeskuksen vaihejakoa, Kesko mal- lia sekä Nokia Elektroniikan dokumen- tointijärjestelmää. Vaihejaon ja sen periaatteiden osalta viitataan kahteen teokseen (Kostamo 1965 ja Jahnukai- nen 1965). Sain tehtäväkseni vetää työryhmää, johon kuuluivat Erkki Eväsoja, Reino Knaapila, Erkki Nie- mi, Kauko Pursiainen, Arto Sainio, Reijo Salo sekä Lauri Varjus.

Teimme innolla ja kaikella vaka- vuudella työtä. Kokosimme ehdotuk- sen systeemityön vaihejaoksi, jonka siis valitsimme systematisoinnin poh- jaksi. Totesimme kuitenkin, että osuu- det voidaan toteuttaa tehtyjä ratkai- suja asteittain täsmentämällä, iteroi- malla. Tuo sana iterointi oli tuolloin käytössä lähinnä matematiikassa, mutta se tuntui soveltuvan hyvin ku-

vaamaan näkemystämme suunnitte- lun etenemistavasta. Samoin me kä- sittelimme perusteellisesti vaihejaon soveltamista eri systeemityötilanteis- sa. Meni kuitenkin todella pitkään en- nenkuin esimerkiksi ylläpitometodiik- ka sai arvoisensa huomion, muutakin kuin maininnan.

Koska olin monella tavalla osal- lisenä systeemin suunnittelun systema- tisoinnissa, pidin aiheesta myös esityk- set alan pohjoismaisissa konferensseis- sa vuosina 1966 ja -67 (Jahnukainen 1966/2 ja Jahnukainen 1967/2). Eero Kostamo oli jo aiemmin kirjoittanut BIT-lehteen artikkelin (Kostamo 1965/2). Mutta oltiin valmiita otta- maan vastaan suomalaista systeemin- suunnittelukulttuuria?

Systeemin suunnittelun koulutuksen aloittaminen

Edellä olevan pohjalta on helppo ymmärtää, että koettiin voimakas tar- ve aloittaa systeemin suunnittelun jär- jestelmällinen koulutus. Jo lukuvuon- na 1961-62 Asko Kousa IBM:stä ja Raimo Suoniemi Kaapelitehtaan Elektroniikkaosastolta olivat pitäneet Kauppakorkeakoulussa luentosarjan, jossa käsiteltiin myös ETK:n yleisim- piä sovellutusaloja. Termiä systeemin- suunnittelu ei käytetä, mutta siihen liit- tyviä asioita otetaan esille, joskin sup- peasti. Aineisto julkaistiin monistemuoi- toisena kirjana (Kousa, Suoniemi 1962).

Otto Karttunen, joka siihen ai- kaan pohjusti Valtion tietojenkäsitte- lykeskuksen perustamista, otti järjes- tääkseen yrityksille ja muille organi- saatioille suunnatun systeemin suun- nittelukurssin mm. Ruotsista saamiensa esikuvien ja mallien pohjalta (Karttu- nen 1986, ss. 68-72 ja ss. 277-278). Kurssi järjestettiin 4-viikkoisena inter- naattikurssina. Kurssiaineisto oli aluksi kansiomuodossa, myöhemmin kirjana

(Kostamo 1963). Itse olin mukana opettajana jo tällä ensimmäisellä kurs- silla. Osa kurssista noudatti systeemin- suunnittelun vaihejakoa ja siinä oli pal- jon ryhmätöitä koskien käytännön sys- teeminsuunnittelutehtäviä.

Minun osuuteni oli vetää yksi ryhmätöistä, aiheena materiaalilas- kenta. Aikaa oli 1,5 päivää, joten asi- aa saattoi käydä läpi varsin perusteel- lisesti sarjana ryhmätöitä. Kävimme läpi mm. tilanne- ja muutosluettelon suunnittelua, inventoinnin suorittamis- ta, varaston täydentämismenettelyä, systeemin tarvitsemia tietoja ja input- lomakkeita, käsitejärjestelmää lyhen- teineen ja laskentaperusteita sekä materiaalilaskennan työkulua.

Myöhemmin sain Tietojenkäsit- telyneuvonta Oy:ltä, joka oli perustet- tu mm. näiden kurssien pitämistä var- ten, tehtäväkseni vetää näitä kurse- ja, joita toistettiin 3-viikkoisina kaksi kertaa vuodessa koko 60-luvun. Se oli haastava ja mielenkiintoinen tehtävä, jossa Suomen silloinen systeemin suun- nittelijakunta tuli varsin tutuksi. Kurs- sit tulivat aina täyteen, eli n. 25 hen- keä per kurssi.

Systeemin suunnittelun opetus Tampereen yliopistossa

Tampereen Yliopisto aloitti en- simmäisenä Suomessa tietojenkäsitte- lyopin opetuksen omana oppiaineenaan syksystä 1965. Minulla oli näkö- alapaikka asiaan, kun aloitimme Rei- no Kurki-Suonion kanssa opetuksen, Kurki-Suonio professorina ja minä as- sistenttina, myöhemmin apulaisprofes- sorin virkaa hoitaen. Meillä oli selkeä roolijako. Kurki-Suonio opetti ohjel- mointikielten ja ohjelmoinnin teoriaa (Kurki-Suonio 1967 ja 1969), jota säesti ohjelmointi algoritmisella ohjelmointi- kielellä Algol. Minä vastasin kauppal- lishallinnollisen atk:n ja systeemin suun- nittelun opetuksesta.

Ensimmäisen peruskurssin pidin syyslukukaudella 1965. Muistelen että peruskurssilla saattoi olla jopa 300 opiskelijaa, koska se lieene kuulunut tiedekunnan pakollisiin peruskursseihin. Oli siinä kokeissa korjaamista. Kurssi pidettiin yliopiston juhlasalissa, joka on auditoriotyyppinen ja varustettu suurella esiintymislavalla. Vähän hankala paikka 28-vuotiaalle nuorukaiselle, näin jälkeenpäin ajateltuna. Opetussisältöön sain tukea mm. Sven Hedin kirjasta (Hed 1966, suomeksi). Myös bittimuotoinen tietojen esitystapa opetettiin.

Kevätkaudella 1966 sitten seurasi approbatur-opintoihin kuuluva systeemin suunnittelukurssi. Olin saanut esimakua sen läpivientiin liittyvistä ongelmista jo pari vuotta aiemmin, kun Olli Lokin pyynnöstä pidin systeemin suunnittelun luentosarjan Teknillisessä Korkeakoulussa. Haaste oli nimittäin siinä, että opiskelijoilla ei ollut käytännön käsitystä yrityksen toiminnasta eikä varsinkaan systeemien suunnittelusta. Minua oli siis hemmoteltu Tietojenkäsittelyneuvonnan kursseilla, joissa kaikilla oli yritykokemusta ja monilla jo systeemin suunnittelustakin konkreettista taustaa. Kuitenkin vedin kurssia käytännönläheisenä ryhmätyösarjan avulla. Pidin sitä tuolloin luonnollisena lähestymistapana tietojenkäsittelyopin approbaturiin. Eikä sopivaa teoreettista perustaa ollut käytettävissäkään. Varsinaista kurssikirjaa en käyttänyt.

Edellä oleva johti siihen, että totesin välttämättömäksi, että opiskelijat saavat käytännön kokemusta yrityksissä tai muissa sopivissa toimintaympäristöissä ennen opintojen jatkamista seuraavana syksynä. Opiskelijoita oli noin 50. Joten siitä vain töihin kesäharjoittelupaikkoja hankkimaan. Onneksi tunsin Suomen atk-päälliköt varsin kattavasti. Aloitin soittelun, ja niin kaikille löytyi kuin löytyikin har-

joittelupaikka. Taisi paikan valinta suunnata joidenkin henkilöiden elämänkulkuakin. Aikamoinen elämän kokemus minullekin. Parin vuoden sisällä opiskelijat itse ottivat harjoittelupaikkojen hankinnan hoitaakseen.

Opetuksessani oli siis selvästi ammattikoulutuksen leima. Alalle aikoville ei ainoastaan tietojenkäsittelyopin mielekäs sisältö ole tärkeä. Muis- tankin aloittaneeni ensimmäisten vuosien peruskurssit antamalla ohjeita sopivaksi aineyhdistelmäksi alalle aikovalle: pääaineeksi soveltava aine (esim. yrityksen taloustiede) ja cum laude tietojenkäsittelyopissa tai päinvastoin. Harva ”totteli”; useimmat ottivat tietojenkäsittelyopin rinnalle metodiaineita, osittain johtuen tiedekuntajaosta. Käsitykseni mielekkäästä aineyhdistelmästä on edelleen sama, paitsi että myöhemmin olen lisännyt psykologian, koska ”ihmisasiat” ovat osoittautuneet yhä tärkeämmiksi.

Systeemiajattelu

Eero Kostamoa rupesi askarruttamaan systeemin suunnittelun teoria. Hän kirjoitti ajatuksistaan kirjan vuonna 1965 (Kostamo 1965/1). Hän etsii ATK-systeemin yleistä, standardoitua mallia.

Minäkin pohdin samansuuntaisia asioita. Lähtökohdaksi tuli systeemiajattelu. Se sallii samanaikaisesti kokonaisuuden hallinnan ja yksityiskoh- tien tarkan tutkimisen (Jahnukainen 1966/1). Ymmärrystäni lisäsi systeemien jakaminen luonnollisiin systeemeihin (esim. aurinkokunta ja veden kiertokulku luonnossa) ja ihmisten tekemiin systeemeihin. Systeemiteorian peruskäsitteistön sain lähinnä kybernetiikasta (Ashby 1956). Systeemin suunnittelun näin metodisesti menetelminä, joita käytetään suunniteltaessa ihmisen muodostamia systeemejä. Verkkoiteorian eli graafien teorian otin

myös avukseni (Berge 1962). Graafi muodostaa varsin käyttökelpoisen systeemin kuvausjärjestelmän, systeemi kuvataan verkkona. Maailmalta tuli tuossa vaiheessa vastaan vain yksi kirjoitus, jossa systeemiajattelua oli sovitettu atk:hon (Sveistrup 1964).

Juuri vaatimus yht’aikaa hahmottaa ja jäsentää epämääräisiä kokonaisuuksia ja tehdä loogista ja tarkkaa työtä yksityiskohdissa on keskeistä atk-systeemien suunnittelussa. Se asettaa suorastaan skitsofreenisen vaatimuksen systeemin suunnittelijan kyvyille. Systeemin suunnittelijalle hahmottelin apuvälineeksi systeemimallia kuvausvälineeksi (Jahnukainen 1966/3 ja 1967/1).

Systeemimallilla tarkoitin sellaista todellisen systeemin mallisysteemiä, joka on muodostettu käyttäen hyväksi systeemiajattelua ja kokonaisvaltaista lähestymistapaa. Olavi Niitamo, joka ohjasi lisensiaattityövaiheittani Helsingin yliopistossa sekä rohkaisi ja avusti edellä mainittujen kahden artikkelin julkaisemista, innostui vertaamaan lähestymistapaani Oscar Langen näkemyksiin (Lange 1965). Joten pääsin arvokkaaseen seuraan.

Näistä lähtökohdista pidin ensimmäisen systeemiajattelun kurssin syksyllä 1966 systeemin suunnittelun cum laude –opintojen lähtökohdaksi. Aikamoista etsintää ja kokoamistahan se oli. Hämmästykseni onkin ollut melkoinen, kun olen myöhemmin saanut palautetta, että tuo kurssi on antanut asianomaisille narunpäitä myös aivan eri asioihin kuin yritysten atk-systeemien suunnitteluun.

Tähän loppuu omien kokemusteni pohjalta sopivasti systeemin suunnittelun historian alkuvuodet. Se oli mielenkiintoista ja innostavaa aikaa, jossa omalle luovuudelle ja rajoille oli jatkuvia haasteita.

Kiitokset

Lämpimät kiitokseni Jouko Sepäselle, joka veti minut mukaan tähän systeemityön historiaproessiin ja joka on tukenut kirjoitustyötäni monin tavoin.

Jouko on myös korostanut lähteiden keskeisyyttä historia-aineistoa koottaessa, joten olen sen mukaisesti liittänyt mukaan laajan viitteiden luettelon. Viimeistelyä helpotti, että saatoin poimia viitteet valmiiksi muotoiluina hänen kokoamastaan tietojenkäsittelyn historian lähdeluettelosta.

Lisäksi Jouko pyysi laajentamaan systeemiajattelua koskevaa osuutta.

Viitteet:

Ashby R. (1956): *An Introduction to Cybernetics*. London, 1956.

Berge C. (1962): *The Theory of Graphs*. London. Dunod (1958), Paris.

IBM (1962): *Study Organization Plan. The Approach*. Special Edition, April, 1962.

Kousa A., Suoniemi R. (1962): *Elektroninen tietojenkäsittely (ETK) ja sen kaupallisia sovellutuksia*. Liiketaloustieteellinen tutkimuslaitos. Helsinki. 122 s.

Kostamo E., toim. (1963): *Automaattisten tietojenkäsittelysysteemien suunnittelu*. Systeeminsuunnittelukursien opettajakunta, Helsinki.

Sveistrup P. (1964): *Systembegrebet og administrativ databehandling*. Erhvervsekonomisk Tidsskrift, nr. 2.

Kostamo E. (1965): *ATK-systeemien suunnittelun perusteista*. Tietokoneyhdistys ry., Helsinki. 80 s.

Kostamo E. (1965): *The Contents of ADP Systems Work in Administrative ADP Applications*. BIT 5, nr. 1.

Jahnukainen M., toim. (1965): *ATK-systeemien dokumentointi ja standardointi*. Tietojenkäsittelyneuvonta Oy, Helsinki.

Lange O. (1965): *Wholes and Parts. A General Theory of Systems Behaviour*. Pergamon Press. PWN.

Jahnukainen M. (1966): *Mitä on systeemijättele*. Sosiologia N:o 1. 4 s.

Hed S. (1966): *Atk: automaattinen tietojenkäsittely*. Tammi, Helsinki.

Jahnukainen M. (1966): *Ett sätt att systematisera systemarbetet - dokumentation under systemarbetet*. NordSAM 1966, Vædbek, Danmark.

Jahnukainen M. (1966): *Systeemimalli kuvausvälineenä*. Kansantaloudellinen Aikakauskirja, Nide 3. 14 s.

Jahnukainen M. (1967): *Systeemimalli yrityksen informaationsysteemin ja sen suunnittelun kehysmallina*. Lisensiaattitutkimus, Valtiotieteellinen tiedekunta, Helsingin yliopisto, Helsinki. 101 s.

Kurki-Suonio R. (1967): *Ohjelmoinnin loogislingvistiset perusteet*. Tampereen yliopiston Tutkimuslaitos, Tampere.

Jahnukainen M. (1967): *Krav til standardmetodik i systemarbetet*. Diskussionsinlegg holdt under NDD-67, Moderne databehandling, Nr. 3, Oslo. 5 s.

Niitamo O.E. (1967): *Miikka Jahnukaisen systeemianalyttinen tutkimus*. Liiketaloudellinen Aikakauskirja N:o III-1967, Helsinki.

Jahnukainen M., Eväsoja E., Knaapila R., Niemi E., Pursiainen K., Sainio A., Salo R., Taimisto H., Varjus L. (1968): *ATK-systeemityön rakenne ja sisältö*. Tietokoneyhdistys ry., Helsinki. 38 s.

Kurki-Suonio R. (1969): *Johdatus ohjelmointiin*. Tampereen yliopiston Tutkimuslaitos, Tampere.

Pietarinen I., toim. (1969): *Tietojenkäsittelyn sanasto*, toinen painos. Tietokoneyhdistyksen julkaisuja numero 5, Helsinki.

Jahnukainen M. (1970): *Yrityksen informaationsysteemin suunnittelun kehysmetodi*. Kansantaloudellinen yhdistys, Helsinki.

Karttunen O. (1986): *Avainpaikalla tietotekniikan kehityksessä*. Tietotekniikan liitto ry., Helsinki. 276 s.

Miikka Jahnukainen

1960-luvun kirjallisesta toiminnasta systeemis suunnittelun opettamiseen

*Pertti Järvinen,
Tampereen Yliopisto,
Tietojenkäsittelyn laitos*

Tausta

Systeemis suunnittelun historian aloitusmerkkipaaluuna voidaan pitää Otto Karttusen vetämää systeemityökerhoa Helsingissä 1960-luvun alussa. Karttunen toimi myöhemmin VTTK:n johtajana ja siirsi tietämystään valtion hallinnon piiriin.

Kansainvälisinä tieteellisinä merkkipaaluina voidaan pitää insinööri Börje Langeforsin artikkeleita BIT-lehdessä (1963). Langefors toimi dosentina Chalmersin teknillisessä korkeakoulussa Göteborgissa 1960-luvun opettaen systeemis suunnittelua. Langeforsin artikkeli (Taggard ja Tarp, 1977) ja Langeforsin kirja vuodelta 1966 (Langerfors, 1966; Couger, 1973) mainitaan ensimmäisinä tieteellisinä julkaisuna alalta.

Ensimmäisen suomalaisen systeemin suunnittelun kirjan kirjoitti Eero Kostamo 1965 (Kostamo, 1965). Pian sen jälkeen ilmestyivät Tietokoneyhdistyksen työryhmän toimesta Tietosysteemien dokumentointi ja standardointi (ns. sininen kirja) ja ATK-systeemityön rakenne ja sisältö (ns. valkoinen kirja). Miikka Jahnukainen toimi työryhmä kantavana voimana. Kaikki edellä mainitut teokset perustuvat vaihejakomalliin, jonka lähtökohdan Maritta Hoffrén (1988) paikansi Agrestin (1986) kautta Canningin kirjaan (1956).

Oma tieni alalle

Kirjoittajan oma alalle tulo tapahtui 1963, jolloin osallistuin ensin IBM:n ohjelmoijateistiin ja myöhemmin kyselemällä työtä IBM:llä. IBM tarjosi työtä Säteri Osakeyhtiöstä Valkeakoskelta, mutta kesken työpaikkaneuvotteluja luokseni tullut Mauno Rautiainen Ovakon Imatran Terästehtaalta sai minut houkuteltua Imatralle. Osallistuin psykologian professori Kullervo Rainion kanssa IBM:n järjestämälle Fortran-ohjelmointuikurssille touku-kuussa 1963 ja aloitin työt Imatralla kesäkuun alussa 1963, jossa sain vastuulleni teollisuuden sovellukset.

Yhtiö oli tilannut IBM 1401 tietokoneen, joka silloin oli yleisin hallinnollisiin sovelluksiin käytetty tietokone maailmassa. Siihen oli tilattu rivi-kirjoitin, reikäkortinlukija ja 4 vaihtolevy-yksikköä, mikä oli varsin modernia, sillä useimmissa tietokoneissa oli silloin nauhamuistit ulkoisina muisteina. Ennen koneen tuloa testasin ohjelmia käymällä ajamassa niitä Helsingissä työtoverini Reino Härkösen kanssa, jonka vastuulla oli taloushallinnon sovellukset. Hän kertoi, miten reikäkorttiajattelu istui tiukassa. Reikäkorttiosaston päällikkö Jussi Nyholm oli suunnitellut tietokoneelle palkka-ajon, jossa tarvittiin 9 summakorttia. Todellisuudessa levymuisti palveli niin hyvin, ettei summakortteja tarvittu lainkaan.

IBM:n myyntimiehet ja sovellustuki, Jussi Klint, Jori Luolamo, Jorma Vapaa ja Bo Nyholm vierailivat eri

yrityksissä, joihin oli hankittu IBM-koneita, ja kertoivat mm. miten Timo Järvi samaan aikaan rakensi Turun Sanomien systeemejä. Mainitut myyntimiehet olivat myös halukkaita oppimaan, miten juuri toimitettu uusi tietokone installoidaan, eli miten siihen ladataan systeemiohjelmisto. Nostimme koneen pystyyn useamman kerran peräkkäin niin, että jokainen varmasti osasi sen tehdä. Tuossa harjoittelussa meni aika myöhään pikkutunneille asti.

Ovakon, Imatran Voiman ja Enso-Gutzeitin Imatran tehtaiden atk-ammattilaiset perustivat Tietotekniikan Liiton ensimmäisen paikallisyhdistyksen, Imatran Atk-killan vuonna 1965 ja olin sitä mukana perustamassa.

Laadin ohjelmistoja pääasiassa Tuotannotarkkailuosaston tarpeisiin. Tuotannotarkkailuosastolla oli keskeytymättömässä kolmivuorotyössä yli 30 henkilöä kirjaamassa eri puolilla tehdasta tuotannon tapahtumia. Tarkkailukirjanpidon varaan voitiin rakentaa tuotannon määrällinen ja laadullinen seuranta. Ammatillisesti haastavimpia tehtäviä olivat seuraavat:

1. Raporttigeneraattorin tekeminen. Välisummien ja loppusumman lisäksi käyttäjä saattoi antaa lausekkeen, joka tuli laskea. Kun lausekkeessa sallittiin sulku-merkit, jouduin tekemään pienen osan tulkkia ja muuntamaan rekursion iteraatioksi.
2. Ohjelmat levyille. Kun normaalisti ajettavat ohjelmat olivat reikäkortteilla, tein pienen ohjelman,

jolla ohjelmakoodi voitiin tallettaa levyille ja lukea sieltä paloina 12 K:n keskusmuistiin. Järjestely vastasi käyttäjärjestelmää alkeellisessa mittakaavassa, ja samalla se poisti ohjelman maksimikoon rajoituksen.

3. Panostuksen tuki. Terästä tehtiin sulattamalla romurautaa. Ruostumattoman teräksen tekemisessä tarvittiin lisäksi nikkeliä, kromia ja molybdeenia, joita lisättiin sulatuksen loppuvaiheessa. Kun alkupanos oli sula, otettiin siitä näyte. Kemiallinen analyysi soitettiin tietokonehuoneeseen, jossa se naputeltiin konsolikirjoittimella tietokoneen muistiin. Ohjelmani, jossa oli 7 yhtälön yhtälöryhmä, ratkaisi sitten, paljonko ferrokromia, piikromia, nikkeliä, molybdeenia jne tuli lisätä, jotta saataisiin tietynlaatuista ruostumatonta terästä. Teimme noin 25 sulatusta, jotka kaikki osuivat rajoihin.
4. Grafiikkaa rivikirjoittimella. Teräksen valmistuksen tietyssä vaiheessa aihiot olivat tietyn ajan lämpökuopissa. Laadin lämpökuoppien käyttöraportin graafisesti siten, että kuvasta oli nähtävissä kunkin kuopan kuormitus vuorokauden eri tunteina.

Siirryin Ovakolta Tampereen yliopiston Tietokonekeskuksen palvelukseen 1967. Kohta siirtymiseni jälkeen entisen työnantajani Tuotannon tarkkailuosaston osastopäällikkö Erkki Ruotsi pyysi minua katsomaan hänen uutta systeemin suunnitteluvälinettä. Hän oli merkinnyt isolle valkealle seinälle tehtaan pohjapiirroksen. Hän kuvasi tietojen siirtoa yksiköiden välillä villalangoilla, jossa roikkui kyseinen lomake, Post It-tarra tms. Myöhemmin ymmärsin, että Erkki Ruotsi

oli keksinyt seinätaulutekniikan jo paljon ennen kuin Kari Saaren-Seppälä toi sen julkisuuteen.

Systeemis suunnittelun opetus yliopistoissa alkaa

Miikka Jahnukainen oli aloittanut kaupallishallinnollisen tietojenkäsittelyn opetuksen Tampereen yliopistossa syksystä 1965. Opetukseen sisältyi peruskurssin ja systeemis suunnittelukurssin jälkeen systeemiajattelun kurssi, jonka tarkoituksena oli antaa käsitteellistä perustaa atk-systeemien suunnittelulle. Lähestymistapaa kuvaa Jahnukaisen oma artikkeli parhaiten (Jahnukainen, 1966). Tukena oli monia teoksi, mutta Börje Langerforsin kirjaa (1966) ei kuitenkaan sen tekemisessä käytetty hyväksi.

Miikka Jahnukainen laati ensimmäisen systeemin suunnittelun väitöskirjan (1970) ja tuli nimitetyksi apulaisprofessuuriin, jota hän ei vakinaisena ehtinyt hoitaa päivääkään, sillä hän aloitti Atk-instituutin vetämisen vuonna 1970.

Lähteet

ATK-systeemityön rakenne ja sisältö (1968) Tietokoneyhdistys ry

Agresti W.W. (1986) The conventional software life-cycle model: Its evolution and assumptions, In Agresti (Ed.), New paradigms for software development, IEEE Computer Society No 707.

Beer S. (1966) Decision & control, Wiley, London.

Canning R.E. (1956) Electronic data processing for business and industry, Wiley, New York.

Couger J.D. (1973), Evolution of business system analysis techniques, Computing Surveys 5, No 3, 167-198.

Hoffrén M. (1988) Prototyypilähestymistavasta ja sen vertailusta ohjelmiston perinteiseen kehitystyömalliin, Tampereen yliopisto, Tietojenkäsittelyopin laitos.

Jahnukainen M. (1970) Yrityksen informaationsysteemin suunnittelun kehysmetodi, Kansantaloudellinen yhdistys

Jahnukainen, M. (1966) Systeemimalli kuvausvälineenä. Kansantaloudellinen Aikakauskirja. Nide 3.

Kostamo E (1965) ATK-systeemien suunnittelun perusteista, Tietokoneyhdistys ry.

Langefors B. (1963) Some approaches to the theory of information systems, BIT 3, No 4, 229-254.

Langefors B. (1966) Theoretical analysis of information systems, Studentlitteratur, Lund.

Shannon C. and W. Weaver (1949) The mathematical theory of communication, University of Illinois Press, Urbana.

Taggard W.M. and M.O. Tarp (1977) A survey of information requirements analysis techniques, Computing Surveys 9, No 4, 273-290.

Wiener N. (1948) Cybernetics, Wiley, New York.

*Pertti Järvinen,
professori
Tampereen Yliopisto,
Tietojenkäsittelyn laitos*

1970-luku: Pohjoismaisen systeemityön tutkimusyhteistyön aloittaminen

*Pertti Järvinen,
Tampereen Yliopisto,
Tietojenkäsittelyn laitos*

Kun prof. Miikka Jahnukainen siirtyi Atk-instituutin vetäjäksi, professori Reino Kurki-Suonio pyysi minua Tampereen yliopiston apulaisprofessorin viran hoitajaksi. Sain tietooni, että professori Börje Langefors järjestää kesällä 1970 pohjoismaisen kesäseminaarin systeemin suunnittelun teoriasta. Osallistuin seminaariin ja sen jälkeen pidettyyn pohjoismaisen SCIP-projektin kokoukseen. Mainitussa tutkimusprojektissa, joka kansainvälisesti ajatellen jo hyvin varhain tähtäsi tietokonetuettuun systeemin suunnitteluun, olivat mukana Bubenkon ja Lundebergin ryhmät Ruotsista, Sølvsbergin ryhmä Norjasta, Sveistrupin ja Relstedin ryhmät Tanskasta sekä Andersinin ryhmä Suomesta. Projekti otti Difo-ryhmän Tampereelta toiseksi suomalaiseksi ryhmäksi.

Myönteinen päätös oli lähtölaukaus pitkäaikaiselle pohjoismaiselle yhteistyölle systeemin suunnittelun alalla ja erityisesti yhteistyölle Bubenkon, Sølvsbergin ja professori Hannu Kangassalon kesken. Viimemainittu johti vuosikausia Difo-tutkimus ry:tä ja oli kansani yhdessä Vesa Savolaisen, Markku Nokso-Koiviston, Sakari Lehtosen ja Timo Tervakon kanssa järjestämässä vuosittaisia Difo-tutkimuspäiviä Tampereen yliopistossa.

SCIP-projektin alkuvaiheen tuotoksia on esitelty kirjassa Systemering 70 (Bubenko et al. 1970), keskivaiheen tuotoksia konferenssikirjassa (Bubenko et al. 1971) ja loppuvaiheen tuotoksia kirjassa Systemeering 75 (Lundeberg and Bubenko, Eds, 1975).

Eräs pohjoismaisen systeemin suunnittelun koulukunnan tulos oli se, että professori Börje Langefors vuonna 1974 Tukholmassa pidetyn IFIPin maailmankongressin yhteydessä valittiin IFIPin teknisen komitean 8 (Information systems) ensimmäiseksi puheenjohtajaksi.

Oppikirjatuotanto Suomessa

1970-luvulla Suomessa oli huomattavaa oppikirjatuotantoa. Tietokoneyhdistyksen ATK-tutkimuskerho (1970) tuotti kirjan päätöstauluista. Itse kirjoitin operaatiotutkimuksesta (Järvinen, 1970) Suomen ensimmäisen oppikirjan ja lisäksi käänsin Olle Doppingin kirjan suomeksi (1971). Kirjoitin Pentti Kerolan ajatusten pohjalta kirjan Systemointi I (Järvinen, 1974). Mainitut Kerolan ajatukset julkaistiin kirjassa Systemointi II (Kerola ja Järvinen, 1975), joka Markku Nurmisen arvion mukaan oli käänteentekevä teos systeemin suunnittelun opetuksen ja tutkimuksen kannalta. Näyttää siltä, että suurin osa Systemointi II-kirjan jäsenyksistä ja ajattelun välineistä on vieläkin täysin kelpoisia. Kirjoitin vielä Systemointi III-kirjan (Järvinen, 1975), joka käsitteli tiedonhallintaa. Atk-instituutin työryhmä Miikka Jahnukaisen johdolla laati hienon oppikirjan (1974), joka kansikuvan perusteella sai nimen ”naula päähän”-kirja. 1960- ja 1970-luvun tietojenkäsittelyn mahtiteikijä valtiohallinnon puolella, Valtion tietokonekeskus (VTKK) laati oman tietosysteemin valmistamisen vaihejaon (1976). Lukisin myös Lauri Fontellin kirjan (1975) Gobil-ohjelmoinnista tärkeäksi oppikirjaksi. Saimme Kerolan kanssa olla mukana, kun Mauri Leppänen kirjoitti (1977) johdatus tietojenkäsittelyyn kirjan.

Alan väitöskirjatuotanto alkaa

Tieteen saralla ilmestyi muutamia väitöskirjoja ja muita opinnäytteitä. Oma väitöskirjani (Järvinen, 1973) kuuluu verkkoteorian tai operaatiotutkimuksen alaan, vaikka sisältääkin tietokoneella toteutettujen algoritmien tehokkuuskokeita. Markku Nurmisen väitöskirja (Nurminen, 1976) kuuluu keskeisesti systeemin suunnittelun alaan. En ole missään väitöstilaisuudessa pelästynyt niin paljon kuin Nurmisen tilaisuudessa, kun heti väitöksen päätyttyä kajahti valtava torviorkesterin tuottama fanfaari. Musiikin harrastajana ja ensimmäisen tietokonetangon säveltäjänä tunnetun Markku Nurmisen muusikkoystävät soittivat mainitun fanfaarin. Difo-ryhmästä Vesa Savolainen väitteli (Savolainen, 1977) levyn hakuvarren liikkeen optimoinnista. Matti Koiranen sai (Koirane, 1978) valmiiksi tietojenkäsittelytoimintaa ja sen johtamista koskevan lisensiaatin tutkimuksensa.

Atk:n sosiaaliset vaikutukset 1970-luvulla

Atk:n sosiaalisia vaikutuksia ei Suomessa oltu 1970-luvun alussa tutkittu samassa määrin kuin muissa Pohjoismaissa. Norjassa oli aivan 1960- ja 1970-lukujen taitteessa käynnistynyt Kristen Nygaardin johdolla Iron and Metal-projekti. Sen innoittamana aloitettiin Ruotsissa 1975 Demos-projekti ja Tanskassa DUE-projekti (ks. Sandberg, 1979). Suomessa kyllä tunnistettiin työolojen ongelmia (1976) ja hiukan tutkittiin automaation vaikutuksia (Rantalaiho ja Heiskanen, 1976) sekä toimistotyön pulmia (Järvinen ja Kirjonen, 1977). Professori Juhani Kirjosen, Anneli Eteläpellon ja Pekka Tyllilän

kanssa tutkimme sitten kolmen vuoden ajan intensiivisesti tietosysteemin käyttöönottoa eräässä toimistossa. Järjestimme käyttäjille myös oman ohjelmointikurssinkin.

Pohjoismainen IRIS-konferenssisarja alkaa

Ehkä kansallisesti merkittävin hanke oli IRIS-konferenssisarjan (Information Systems Research Seminar in Scandinavia) käynnistäminen vuonna 1978 Tampereella. Se on toistaiseksi pisimpään jatkunut alan konferenssisarja maailmassa, sillä ICIS-konferenssit käynnistettiin vasta 1980 ja ECIS-konferenssit paljon myöhemmin. Sain olla Iris-konferenssien käynnistyksessä mukana Pentti Kerolan, Markku Nurminen, Eero Peltolan, Kalle Lyytisen ja Juhani Iivarin kanssa (vain muutamia nimiä mainitakseni). Markku Nurminen on Mr. IRIS, sillä hän on osallistunut kaikkiin 25 konferenssiin. Konferenssin luonne on säilynyt samana. Pääsylippu konferenssiin on oma artikkeli, jota käsitellään seniori- ja junioritutkijoiden ryhmässä varsin pitkään. Tarkoitus on antaa kirjoittajalle konstruktivisia ohjeita artikkelin parantamiseksi.

Ensimmäisessä Iris-seminaarissa skandinaaveja edustivat Mats Lundeberg ja Göran Goldkuhl. Neljä ensimmäistä konferenssia järjestettiin Suomessa (Tampere, Turku, Jyväskylä ja Oulu). Sen jälkeen on noudatettu periaatetta, jonka mukaan konferenssi on joka neljäs vuosi Ruotsissa, Norjassa, Tanskassa ja Suomessa. Sen lisäksi, että pohjoismaiset tutkijat tapaavat toisensa, konferenssi on alkanut kiinnostaa myös pohjoismaiden ulkopuolisia tutkijoita. Niinpä Italiasta, Saksasta ja Englannista on viime vuosina ollut mukana varsin suuria ryhmiä ja Australiasta astikin on tullut muutamia. Konferenssin Oulun tapaamisessa 1981 päätettiin perustaa Tietojenkäsittelytieteen seura, joka virallisestiperustettiin Helsingin yliopiston juhlasalissa vuonna 1982. Myös

arvostettu aikakauslehti Scandinavian Journal of Information Systems sai alkunsa IRIS-konferenssisarjan seurauksena.

Lähteet

Päätöstaulut (1970) Tietokoneyhdistyksen julkaisuja No 8, Helsinki.

Tietosysteemin valmistamisen vaihejako (1976) VTKK:n julkaisusarja A:2, Helsinki.

Työolojen tutkimisen teoreettisia lähtökohtia (1976) Työsuojelututkimuksen jaosto, Suomen Akatemia.

ATK-instituutin työryhmä (1974) Tietosysteemin rakentaminen, Tietojenkäsittelyliiton julkaisu no 25, Helsinki.

Bubenko, Källhammar, Langefors, Lundeberg och Sölvberg (1970) Systemering 70, Studentlitteratur, Lund.

Bubenko, Langefors and Sölvberg (Eds.) (1971) Computer-aided information systems analysis and design, Studentlitteratur, Lund.

Dopping ja Järvinen (1971) Lyhyesti ja ytimekkäästi ATK:sta, OtaDATA, Espoo.

Fontell (1975) Gobol-ohjelmointi, imes ry, Helsinki.

Järvinen (1970) Johdatus operaatiotutkimukseen, Tietoneyhdistyksen julkaisuja No 9, Helsinki.

Järvinen (1973) On a routing problem in a graph with two types of costs associated with the edges, Acta Universitatis Tampensis A 50, Tampereen yliopisto.

Järvinen (1975) Systemointi III – tietokannanhallintajärjestelmät, Gaudeamus, Helsinki.

Järvinen ja Kerola (1974) Systemointi I – käytäntö tietosysteemin rakentamisessa, Gaudeamus, Helsinki.

Järvinen ja Kirjonen (1977) Työkokonaisuus ja sen suunnittelu ihminen-kone-järjestelmässä, Matemaattisten tieteiden laitos C-1, Tampere.

Kerola ja Järvinen (1975) Systemointi II – tietosysteemin rakentamisen ja käytön systeemiteoreettinen malli, Gaudeamus, Helsinki.

Koiranen (1978) Tietojenkäsittelytoiminta ja sen johtaminen, Helsingin kaupapakorkeakoulun julkaisuja B-29, Helsingin kaupapakorkeakoulu.

Leppänen, Järvinen ja Kerola (1977) Johdatus tietojenkäsittelyyn – tietojärjestelmien hyväksikäytön näkökulma, Tietojenkäsittelyliiton julkaisu no 37, Helsinki.

Lundeberg and Bubenko (Eds.), Systemering 75, Studentlitteratur, Lund. Nurminen (1976) Studies in systemering on fuzziness in the analysis of information systems, Publications of the Institute for Applied Mathematics No 9, University of Turku.

Rantalaiho ja Heiskanen (1976) Automaatio ja teollisuustyöntekijät, Tampereen yliopiston sosiologian ja sosiaalipsykologian laitos No 17, Tampere.

Sandberg Å. (1979) Computers dividing man and work, Arbetslivcentrum, Stockholm.

Savolainen (1977) File assignment and sequencing on disc storages, Department of Mathematical Sciences Report A-12, University of Tampere.

*Pertti Järvinen,
professori
Tampereen Yliopisto,
Tietojenkäsittelyn laitos*

Tietojärjestelmäprosessi-innovaation omaksuminen, oppiminen, poisoppiminen, sovittaminen ja evoluutio organisaatiossa: pitkittäistapaustutkimus

*Erja Mustonen-Ollila,
Lappeenrannan teknillinen
korkeakoulu,
Tietotekniikan osasto*

Tausta

Tätä tutkimusta on tehty vuodesta 1995 lähtien ja sen tavoitteena on selvittää, miten menetelmien eli tietojärjestelmäprosessi-innovaatioiden omaksuminen, sovittaminen, oppiminen, poisoppiminen ja evoluutio organisaatioissa tapahtuu.

Tutkimus on pitkittäistutkimus menetelmien omaksumisesta kolmessa suomalaisessa organisaatiossa. Tutkimme tietojärjestelmien analysoinnissa, suunnittelussa, testauksessa, toteutuksessa ja projektien hallinnassa käytettyjen tietojärjestelmäprosessi-innovaatioiden valintatekijöitä, oppimismekanismeja, poisoppimismekanismeja, sovittamismekanismeja ja evoluutiota 43 vuoden ajalta. Aikakaudet jaettiin neljään aikaperiodiin: 1) 1949-1965, 2) 1965-1983, 3) 1983-1991 ja 4) 1991-1997 Friedmannin ja Cornfordin (1989) mukaan. Tutkittavat 22 tietojärjestelmää olivat konsernitason tie-

tojärjestelmiä. Lisäksi tutkittiin, miten ulkoistaminen vaikutti tietojärjestelmäprosessi-innovaatioiden päättäjiin ja heidän keskinäisiin suhteisiinsa.

Toivomme tutkimustulosten autavan organisaatioita ymmärtämään monimutkaisten toimintatapojen kehittymisen ja sen, että tehty tietojärjestelmäprosessi-innovaatiopäätös vaikutti pitkälle organisaatioiden kehitymisessä tietojärjestelmäorientoituneiksi.

Pääkäsitteet

Menetelmiä voidaan luonnehtia tietojärjestelmäprosessi-innovaatioina, joihin sisältyvät teknologiset menetelmät ja välineet, tiedonhallintamenetelmät, tietoliikenneverkon hallintamenetelmät, systeemin kuvaustekniikat ja projektinjohtamisvälineet ja -menetelmät. Tämä on josain määrin laajempi käsite kuin normaalisti käytetty, mutta me pidämme tätä tärkeänä sen vuoksi, että nämä kaikki vaikuttavat siihen tapaan, jolla systeemyön laatua voidaan parantaa. On myös tunnettu tosiseikka, että välineet ja menetelmät ovat yhä enem-

män kiinteästi sidoksissa toisiinsa, eikä niitä voida erottaa toisistaan. Esimerkiksi CASE- väline on hyvä esimerkki tästä. Projektijohtamisvälineet on sidottu taas tiettyyn prosessi-malliin.

Menetelmien omaksumispäätös eli adoptiopäätös tarkoittaa päätöstä ottaa menetelmä käyttöön. Päätökset taas tehdään organisatorisessa yksikössä, jota kutsutaan lokaaliksi. Lokaali on fyysinen ympäristö, jossa ihmiset kehittävät yhdessä tietojärjestelmiä tekemällä systeemyötä. Lokaali voi olla yksittäinen, muodollinen organisatorinen yksikkö tai useita yksiköitä tai osa yksiköstä, jos se on hajautettu. Tutkimuksessamme on kolme lokaalia A, B ja C, jotka sijaitsevat Imatralla, Helsingissä ja Lappeenrannassa.

Päätöksentekijä määrittellään henkilöksi, jolla on auktoriteetti eli valta päättää menetelmän käyttöönotosta. Menetelmät jaetaan neljään aika-kauteen: 1) 1949-1965, 2) 1965-1983, 3) 1983-1991 ja 4) 1991-1997 (Friedmann ja Cornford, 1989). Aikakaudet perustuvat 1) teknologisiin muutoksiin laitteistoissa ja ohjelmistois-

sa, 2) tietojärjestelmätyyppien muutokseen ja 3) käyttäjäsuhteiden muutokseen systeemityössä. Menetelmät jaettiin neljään menetelmäluokkaan: 1) projektinhallinta ja projektinjohtamisen menetelmät; 2) kuvaustekniikat, standardit, notaatiot, esimerkiksi tietovuokaaviot, seinätekniikka, UML-menetelmät; 3) kehitystyökalut, kuten sovellusgeneraattorit, CASE-työkalut, tietohakemistot tai työkalut, joilla hallitaan ohjelmistokehitystä ja 4) teknologisiin menetelmiin, joita ovat ohjelmointikielet, tiedonhallintamenetelmät ja tietoliikenneverkonhallintatyökalut.

Teoreettinen viitekehys

Kehitimme kuvan 1 mallin tutkimustamme varten. Kuvassa olevat käsitteet ovat teknologiamuutokset, ulkoinen tiedonsiirto, sisäinen oppiminen, omaksumispäätökset, teoreettinen diskurssi ja teoreettiset viitekehykset, kuten organisaatio-oppiminen ja innovaatioiden diffuusio. Empiiristä tutkimusta diskurssien kehityksestä ei ole, joten jätämme se tutkimuksen ulkopuolelle.

Tekijöitä, jotka johtavat menetelmien omaksumiseen voidaan tarkastella diffuusioteorialla (Rogers, 1995). Menetelmän omaksuminen on todellinen ilmiö ja diffuusio-innovaatioteoria on teoreettinen kehikko sen tulkitaan. Käyttöä ja käyttöönottoa, oppimista, voidaan tarkastella organisaatio-oppimisen kautta. Organisaatiot oppivat sisäisestä kokemuksesta ja ulkoa siirretystä tietämyksestä. Lisäksi organisaatiot etsivät ja tunnistavat uutta tietämystä. Organisaatiot muistavat ja siirtävät rationaalisesti tai jollain muulla poliittisella strategialla tietoa joko sisäisesti tai ulkoisesti (Huber, 1991).

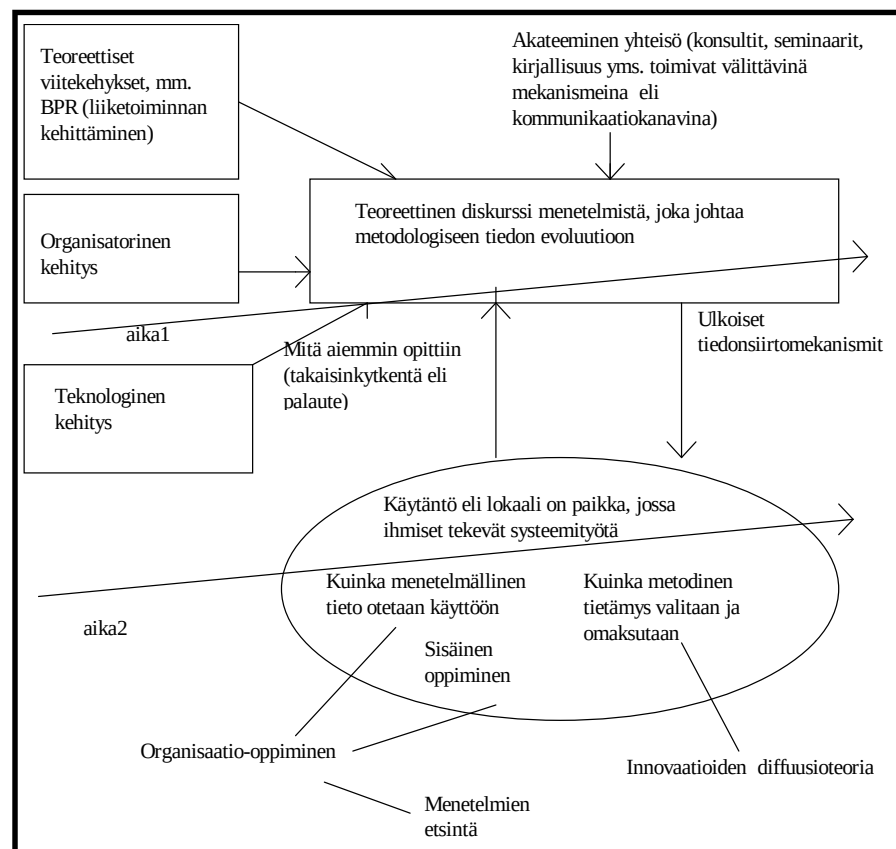
Teoreettisessa diskursissa syntyy menetelmiä. Diskurssin kehittymisen voidaan selittää teknologisella

kehityksellä, organisaation kehittymisellä tai teoreettisten viitekehysten kautta, kuten liiketoimintojen kehittymisellä. Lokaalissa tehdään käytännön systeemityötä. Teoreettisen diskurssin ja lokaalin välissä ovat ulkoiset tiedonsiirtomekanismit. Lokaalin sisällä valitaan ja omaksutaan menetelmiä ja otetaan menetelmällinen tieto käyttöön sisäisesti. Lokaalin oppimiseen vaikuttavat myös ulkopuoliset tiedonsiirtomekanismit. Lokaalista on takaisinkytkentä eli palaute teoreettiseen diskurssiin.

Tutkimuskysymykset

Tutkimme menetelmätietämyksen omaksumista, tietämyksen siirtämistä, tietämyksen etsintää ja poisoppimista kolmessa lokaalissa. Lisäksi tutkimme toimijoiden ja ulkoistamisen vaikutusta edellämainittuihin prosesseihin. Etsimme vastauksia seuraaviin tutkimuskysymyksiin:

1. Mitkä tekijät selittävät menetelmien omaksumista. Riippuvatko nämä tekijät siitä, minkä tyyppisiä menetelmiä valitaan? Riippuvatko tekijät siitä, milloin valinta on tehty ja onko teknologisilla sukupolvilla merkitystä? Kenellä on valta päättää menetelmien omaksumisesta?
2. Mitä mekanismeja käytetään menetelmätietämyksen välittymiseen? Ovatko mekanismit sisäisiä ja/tai ulkoisia? Miten menetelmätiedon kerääminen ja välittyminen on organisoitu? Miten menetelmät löydetään? Mikä on mekanismien suhde toisiinsa? Onko erilaisten innovaatioiden suhteen eroja siinä, minkälaisia etsintä- ja siirtomekanismeja käytetään ja ovatko mekanismit muuttuneet?
3. Mikä oli ulkoistamisen vaikutus menetelmätietämykseen? Millä



Kuva 1. Menetelmien teoreettinen diskurssi, käytäntö ja vuorovaikutukset.

lailla organisaatiorakenne ja menetelmätietämyksen välittyminen kehittyi ulkoistamisen takia? Miten systeemytön luonne muuttuu tietämyksen siirron yhteydessä? Kuka hallitsee menetelmätietämystä?

4. Miten menetelmät otetaan käyttöön eli sovitetaan organisaatioon?
5. Missä vaiheessa menetelmät häviävät organisaatioista? Mitkä tekijät ovat johtaneet siihen?

Tutkimusmetodologia

Valitsimme case-pohjaisen tutkimusmetodologian (Yin, 1993) kolmen organisaation tutkimiseen. Organisaatioiden johtamisjärjestelmä ja taloudellinen ympäristö oli sama, joten taloudelliset vaihtelut vaikuttivat kaikkiin yhtä paljon. Ulkoisia tekijöitä, lokaalia, aikaa ja menetelmäluokkaa voidaan tällöin kontrolloida. Johtamispolitiikka yrityksissä on ollut suhteellisen hajautettu.

Tarkoituksena on testata ja laajentaa olemassaolevia teorioita laadullisella pitkittäistutkimuksella ajan suhteen. Paneutuminen yhteen vertikaaliseen organisaatioon ei ole välttämättä järkevää, koska silloin ei voida katsoa näiden menetelmävalintojen eroja erilaisissa lokaaleissa, joka on yksi muutuja. Horisontaalasti tutkimuksessamme on kolme havaintosarjaa. Yritämme testata mallilla sekä väitteiden paikanpitävyyttä ja sitä kautta eri teorioita sekä käytämme mallia generoimaan joukon kysymyksiä tai väitteitä, jotka nousevat aineistosta empiiriseen dataan perustuvan tutkimusotteen mukaisesti.

Kvalitatiivinen aineisto koostuu haastatteluista ja arkistomateriaalista ja historiallisen aspektin takia historialliset tutkimusmenetelmät ovat perusteltuja. Kun tutkimme historiallista

aineistoa, niin meillä ei ollut täydellisiä havaintopistettä, koska oli vaikeaa rekonstruoida mennyttä aikaa. Havainnot koottiin haastatteleamalla eri organisaatiossa olevia ihmisiä: systeemisunnittelijoita, projektipäälliköitä, menetelmistä vastaavia henkilöitä ja menetelmäryhmäpäälliköitä, ohjelmiojia, tietohallintojohtoa, systeemipäälliköitä, osastopäälliköitä, toimitusjohtajia, asiakkaita, käyttäjiä, systeeminhoitajia ja liiketoiminnoista vastaavia johtajia. Tiedot kerättiin arkistomateriaaleista, kenttähavainnoista, systeemidokumentaatioista, henkilökohtaisista arkistoista ja julkisista arkistoista. Lehdiä ja muuta kirjallisuutta käytettiin parantamaan haastatteluissa saadun tiedon oikeutta ja luotettavuutta. Analysoitava tieto oli joukko laadullisia, retrospektiivisiä, prosessuaalisia, historiallisia ja aikaan sidottuja tietojoukkoja. Kerättävä aineisto koostui 22 erilaisesta konsernitason tietojärjestelmästä, jotka oli rakennettu vuosina 1954-1997.

Haastattelumateriaalin ja arkistomateriaalin validiteetti ja reliabiliteetti huomioitiin käyttämällä erilaisia tiedonlähteitä ja data sovitettiin muuttujiin, jotka on jo validoitu aiemmissa muiden tekemissä tutkimuksissa (Wolfe, 1994).

Omaksumispäätökset luokiteltiin menetelmäluokkiin, aikaperiodeihin ja lokaaleihin. Myös menetelmien tiedonsiirtomekanismit, etsintämekanismit ja poisoppimismekanismit luokiteltiin vastaaviin luokkiin. Päätöksentekijät luokiteltiin omiin luokkiin auktoriteetin mukaan.

Rogersilaisessa diffuusio-innovaatioteoriassa (Diffusion of Innovations eli DOI) (1995) tutkitaan omaksumistekijöitä, jotka vaikuttavat omaksumispäätöksiin (Sauer ja Lau, 1997). Omaksumistekijät voidaan jakaa pienempiin tekijöihin, joita kutsutaan ite-

meiksi (Prescott ja Conger, 1995). Tieto reflektoidaan teoreettisen DOI mallin lävitse.

Vastaavasti tieto reflektoitui organisaatio-oppimisteorian, valtateorian ja aiemman kirjallisuuden kautta.

Tilastolliset testit ja tulosten tulkinta

Tilastollinen testi on menetelmä, jolla vahvistetaan tai kumotaan tutkimuskysymyksen eli hypoteesin arviointi. Testasimme oliko tilastollisesti merkitseviä eroja siinä millä tavalla eri aikakausina omaksuttiin menetelmiä, mitä menetelmiä eri aikakausina omaksuttiin tai olivatko omaksuttujen menetelmien tyypit tilastollisesti merkitsevästi erilaisia eri lokaaleissa.

Tilastollisina testeinä eri tutkimuskysymyksissä käytettiin sisältöanalyysiä, X^2 -riippumattomuustestiä, itseorganisointia kartoja, ehdollista todennäköisyyslaskentaa, lineaarista korrelaatiota ja Bayesian mallinnusta.

Aineistosta löytyneet uudet tekijät, joita ei teoria ei tunne, laajensivat teorioita.

Teoriakehikkoa sovellettiin aineistoon ja näistä saatiin yleistyksiä ja abstraktioita, jotka tulkittiin. Yleistyksiä ja abstraktioita käytettiin viittaamaan takaisin käytettyyn teoriaan. Tulkintana käytettiin heuristista tulkintaa.

Tutkimustulokset

Menetelmien omaksuminen

Yksi näkökulma menetelmien kehitykseen on diffuusio-innovaatioteoria (DOI), koska menetelmää voi tarkastella innovaationa. Menetelmien omaksumiseen DOI kirjallisuuden

mukaan vaikuttaa erilaisia tekijöitä, faktoreita. Valitsemmamme faktorit ja itemit ovat 1) innovaatiofaktori, itemeinä helppokäyttöisyys tai standardi, 2) yksilölliset faktorit, itemeinä omat kokeilut tai itsenäinen työ tai oma oppiminen, 3) tehtäväfaktorit, iteminä käyttäjätarpeiden huomiointi, 4) ympäristöfaktorit, iteminä teknologinen infrastruktuuri ja 5) organisatoriset faktorit, iteminä aiempi teknologinen kokemus. Kutakin faktoria mittasimme eri itemeillä. Tässä tutkimuksessa tarkastellaan menetelmien omaksumista siinä mielessä, että tutkitaan niitä tekijöitä, jotka vaikuttavat innovaation valintaan.

Tutkimuksessa todettiin, että tärkeimmät tekijät ovat käyttäjätarpeen tunnistaminen, teknologisen infrastruktuurin olemassaolo, aiempi kokemus, omat kokeilut, työn itsenäisyys, helppokäyttöisyys, oppiminen itse tekemällä ja standardit. Kuitenkin iso joukko menetelmäomaksumisia ei noudattanut mitään erityistä hahmoa.

Projektiryhmät päättivät menetelmien omaksumista. Ulkoinen oppiminen ei ollut niin yleistä, paitsi jos kyseessä oli teknologinen menetelmä. Useimmissa tapauksissa sisäinen oppiminen oli tärkeämpää. Näissä kolmessa organisaatiossa omaksumiset vaihtelivat suuresti oppimismekanismien, teknisten laitealustojen ja kehitysympäristöjen mukaan.

Menetelmien etsintä- ja tiedonsiirtomekanismit

Systeemityö organisaatiossa on tietty joukko rutiineja, joissa ihmiset etsivät koko ajan uusia menetelmiä johtuen nykyisen tietämyksen ja uuden tietämyksen välisestä aukosta. Tutkimuksessa meitä kiinnosti, mitä mekanismeja eri toimijat käyttävät tiedon siirtämiseen osapuolelta toiselle. Tämän vuoksi tarkensimme Huberin

yleistä teoriaa oppimisesta käsittämään myös tiedonsiirtomekanismit tarkemmin. Attewellin (1992) mukaan tiedonsiirtomekanismeilla on ollut merkittävä vaikutus siihen, kuinka hyvin innovaatiot on otettu käyttöön. Organisaatiot, jotka siirtävät tietämystä, investoivat isoja määriä resursseja ja rahaa tietämyksen siirtoon käyttäjäorganisaatioon.

Tutkimustulokset osoittivat, että organisaation omat kokeilut olivat tärkein etsintämekanismi 1980-luvun alkuun saakka, mutta sen jälkeen 1990-luvun loppuun saakka tietoisesta etsinnästä tuli tärkein mekanismi. Projektiryhmät toimivat tärkeimpänä sisäisenä tiedonsiirtomekanismina vuodesta 1965 1990-luvun loppuun. Laitetoimittajat siirsivät ohjelmointikielten, tiedonhallintamenetelmien, laitteiden ja varusohjelmistojen tietämyksen organisaatioihin. Konsultit siirsivät projektin hallintamenetelmätietämyksen ja systeemin kuvaustekniikat organisaatioihin.

ATK-osaston ulkoistaminen

Ennen ulkoistamista vuonna 1984 lokaalissa A oli keskitetty auktoriteetti ja sekä asiakkaan että suunnittelijan piti käyttää samaa menetelmää. Ulkoistamisen jälkeen auktoriteetti hajautui. Tutkimustulokset osoittivat, että ennen ulkoistamista 1984 päätösvalta oli sisäisellä atk-osastolla ja sisäisellä asiakkaalla. Ulkoistamisen jälkeen päätösvalta muuttui keskiteytystä hajautetuksi, jossa ulkoinen asiakas, kaksi ohjelmistotoimittajaa ja asiakkaan oma sisäinen tietotekniikan osasto käyttivät yhdessä päätösvaltaa.

Menetelmien sovittaminen organisaatioon

Vallalla tarkoitetaan auktoriteettiä ja suoraa ohjausta, jossa henkilö käyttää valtaansa muiden ylitse ja seu-

raa päätöstensä toteutumista (Orlikowski, 1991). Henkilöllä voi olla myös sisäisiä riippuvuussuhteita toisiin päättäjiin.

Tutkimuksessa löydettiin kymmenen päättäjää, jotka päättivät menetelmistä ja olivat sisäisesti riippuvuussuhteessa toisiinsa. Ennen ulkoistamista 1984 atk-osaston projektiryhmät ja projektin ohjausryhmät, käyttäjät ja atk-ammattilaiset olivat riippuvia toisistaan päätöksenteossa monimutkaisten valtarakenteiden takia. Ulkoistamisen jälkeen järjestelmien toimittajaorganisaatio päätti teknologisten menetelmien käytöstä. Valtarakenteet siirtyivät tuottajaorganisaatioon ihmisten myötä.

Menetelmien poisoppiminen

Organisaatioihin tulee koko ajan uudenlaisia tietämystä, menetelmiä ja teknologioita, jotka samojen resurssien on pystyttävä oppimaan ja hallitsemaan. Tämän vuoksi organisaation pitää pystyä unohtamaan vanhat tai jopa käytössä olevat menetelmät.

Tutkimuksessa löysimme viisi tärkeää poisoppimismekanismia: uuden tiedon luonti, huono suorituskyky, reagointi ympäristömuutoksiin, muutokset johtajuudessa ja vaikeakäyttöisyys. Mekanismit vaihtelivat ajan ja menetelmäluokan mukaan. Teknologinen kehitys ja mikrotietokoneiden esiintulo 1980-luvun alussa muuttivat systeemien kehitystyötä ja uusia työtapoja tarvittiin. 1990-luvun alussa teknologiset laitealustat, käyttöjärjestelmät, tietokannat ja systeemien suunnittelutyötavat muuttuivat oliopohjaisiksi ja vanhoista menetelmistä oli pakko oppia pois.

Lähdekirjallisuus

Attewell, P. (1992) Technology Diffusion and Organisational Learning:

The Case of Business Computing, Organization Science, Vol. 3, No. 1, pp. 1-19.

Friedman, A., Cornford D. (1989) Computer Systems Development: History, Organization and Implementation, John Wiley & sons, Information Systems Series.

Huber, G.P. (1991) Organizational Learning: The Contributing Processes and The Literatures, Organization Science, Vol. 2, No. 1, pp. 88-115.

Mustonen-Ollila, E. (1997) Baseline Story Data Manuscript, pp.1- 232.

Mustonen-Ollila, E. (1995) Methodologies Use and Their Evolution in Organisations, an abstract, CAISE *95, Information Systems Engineering: Current Practice and Future Prospects, Doctoral Consortium (2nd edition), 12-13 June, Jyväskylä, Finland.

Mustonen-Ollila, E. (1998) DIGIT meeting 12th of December 1998 Helsinki, Finland, Methodology Choice and Adoption: Using Diffusion of Innovations as the Theoretical Framework, belonging to the IFIP Working Groups 8.2 and 8.6 Joint Working Conference on Information Systems: Current Issues and Future Changes, December 10-13, Helsinki, Finland, December 1998; and Nineteenth Annual International Conference on Information Systems ICIS' 98, December 13-16 Helsinki, Finland. Available in the Internet.

Mustonen-Ollila, E. (1999) Methodology Choice and Adoption: Using the Diffusion of Innovations Theory (DOI) as the Theoretical Framework, Proceedings of the 22nd Information Systems research Seminar in Scandi-

navia (IRIS22): Enterprise Architectures for Virtual Organisations, 7-10 August, Keuruu, Finland, Vol. 2, Edited by Timo K. Käkölä.

Mustonen-Ollila, E., Lyytinen, K. (2002b) Why organizations adopt IS process innovations: a longitudinal study using the Diffusion of Innovation theory. Accepted for publication to Information Systems Journal (ISJ).

Mustonen-Ollila, E. (2001) The Impact of Information Technology (IT) Outsourcing, to IS processes, <http://www.it.lut.fi/opetus/00-01/010976000/sche.html>. Lappeenranta University of Technology, Department of Information Technology.

Mustonen-Ollila, E., Heikkonen, J. (2002a) Gatekeepers in Information System Process Innovation Adaptation: A Longitudinal Case Study. Proceedings in the PACIS2002 (The 6th Pacific International Conference of Information Systems), 2-4.9, 2002, Tokyo, Japan, pp. 1043-1047.

Orlikowski, W. (1991) Integrated Information Environment or Matrix of Control? The Contradictory Implications of Information Technology, Accounting, Management, and Information Technologies, Vol. 1, No. 1.

Prescott, M.B., Conger, S.A. (1995) Information Technology Innovations: A Classification by IT Locus of Impact and Research Approach, The Data Base for Advances in Information Systems, A Quarterly publication of SIGMIS, Special Issue of Technological Diffusion of Innovations, Vol. 26, Nos 2-3, Special Double Issue: of Diffusion Technological Innovation, pp. 20-41.

Rogers, E. M. (1995) Diffusion of Innovations, Fourth Edition, The Free Press.

Sauer, C., Lau, C. (1997) Trying to adapt systems development methodologies- a case-based exploration of business users' interests, Blackwell Science Ltd., Information Systems Journal, Vol. 7, pp. 255-275.

Wolfe, R.A. (1994) Organizational Innovation: Review, Critique and Suggested Research Directions, University of Alberta, Journal of Management Studies 3.

Yin, R.Y. (1993) Applications of Case Study Research, Applied Social Research Methods series, Vol. 34, SAGE publications.

Tutkimuksen tila

Tutkimusta ohjaavat TkT Jukka Heikkonen (Teknillinen korkeakoulu), professori Kalle Lyytinen (Case Western University, USA) ja professori Jouni Lampinen (Lappeenranta teknillinen korkeakoulu). Tutkimuksesta on laadittu seitsemän eritasoista julkaisua. Lisäksi kolme lehtiartikkelia ja yksi konferenssiartikkeli ovat arvioitavana. Tutkimus jatkuu edelleen ja tavoitteena on saada väitöskirja vuoden 2003 aikana.

Lisätietoja:

Erja Mustonen-Ollila,
Lappeenranta teknillinen korkeakoulu,
Tietotekniikan osasto,
e-mail: erja.mustonen-ollila@lut.fi,
<http://www.lut.fi/~emo>

Systemityö-yhdistys

Johtokunta 2002

sytyke-hallitus@pcuf.fi
http://www.pcuf.fi/sytyke

Kati Ahlgrén
Artemis Finland Oy
Maapallonkuja 1 A, 02210 Espoo
puh. 020 4362 000
ext-kati.ahlgren@nokia.com

Pekka Forselius
(puheenjohtaja)
STTF Oy (Software Technology Transfer
Finland Oy)
Porslahdentie 23 G 40, 00980 Helsinki
puh. 050-516 0416,
pekka.forselius@sttf.fi

Petteri Holländer
SysOpen Oy
Hiomotie 19, 00380 Helsinki
Puh. 0424 2020 457
petter.hollander@sysopen.fi

Pirkko Leivo
Pohjolan Systemipalvelu Oy
00013 Pohjola
puh. 010 559 2786, 050-567 9352
pirkko.leivo@pohjola.fi

Minna Oksanen
TietoEnator Digital Finance Oy
Kutojantie 10, PL 33, 02631 Espoo
puh. 09-8626 3272, 040-577 6640
minna.m.oksanen@tietoenator.com

Erkki Pöyhönen
Nokia Research Center,
Software Technology Laboratory
PL 407, 00045 NOKIA GROUP
puh. 07180 37595
erkki.poyhonen@nokia.com

Simo Vuorinen
TietoEnator Corp., DGSF/dGov
Tietotie 6, 02101 Espoo
puh. 040-524 8936
simo.vuorinen@tietoenator.com

Varajäsenet

Mustonen-Ollila Erja
Lappeenrannan teknillinen korkeakoulu
Tietotekniikan osasto
PL 20, 53851 Lappeenranta
puh. 05-621 2857
erja.mustonen-ollila@quicknet.inet.fi

Helena Venäläinen
FD Finanssidata Oy
Teollisuuskatu 1 B,
PL 308, 00101 Helsinki
puh. 09-404 3690, 050-568 6690
helena.venalainen@osuuspankki.fi

Liittokokousedustajat

- Lauri Laitinen
lauri.laitinen@nokia.com
- Silja Räisänen
silja.raisanen@pohjola.fi
- Pekka Forselius
pekka.forselius@sttf.fi
- Helena Venäläinen
helena.venalainen@osuuspankki.fi

Web-toimikunta

sytyke-web-toimikunta@pcuf.fi

- Helena Venäläinen (pj)
- Heini Holopainen
- Eija Kalliala
- Lauri Laitinen
- Juha Mäkeläinen
- Agnieska Nummi
- Erkki Pöyhönen

SYTYKE-kerhot

- Testauskerho (FAST)
erkki.poyhonen@nokia.com
- JavaSIG
simo.vuorinen@tietoenator.com
- DT-kerho (Datanomit ja tradenomit)
jani.vesterinen@surfeu.fi
- Relaatiokerho (RELA)
marja.karmeniemi@oracle.com

Toimisto, sihteeri, Systemityö-lehden tilaukset ja talouden hoito

Anne-Maj Viio (viio@vt.inet.fi)
Henrikintie 7 A, 00370 Helsinki
puh. 09-5607 5363, fax. 09-5607 5365

Jäsenyysasiat

Jäsenneksi liittymis- tai jäsentietojen muutos-
lomakkeiden tilaus:
Tietotekniikan liitto ry/Piia Heiniö
Lastenodinkuja 1 A (5.krs),
PL 325, 00181 Helsinki
puh. 09-476 5800, http://www.tt-tori.fi

Systemityö-lehti

Julkaisija

Systemityöyhdistys SYTYKE ry
ISSN 1237-0525
10. vuosikerta no 1.
Painos 2500 kpl.
Paino: Tuokinprint Oy
Sirrikuja 4 C, 00940 Helsinki
puh. 09-342 4150
tuokinprint@co.inet.fi

Toimitussihteeri

Susanna Koskinen,
Toimistopalvelu Hennax Tmi
Tikkurikuja 10 T, 00750 Helsinki
puh. 09-5617 830
fax. 09-388 2069
susanna.koskinen@hennax.inet.fi

Päätoimittaja

Lauri Laitinen
Nokia Tutkimuskeskus
PL 407, 00045 Nokia Group
Itämerenkatu 11-13, 00211 Hki,
puh. 07180 36551 (NRC)
Tornitaso 3 A 37, 02120 Espoo
puh. 09-465 959
lauri.laitinen@nokia.com

Toimituskunta 1/2003

Erja Mustonen-Ollila ja Lauri
Laitinen

Seuraava numero:

2/2003 Koulun penkiltä sorvin
ääreen

Toimituskunta: Pirkko Leivo ja
Kati Ahlgrén

Tilaukset

Systemityölehti sisältyy yhdis-
tyksen Tietotekniikan liiton suositus-
ten mukaiseen yhdistyksen jäsen-
maksuun, joka vuonna 2003 on 62,
42, 25, 10 euroa/vuosi. Vuositilaus
10 euroa. Irtonumerot 5 euroa. Hy-
vissä ajoin ennen lehden painatusta

Ilmoitushinnat

	mv	4 väriä
Takakansi	-----	1200 euroa
Sisäkannet (K1, K2)	-----	1000 euroa
Sisäsivut A4	400 euroa	800 euroa
Sisäsivut A5	200 euroa	600 euroa
Sisäsivut A6	100 euroa	-----

Vakiopaikan vähintään vuodeksi varanneille 20 % alennus.

tehty vähintään 50 kpl:een lisätilaus,
2 euroa/kpl. Tilaukset yhdistyksen
toimistosta.

Internetin WWW-sivut

http://www.pcuf.fi/sytyke
Systemityö-lehden sisällysluettelo
1988- ja useimmat artikkelit PDF-
muodossa.

