

Pääkirjoitus

Syksyt ovat meille IT-ihmisille perinteisesti kiireistä aikaa, joskin nykyisin yhä useammalla meistä näyttää olevan ikuinen syksy. Kiireet eivät hellitä ja toisinaan siihen ovat syynä puutteelliset hallinnan työkalut ja usein taas liian monet yhtäaikaiset hankkeet. Tästä syystä halusimme koota vuoden viimeisen SYTYKE-lehden ”Tietohallinto”-teeman alle artikkeleita, joissa käsitellään mm. prosesseihin, hankkeiden ohjaukseen, yhteistyöhön ja koulutukseen liittyviä teemoja.

Hyvin useat IT-alan asiantuntijalehdet koostuvat toimittajien näkemyksistä maailman menosta. Tähän lehteen halusimme saada myös asiakkaan näkemyksiä toimittajayhteistyöstä sekä hankkeiden ja hankesalkun hallinnasta.

Myös Gartner Groupin Executive Programmin vetäjä Pekka Sipola kertoo tietohallintojohdolle suunnatusta EXP-palvelusta, johon on liittynyt jo useiden Suomessa toimivien yritysten IT-johtajia. EXP-palvelu on toimiva ja helppo tapa päivittää omat tietonsa kiireen keskellä.

Lehdessä käsitellään myös paljon esillä ollutta relaatiotietokantojen ja oliomallinnuksen kohtalonyhteyttä. SYTYKE Ry:ssä aktiivisesti toimiva RELA-kerho julkaisee artikkeleita koskien sitä, miten oliomallinnus vaikuttaa relaatiotietokannan syntyyn.

Samaten lehdessä esitellään jo perinteisen SYTYKE-syysristeilyn satoa tekstein ja kuvin. Tällä kertaa risteilyn teemana oli aina yhtä ajankohtainen testaus ja laatu systeemityössä.

Hyvää loppuvuotta ja leppoisaa joulun odotusta!

Päivi Hokkanen, CIO, Stockmann Oyj

Johtokunta 2003

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 Systeemipalvelu 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

Projektin ohjauksesta hankkeen elinkaaren aikaiseen ohjaukseen

Sirpa Creutz,
VR

Projektien hallinta on Akilleen kantapää kaikille IT työtä tekeville, riippumatta siitä toimiiko asiakkaan vai toimittajan puolella. Projektin hallinnasta ja haasteista voi puhua monesta näkökulmasta. Tämän kirjoituksen takana on kokemus siitä, että vaikka tekniikka muuttuu hankkeiden ja projektityön hallinnan kokemusta tarvitaan aina. Kokemukseen kuuluu myös se, että asiakkaan puolella joudutaan useimmiten puhumaan laajemmista hankekokonaisuuksista, kun taas toimittajalle usein riittää hyvä projektityön hallinta. Edelleen voi todeta, että projektityössä on aina voimassa olevia hyviä periaatteita. Haasteen sen sijaan tuo jokaisen projektin kohdalla uusi kokonaisuus aikataulussa ja budjetissa hoidettavista asioista ja erityisesti aina uudelleen koottava tiimi tekijöitä, joilla on toisistaan poikkeava kokemus/näkemykset projektityöstä.

Hyvän projektityön menestymisen yksi osuus on hyvät toimintatavat ja niitä tukevat välineet. Hankekokonaisuuksien hallinnan osalta kysymys on hyvin paljon ajattelua helpottavasta etenemisprosessista, jolloin kunkin vaiheen kohdalla keskitytään vaiheelle olennaisiin kysymyksiin ja annetaan niihin vastauksia. Prosessi sisältää myös check-listoja ja sisällysluetteiloita, joihin ajattelun tulokset tulee dokumentoitua työn jatkamisen pohjaksi.

Tämän artikkelin kokonaisuus perustuu tähän laajempaan, hankkeen koko elinkaaren kattavaan ohjaustapaan ostavan yrityksen näkökulmasta.

Hankehallinnan taustaedellytyksiä

Koko hankehallinnan taustalla on hyvä liiketoimintastrategia, jonka suuntaviivojen mukaan voidaan rakentaa kehittämisohjelma valitulle ajanjaksolle. Jollei selkeää strategiaa ja siitä johdettavaa kehittämisohjelmaa ole, on hankkeen perusteluvaiheen kokonaisuus haasteellisempi, mutta ei mahdoton. Sen sijaan hankkeiden priorisointi liiketoiminnan näkökulmasta on tietenkin mahdoton, jollei suuntaviivoja strategian kautta ole saatavissa.

Hankkeen elinkaariajattelun mukaan on mahdollista hankkeen perusteluvaiheessa täydentää ja täsmentää strategiassa ja kehittämisohjelmassa olevia ylemmän tason tavoitteita.

Kehittämissuunnitelmissa olevisista töistä yhdistetään hanke, jonka tavoitteena on varmistaa, että tärkeideltään, laajuudeltaan ja/tai ohjattavuudeltaan vaativien kehittämistöiden tavoitteet saavutetaan. Hanke ei ole vain investointi, vaan se voi olla mikä tahansa asia, jota halutaan kehittää ja ohjata kehittämisen ennalta sovittavien vaiheiden kautta tulokseen.

Hankkeen valmistelu

Hankkeen elinkaaren ensimmäinen vaihe on varmasti idea, ongelma tai jo selkeästi määritelty tavoite kehittämisohjelmassa. Hankkeen valmisteluvaiheessa ensimmäisenä jonkun tulee istua alas ja miettiä, miten kysymykseen paneudutaan. Jotta asiaa ei tarvitsisi lähteä miettimään tyhjälle A4:lle, voidaan tukena käyttää esim. alustavan hankeperustelun perusrunkoa. Rungon mukaisesti ku-

vataan selkokielellä mitä tulee kehittää, mikä hanke on, mitkä ovat hankkeen perustelut ja mitkä ovat kehittämisen muutostavoitteet. Millaisia vaihtoehtoisia etenemistapoja kehittämiseen on, jotka tulisi tarkemmin kartoittaa? Yhtenä vaihtoehtona tarkastelussa tulee aina olla mukana nollavaihtoehto, ”ei tehdä mitään”, jotta nähdään todellisuudessa mikä vaikutus uudella kehittämisellä on. Alustavan hankeperustelun osana on myös työsuunnitelma vaihtoehtojen kartoittamiseen. IT-ihmisille voi todeta, että vaihtoehtot eivät hankkeessa ole eri tietojärjestelmiä, vaan todellisia vaihtoehtoja kehittämiseen, joista joku voi olla täysin irti tietojärjestelmistäkin.

Alustava hankeperustelu on siis tiivis kuvaus kehitettävästä asiasta ja siitä, miten ensimmäinen vaihe hankkeesta tehdään eli millä aikataululla ja millä resursseilla tuotetaan varsinainen hankeperustelu ja sen osana oleva investointiperustelu.

Alustavan hankeperustelun avulla saadaan tehokkaasti liikkeelle ne hankkeet, joiden nopea eteneminen on yrityksen kannalta tärkeää. Toisaalta voidaan siirtää dokumentoituina odotamaan ne ideat ja ongelmat, joihin nyt ei haluta panostaa resursseja.

Alustavan hankeperustelun tarkoituksena on mahdollistaa kehittämisasioiden tehokas valmistelu ja päätöksenteko. Johdon osalta voidaan valmistelua seurata panostuksien ja aikaansaannosten perusteella jo alkuvaiheessa. Jos on tarpeen, voidaan hankkeen johtoryhmä perustaa jo tässä vaiheessa. Alustavan hankeperustelun hyväksymisessä ohjataan voimakkaasti sitä, miten syvälle eri vaihtoehtojen kartoituksessa mennään.

Hankkeen valmistelun seuraavassa vaiheessa suurin työ käytetään eri vaihtoehtojen kartoitukseen. Kustakin vaihtoehdosta tulisi laatia vertailukelpoiset selvitykset, joiden tulisi sisältää toimenpiteet tavoitteiden toteuttamiseksi, hyödyt ja tuotot, kokonaiselinkaari- ja vuosikustannukset, riskit ja yhteenveto ko. vaihtoehtoista.

Valmisteluvaiheessa päätetään edellisten vaihtoehtojen perusteella mikä/mitkä vaihtoehdot ovat todennäköisimmät ja ko. vaihtoehdosta laaditaan hankeperustelu.

Hankeperustelua tehdessä täydennetään alustavaa hankeperustelua kerättyjen tietojen pohjalta. Hankeperustelun tulee edelleen sisältää täydennettynä hankkeen kuvaus, perustelu, muutostavoitteet, hyödyt ja tuotot, kustannukset, riskit, toteutuksen varmistusnäkökohdat sekä jälkiseurantasuunnitelma. Hankkeesta tulee olla karkea aikataulu ja organisointi.

Hankkeen kustannuksiin tulee laskea kaikki hankkeen aikaiset kustannukset. Helposti unohdetaan, että omat käytetyt resurssit, ihmiset, tilat ja laitteet ovat myös kustannuksia. Usein tähän kiinnitetään huomiota vasta, kun omat resurssit eivät ehdi tai osaa tehdä heille varattuja tehtäviä ja joudutaan hankkimaan resurssit täydennystä talon ulkopuolelta kalliilla hinnalla.

Hankeperustelun ja sen liitteiden (mm. muiden kartoitusten tulokset) avulla hankkeen päätöksentekijä voi esittää kysymyksiä ja täydennyksiä. Viimeistään hankeperustelun käsitteilyn yhteydessä päättäjä joutuu ottamaan kantaa käynnistetäänkö hanke vai ei. Hyväksyttäessä hanke saa käynnistyslupaa ja siirtyy toteutusvaiheeseen. Jos hanketta ei tässä vaiheessa hyväksytty, perustelut sen siirtämiseksi tulevaisuuteen tai hylkäämiseksi kokonaan löytyvät samaisesta hankeperustelusta.

Kun hankkeen valmisteluvaihe tehdään tiiviisti ja dokumentoiden hankeperusteluun, pääsee hanke vauhdikkaasti liikkeelle. Tässä vaiheessahan usein johdon ajankäyttö siirtyy vain johtoryhmän kautta työskentelyyn ja tavoitekapulaa lähtee jopa täyspäiväisestikin viemään hankevetäjä.

Hankkeen käynnistäminen ja toteuttaminen

Hankkeen suunnittelu on kuten projektin suunnittelua, mutta helikopteriperspektiivistä. Hankesuunnitelman tekemiseen tulisi uhrata riittävästi aikaa, koska siinä paloitellaan isonkin kokonaisuuden työt projekteihin, vaiheistetaan ne synkronoitumaan yhteen, pohditaan organisointia, oikeiden osaajien (erityisesti yrityksen sisäisten) saatavuutta, aikatauluja, tarkempaa budjettia, riskejä, hallinta- ja laadunvarmistusmenetelmiä.

Tämä pohjatyö on hankevetäjän näkökulmasta erityisen tärkeää, jos hanke joudutaan jakamaan useisiin alaprojekteihin (esim. toiminnan muutos, tietojärjestelmä...). Mitä paremmin kokonaisuuden jako on hankevetäjällä selvänä, sitä paremman lähtökohdan saavat osaprojektien vetäjät omissa kick-offeissaan. Hankevetäjän on johdon avustuksella hankkeen käynnistysvaiheessa panostettava erityisesti hankkeen tavoitteiden ja hankesuunnitelmissa annettujen periaatteiden viestittämiseen kaikille hankkeessa mukana oleville. Jos tässä ei onnistuta kyseenalaistaminen ja epäselvyydet jatkuvat koko hankkeen ajan.

Hankkeen toteuttamisen aikana osa töistä tehdään projektimuotoisena osa työsuunnitelman mukaan erillisinä hanketöinä. Projektimuotoinen työ on aina yhtä haastavaa, mutta se alkaa olla sekä ostaja että toimittaja puolella onneksi käsitteellisesti laajempi, kuin vain janakaavio aikataulusta, siksi en käsittele sitä hankkeen elinkaaren yhteydessä tarkemmin.

Hankkeen toteuttamisen aikana on oleellista, että hanketta seurataan kokonaisuutena erikseen. Hankejohtoryhmän tehtävänä on seurata koko hankkeen etenemistä tavoitteiden suuntaan annettujen resurssien ja aikataulun puitteissa. Hankevetäjän tehtävänä on tilanneraportissaan hankkeesta tuoda yhteenvetona esille kaikki seurattavat asiat ja johtoryhmän tulee erityisesti pitää huolta, että kaikkia projekteja seurataan yhtälailla. Jos järjestelmän rakentamisprojekteilla onkin aina tekniset ongelmansa, saatetaan hankkeen kannalta haasteellisimmaksi muodostuakin uuden toimintamallin mukaiset toiminnan muutokset sisäisesti johdetussa projektissa ja samalla tiukalla aikataululla. Hankkeen kokonaisseuranta johtaa usein esim. sisäisten töiden osalta tuntiraporttien täyttämiseen samalla tavalla kuin toimittajilta vaaditaan. Kuten aikaisemmin ehdotin, hankkeen johtoryhmä kannatta perustaa jo valmisteluvaiheessa, jolloin toteuttamisvaiheessa johtoryhmän työskentely on jo vakiintunutta toimintaa hanketason näkökulmasta.

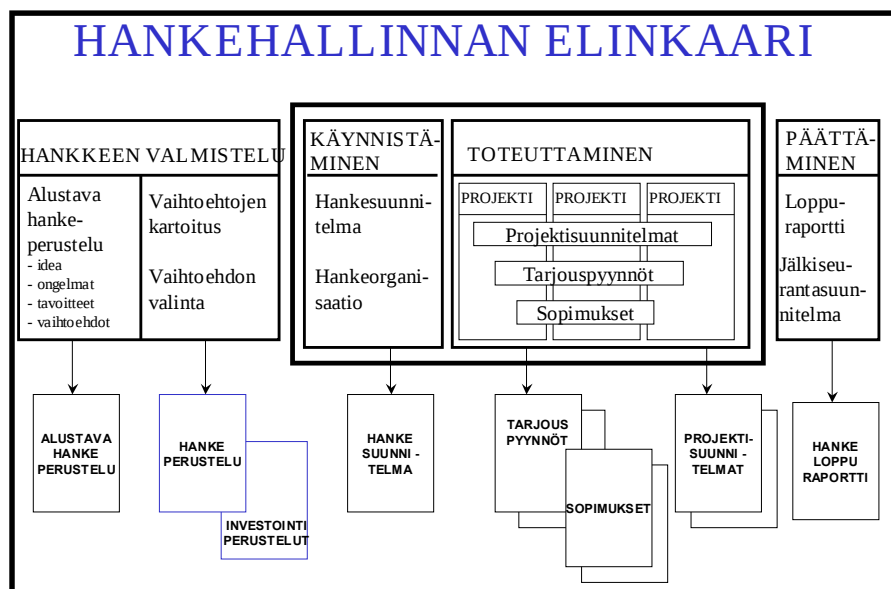
Hankkeen päättäminen

Hankkeen päättäminen on yleensä lopputulema onnistuneesta, mutta loppua kohti tiivistyneestä työrupeamasta. Hankkeen todetaan usein loppuneen, kun kaikki sen projektit ovat saavuttaneet tuloksensa. Hanke voidaan organisatorisesti purkaa kun työ on tehty ja loppuraportti kokemuksista kirjoitettu ja loppujuhla pidetty. Usein kuitenkin hankesuunnitelman mukaiset tavoitteet/hyödyt on jo perustelussa todettu toteutuvan pidemmällä aikavälillä, siksi hankkeen aktiivisen päättämisen yhdessä täytyy jälkiseuranta tältä osin aina aikatauluttaa ja vastuuttaa jollekin henkilölle.

Hankkeen elinkaaren mukaisella prosessilla ja sitä tukevalla päätöksenteolla ja tuotettavien dokumenttien muodolla saadaan eri kehittämisasiat helposti ymmärrettävään ja yhdenmu-

kaiseen muotoon. Prosessilla luodaan yhteinen kieli, jossa dokumentit toimivat loistavana viestin viejänä, kun hankkeen elinkaaren aikana mukana olevat osapuolet vaihtuvat.

Sirpa Creutz,
CIO,
VR



Systemityön historia - Sinustako kirjoittaja

Sytykesanomien vuoden 2003 ensimmäinen maaliskuun teemanumero on omistettu aiheelle

"Systemityön historia Suomessa".

Aihe kattaa systemityön trendien, menetelmien, toimintapojen jne. käytön ajan kuluessa.

Etsimme kirjoittajia, joilla on tietämystä sekä käytännön systemityöstä että teoreettisesta systemityön käsittelytavasta. Aihe on laaja ja kiintoisa ja vaatii kirjoittajilta käytännön työkokemuksen antamaa historiallista perspektiiviä suurten muutosten tai mielenkiintoisten yksityiskohtien näkemiseksi.

Jos omaat näkemystä systemityön muutoksista pitkällä aikavälillä ja koet systemityön historian Suomessa itsellesi tärkeänä omaan uraasi liittyen, niin ota yhteyttä Lauri Laitiseen sähköpostitse:

lauri.laitinen@nokia.com.

Useimmat lukijoistamme ovat käyttäneet, yhä käyttävät ja saaneet koulutusta joistakin systemityön menetelmistä ja työkaluista. Heistä on mielenkiintoista lukea, missä ajallisessa jatkumossa he itse asiassa olivat mukana rakentamassa suomalaisen systemityön historiaa.

Systemityön historiaa voidaan tarkastella kahdella tavalla. Systemityön kehitys voidaan nähdä juutalais-kristillisen aikäkäsityksen mukaisesti jatkuvana kehityksenä kohti parempaa (lineaarinen) tai erilaisina sopeutumisinä vallitseviin, muuttuviin ja osin myös toistuviin (syklisen) olosuhteisiin.

Syklisessä lähestymistavassa jotakin systemityön menetelmää, toimintatapaa, ajattelutapaa, kuvaustapaa, organisointitapaa, trendiä voidaan tarkastella esimerkiksi "nousu - kukoistus - tuho" kehyksessä. Systemityön historia on sarja perättäisiä, limittäisiä tai samanaikaisia trendejä ja kehityskaaria. Tällöin herää mm. seuraavanlaisia kysymyksiä:

- Miksi X syntyi? Mitkä tekijät tai olosuhteet synnyttivät tarpeen X:ään? Mistä X syntyi? Mitkä tekijät, ajatukset tai olosuhteet mahdollistivat X:än? Miten X syntyi? Kuinka, milloin ja kenen toimesta X syntyi ja kehittyi?
- Milloin, missä ja miten X:ää sovellettiin sen huippuaikoina?
- Miksi X:stä luovuttiin? Mikä olosuhteiden muutos (tekni- nen, bussiness, organisatorinen, jne..) tai kilpaileva Y aiheutti X:n käytön hiipumisen.
- Mitä X:stä jäi eloon? Minkä Z:n osana se jatkaa elämäänsä. Onko X yhä elossa toisella nimellä?
- Mitä X:stä opimme? Mitä jäi oppimatta?
- Mitä X:siä ei koskaan päässyt kukoistukseensa ja miksi?

Vai onko sittenkin mielestäsi systemityössä havaittavissa kehityslinjoja, trendejä, läpi vuosikymmenien?

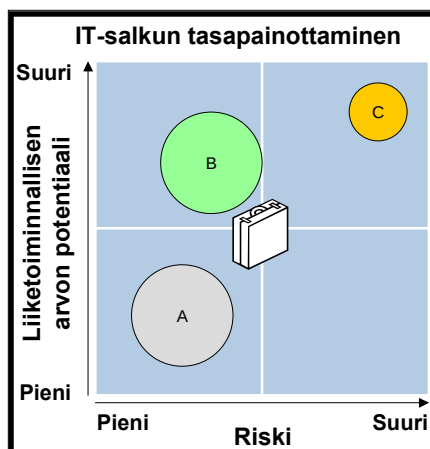
Esimerkiksi näitä ja monia muita systemityön historiaan liittyviä aiheita voit käsitellä kirjoituksessasi.

IT Portfolio Management johtamisen välineenä

Aarni Heiskanen,
AE Thinking Business Group

Vaikka suuri osa yritysten IT-investoinneista on välttämättömiä, niiden kytkeminen yrityksen liiketoiminnan strategiaan tavoitteisiin ei ole itsestään selvyyttä. Meta Groupin kyselytutkimuksen¹ mukaan

- 84 % yrityksistä ei tee liiketoiminnallista tarkastelua IT-projekteistaan tai tekee sen vain valituista avainprojekteista
- 83 % organisaatioista ei pysty tarkistamaan tai uudelleen suunnittelemaan budjettejaan kuin kerran tai kaksi vuodessa
- 67 % IT-organisaatioista eivät ole ”markkinavalmiita” - benchmarkingia tehdään harvemmin kuin vuosittain
- 89 % organisaatioista ”lentää pimeässä”, käyttäen ainoastaan talousmittareita
- 57 % yrityksistä tasapainoilee mielestään vastakkaisten paineiden - kustannusleikkausten ja IT:n suoritusvaatimusten – välillä, vaikka tämä havainto ei olisi sikaan totta.



Kuva 1: Salkun tasapainottaminen

Nopeiden muutosten maailmassa kaivataan myös IT-toimintojen ketteryyttä. IT-salkunhallinta (IT Portfolio Management) on menetelmäkokonaisuus, joka antaa sekä tietohallinnolle että liiketoimintajohdolle työkalut ajantasaiseen IT-omaisuuden kehittämiseen ja hallintaan.

Periaate

IT Portfolio Management on menetelmällinen, jäsennelty, helposti ylläpidettävistä prosesseista koostuva ratkaisumalli, jolla yhdistetään liiketoiminnan vaatimukset IT-päätösten tekoon. Ratkaisu mahdollistaa IT-omaisuuden ja -prosessien

- luokittelun
- arvioinnin
- priorisoinnin
- hankinnan ja
- hallinnan

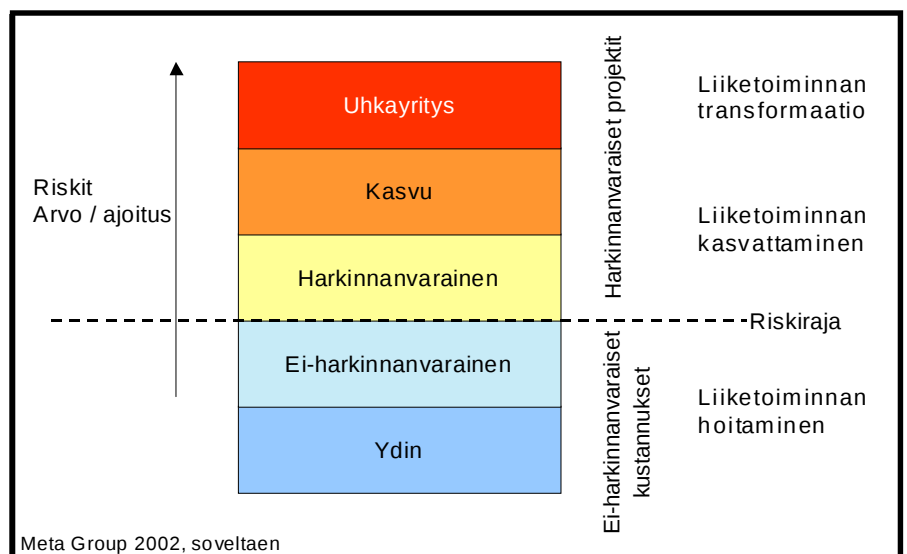
siten, että palvellaan liiketoiminnan nykyisiä ja tulevat tarpeita, määritellyllä tulos- ja riskitasolla. Salkun sisältöä käsitellään osakesalkutuksesta tutuilla

”hanki, pidä, luovu”-operaatioilla sekä tasapainotetaan liiketoiminnallisen arvon ja riskien perusteella (kuva 1).

Mitä IT-salkut sisältävät?

IT-salkutus alkaa usein olemassa olevien projektien luokittelulla, koska IT-toimintoja kehitetään useimmiten juuri projektien kautta. Projektille määritellään liiketoiminnallinen arvo ja mittarit, joilla arvoa kuvataan (kuva 2). Projekti voi kohdistua liiketoiminnan kannalta ydinjärjestelmiin (ei-harkinnanvarainen), sillä voidaan tavoitella liiketoiminnallista kasvua (harkinnanvarainen) tai sillä voidaan pyrkiä suoranaiseen liiketoiminnan transformointiin (”uhkayritys”).

Salkunhallintaan kuuluu oleellisesti riskiluokitus. Riskejä tarkastellaan esim. niiden vaikuttavuuden, vaikutuksen kohteeseen ja todennäköisyyden perusteella. Riski ei liity pelkästään projektin toteuttamiseen: Joskus suurin riski syntyy, jos projektia ei toteuteta.



Meta Group 2002, soveltaen

Kuva 2: IT-hankkeiden liiketoiminnallinen luokittelu

1 Tunnistaminen	Tieto kaikista hankkeista ja IT-järjestelmistä kootaan yhteen paikkaan, vertailukelpoiseen muotoon.
2 Salkutus	Olemassa olevat järjestelmät, projektit, ihmiset ja prosessit, jäsennetään ja priorisoidaan salkkuihin. Salkutusta hyödynnetään kommunikointivälineenä.
3 Salkunhallinta	IT-salkutus on korkean tason hallinta- ja päätöksentekoväline. Käytettyjä tekniikoita ovat mm. riskien ja tuoton tasapainotus, tuottojen arviointi ja syklien tai muutoskohtien tunnistaminen.
4 Edistyksellinen salkunhallinta	Portfolio Management-tekniikoita käytetään yksityiskohtaisesti kaikessa IT-päätöksenteossa ja prosessien sekä organisaation hallinnassa. Salkunhallintaan käytetään automatisoitua järjestelmää.
5 Maailmanluokka	Yksityiskohtainen Portfolio Management on koko liiketoiminnan toimintamalli ja IT on yksi salkkukokonaisuus muiden joukossa.

Kuva 3 IT Portfolio Management-kypsyystasot

Muita IT-salkuissa hallittavia kokonaisuuksia ovat laitteet, ohjelmistot, osaaminen, henkilöresurssit ja prosessit. Kullekin salkutuksen kohteelle on

määriteltävissä yrityksen kannalta oleelliset luokitustekijät. Riski-tuotto-luokituksen lisäksi muita tasapainotettavia tekijöitä ovat esim.

- ajanjakso: lyhyt – pitkä
- koko: pieni – suuri
- kattavuus: paikallinen – koko organisaation laajuinen
- investoinnin koko: pieni - mittava

Luokitus toimii hyvänä lähtökohana mm. vaaditulle päätöksenteon tasolle ja ajankohdalle sekä päätöksentekoprosessille.

Salkunhallinnan implementointi

IT-resurssien ja prosessien salkuttaminen on vielä suhteellisen uusi ajattelutapa. Kuvassa 3 on esitetty implementoinnin viisi tasoa². Tarkoituksenmukainen taso riippuu johdon sitoutumisesta ja johtamistavasta. IT Portfolio Management on tapauksessa kaivattu konkreettinen menetelmä liiketoiminnan dynamiikan toteuttamiseen IT-johtamisessa. Edelläkävijät ovat raportoineet merkittävistä arvonnäkökohdista ja samanaikaisesta IT-menojen pienentymisestä, mikä kiinnostanee kaikkia IT-päätäjiä.

*Aarni Heiskanen,
Senior Partner, LJK,
AE Thinking Business Group,
aarni.heiskanen@thinking-
business.net,
p. 040-730 4812*

Lähteet:

1) The Business of IT Portfolio Management: Balancing Risk, Innovation, and ROI, Meta Group White Paper January 2002

2) Do the Math, CIO Magazine October 1, 2001

Projektisalkun hallinta

Päivi Hokkanen,
Stockmann Oyj Abp

Tutkimusten mukaan edelleenkin 1/3 kaikista IT-projekteista epäonnistuu asiakkaan tavoitteiden perusteella mitattuna. Mitä isompi ja pitkäkestoisempi projekti, sen varmemmin se menee pieleen.

Hankkeiden onnistumisen varmistaminen onkin eräs tärkeimmistä ja samalla hankalimmista IT-johdon haasteista.

Projektitasolla työkaluina voidaan käyttää paitsi kunnollista projektihallintatyökalua, myös kunnollista esityötä. Esitutkimuksella ja vaatimusmäärittelyllä on selvä yhteys hankkeen

onnistumiseen. Mitä paremmin sekä oma organisaatio että toimittajan organisaatio ymmärtävät mitä on tarkoitus tehdä, sitä paremmat mahdollisuudet hankkeella on onnistua. Lisäksi kannattaa varmistaa, että osapuolet ymmärtävät asian samalla tavalla, jotta projekti saadaan liikkeelle yhteisistä lähtökohdista käsin, eikä niin, että eri osapuolilla on eri agenda.

IT-johdon kannalta projektien tarkastelu projektin hallintatyökalujen avulla on usein liian työlästä, koska yhtäaikaista projekteja on liikkeellä vähääkään isommassa yrityksessä kymmeniä tai jopa satoja. IT-johdolla ei ole useinkaan minkäänlaisia mahdollisuuksia tarkastella kutakin käynnissä olevaa projektia osallistumalla

sen ohjausryhmään, koska niitä on yksinkertaisesti liian monta. Tämän vuoksi uskonkin, että projektien seuranta tulisi isoissa ympäristöissä tehdä salkkutasolla, jotta kokonaisuus on mahdollista pitää hallinnassa.

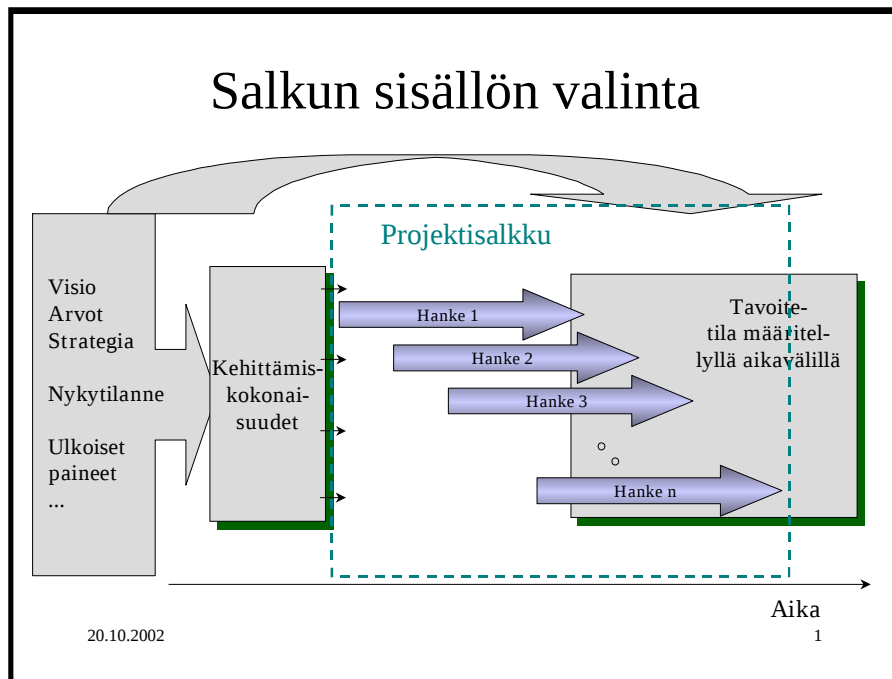
Projektisalkun sisältö

Kun yritys laatii tai tarkistaa strategiaansa seuraaville vuosille, siihen vaikuttavat yhtäläillä yrityksen arvot ja visio, nykytilanne kuin ulkoiset paineetkin. Riippumatta toimialasta yhä useampi strateginen hanke, jolla pyritään tavoiteltuun tilaan määritellyllä aikavälillä tulevaisuudessa, vaatii nykyisin myös IT-projektin toteuttamista, jotta tavoitetilaan päästään.

Näin yrityksen liiketoimintastrategia määrittelee pitkälti myös yrityksen projektisalkun sisällön. Kun liiketoimintastrategia on tehty, laaditaan/tarkistetaan IT-strategia, jonka merkittävin tehtävä on tukea ja toteuttaa hyväksyttyä liiketoimintastrategiaa.

Budjetoinnin yhteydessä investointibudjettiin otetaan mukaan yrityksen liiketoimintastrategioiden mukaiset hankkeet, jotka täydentävät jo olemassa olevaa projektisalkkua.

Käytössä olevia budjetointimalleja on useita, mutta investointeja ajatellen tehokkaimmillaan se käsitykseni mukaan on silloin, kun IT-investointien maksimi määrä on asetettu



Projektisalkun seuranta

Projektin nimi	Omistaja	Status	Valmispvm	Budj. eur	Toteur	Vaikeusaste	Hyöty	Riskit	Prioriteetti
Alkaneet projektit									
1									
2									
3									
4									
...									
Hyväksytty alkamaan									
1									
2									
3									
4									
...									
Listalla olevat									
1									
2									
3									
4									
...									
20.10.2002									1

set viedään salkkuun, jotta mikään ehdotettu case ei häviä, vaan niistä jää jälki. Toteutettavaksi hyväksytyt BusinessCaset priorisoidaan johtoryhmässä niiden tuottaman liiketoimintahyödyn, taloudellisen hyödyn, teknisen sopivuuden ja resurssoinnin kannalta katsottuna.

Projektisalkku voi olla yksinkertaisimmillaan vaikkapa vain alla olevan mallin kaltainen Excel-taulukko liitettynä Intranetiin siten, että kaikki BusinessCaset ja niitä koskeva muu dokumentaatio voidaan sitä kautta aina helposti löytää.

Projektisalkkua seurataan johtoryhmässä ja se on parhaimmillaan hyvä työkalu salkun riskin ja mahdollisten ongelmien seuraamiseen. Salkkuun sisältyvät riskit syntyvät tietysti suoraan sen sisältämistä projekteista, ts. toteutuvatko ne suunnitelman mukaisesti (aikataulu, resurssointi, budjetti) ja saavutetaanko tavoiteltu hyöty.

Salkun seuranta luo siis mahdollisuuden valvoa projekteja yhtenä kokonaisuutena ja tavan hallita hankkeisiin sisältyvää riskiä. Se myös varmistaa kehitysinvestointien kohdistamisen ja seuraamisen hallitulla tavalla.

*Päivi Hokkanen,
CIO,
Stockmann Oyj Abp*

vuositasolle (hankkeiden kautta suunniteltuna) budjettiin, mutta investointeja ei tämän perusteella automaattisesti hyväksytty toteutettaviksi laadittujen budjettien mukaisesti. En nimittäin usko, että IT-investointien suunnittelu muuten kuin karkealla tasolla onnistuu kerran vuodessa tehtynä koko seuraavaksi vuodeksi. Tästä syystä mielekäs käytäntö onkin käyttää projektisalkkua apuvälineenä investointien suunnittelu- ja seurantaprosessissa.

Investointien suunnittelu- ja seurantaprosessi

Kaikista merkittävistä hankkeista (tietyn €-määrän ylittävät hankkeet, raja asetetaan yrityskohtaisesti) laaditaan ns. Business Case, jossa kuvataan hankkeen lähtökohdat, tarve, saa-

vutettavat hyödyt, sen riskit sekä alustava investointikustannus. Tämä BusinessCase laaditaan liiketoimintayksikössä ja IT-yksikkö voi auttaa mm. investointikustannuksen laskennassa, jos tarpeen.

BusinessCaset käsitellään 1x/kk (tai harvemmin, taas riippuen organisaation koosta) investointien johtoryhmässä, joka hallinnoi IT-budjetin mukaista kokonaisinvestointipottia. Investointien johtoryhmään kuuluu ylimmän johdon edustajat ja tietohallintojohtaja, jotta tällä elimellä on riittävä valtuutus IT-investointien suunnittelussa.

BusinessCase voidaan hyväksyä toteutettavaksi, jättää se hetkeksi odotamaan tai hylätä kokonaan investointien johtoryhmän päätöksellä. Riippumatta päätöksestä kaikki BusinessCa-

Prosessien kehittämisen viitekehys

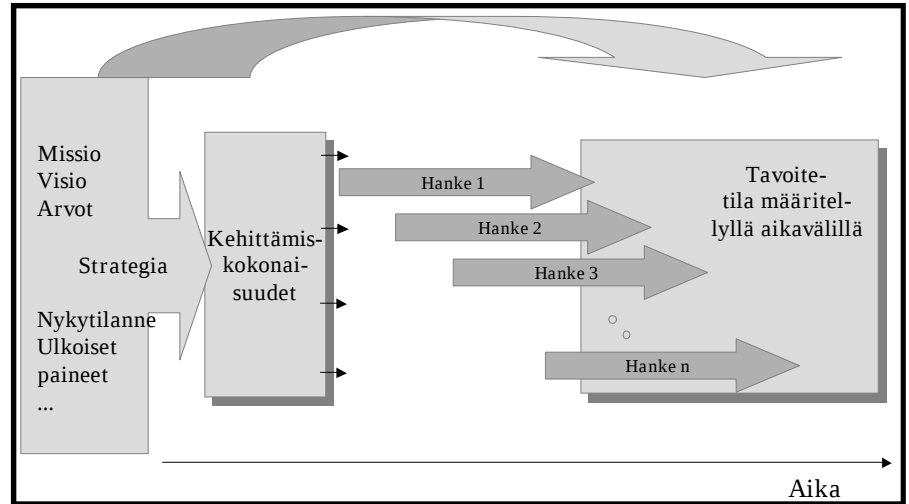
Pekka Särkkinen,
EDS

Toimintaprosessien ja tietojärjestelmien kehittäminen ilman strategista suuntaa ja selkeää tavoiteasetantaa tuottaa harvoin koko organisaation toiminnan kannalta parhaita ratkaisuja. Toisaalta pelkästään tehokkuutta ja suorituskykyä parantavat tietojärjestelmähankkeet – ilman toimintatapojen muutosta – tuottavat harvoin merkittävää etua. Parhaiten onnistuneissa toiminnan kehittämishankkeissa on pystytty yhdistämään strategiaan perustuva selkeä tavoitetilä prosessien suunnittelun kautta uudenlaisen toiminnan mahdollistaviin tietojärjestelmäratkaisuihin. Joissain tapauksissa tekniikan ja järjestelmien avulla on uudistettu radikaalisti nykyisiä toimintatapoja. Toiminnan tasapainoisen kehittämisen kannalta oleellista on myös asiakas-kohtaamisten, kanavien, henkilöiden ja kumppanuuksien kehittäminen sekä toiminnan johtaminen. Hyvin tehty toimintaprosessien kehittäminen ottaakin huomioon koko ”prosessiympäristön”.

Strategia ohjaa kehittämistä

Strategia vastaa yleensä kysymyksiin mitä, kenelle ja miten. Strategian vieminen käytäntöön vaatii edellä mainittuihin kysymyksiin vastaamisen lisäksi konkreettisen kehityssuunnitelman ja mittariston. Strategia jää vaja-vaiseksi, ellei siihen yhdistetä strategisten tavoitteiden saavuttamiseksi käynnistettäviä hankkeita sekä tavoitteiden ja muutoksen saavuttamisen mittaamista.

Strategian oleellinen muutos johtaa yleensä aikaisemmasta poikkeavan mittariston kehittämiseen. Mittariston, jolla pystytään mittaamaan muutoksen toteutumista. Mittaaminen toteutetaan nykyään yleisesti tasapainoisen mittariston kautta.



Kuva 1: Kehittämisen viitekehys.

Tasapainoisen mittariston viitekehys muodostuu neljästä osa-alueesta – taloudellisista mittareista, asiakkuusmittareista, prosessimittareista ja henkilöstön kehittymiseen liittyvistä mittareista. Organisaatiot käyttävät tätä viitekehystä omien tarpeidensa mukaisesti ja joissain tapauksissa viitekehukseen on lisätty myös viides osa-alue tai mukaan on otettu vain kolme aluetta. Vaikka tässä yhteydessä ei paneuduta lähemmin tasapainoisen mittariston kehittämiseen, pitää jokaisessa organisaatiossa strategian ja sen seuraamiseen luodun mittariston asettaa selkeä tavoite toiminnan kehittämiseksi, prosesseille ja toiminnan mahdollistaville tietojärjestelmille. Kehittämishankkeilla pitää olla konkreettiset tavoitteet, joiden saavuttamista pystytään hankkeen aikana seuraamaan, välittämään tavoitteet hankkeen osa-alueille ja projekteille sekä myös tietojärjestelmille. Kehittämishankkeen toteuttamisen jälkeen tulee pystyä seuraamaan strategian toteutumista ja tavoitteiden saavuttamista uuden mallin mukaisesti toimittaessa.

Strategiaa muodostettaessa toiminta joudutaan hallittavuuden saavuttamiseksi jakamaan sekä suunnittelun että toteuttamisen osalta osakokonaisuuksiin. Yksinkertaistettuna kehittämisen voidaan kuvata tapahtuvaksi kuvan

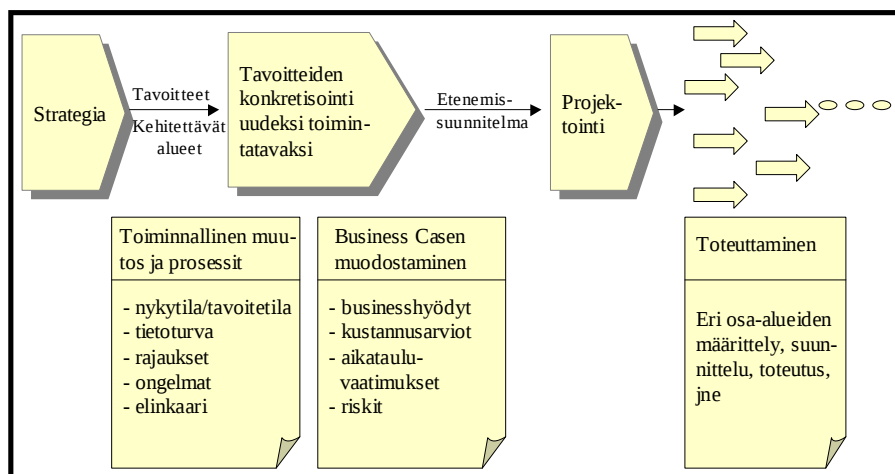
1 esittämällä tavalla, jossa hankkeet muodostavat strategiaan liittyvän aikataulutetun kehittämissuunnitelman tavoitteiden saavuttamiseksi.

Hankkeesta projekteiksi

Kehittämishankkeen konkretisointi toteutusprojekteiksi vaatii tulevan toimintatavan tarkemman suunnittelun. Tämä vaihe tunnetaan prosessien mallintamisena, joka tehdään ennen tarkempaa määrittelyä. Tässä yhteydessä määrittely -käsitettä ei rajata tietojärjestelmän määrittelyyn, joka ottaa kantaa vain järjestelmiin liittyviin toiminnallisiin asioihin. Uuden toimintatavan toteuttaminen vaatii monien tietojärjestelmien ulkopuolella olevien asioiden määrittelyä ja toteuttamista.

Strategian tekovaiheessa hahmotellaan usein prosessivisioita, mutta tarkempi prosessin ja sen ympäristön suunnittelu toteutetaan yleensä omana kokonaisuutenaan. Seuraava kuva 2 luonnehtii kehittämisen etenemistä yhden hankkeen osalta ja liiketoiminnallisten vaatimusten ja prosessien mallintamisen sijoittumista kehittämisessä.

Kehittämisen onnistumisen kannalta oleellista on kaikkien toiminnan tavoitetilan saavuttamiseen liittyvien



Kuva 2: Tavoitteiden konkretisointi projekteiksi.

asioiden huomioiminen. Tavoitteena voi olla asiakaspalvelun kehittäminen, asiakkaan toiminnan helpottaminen uuden kanavan kautta, kumppanien suuremman roolin aikaansaaminen toiminnassa tai toiminnan jonkin osan suorituskyvyn kehittäminen. Mihin tahansa osa-alueeseen strategiset tavoitteet liittyvät, joudutaan konkreettinen eteneminen toteuttamaan toimintaprosesseja kehittämällä. Asiakaspalvelun kehittäminen vaatii kaikkien niiden prosessien tarkastelun, joissa asiakas kohdataan. Uuden kanavan kehittäminen vaatii tarkastelemaan prosessien suoritustapaa kyseisessä kanavassa ja vaikutusta samoihin prosesseihin muissa kanavissa. Kumppanuuskehittäminen vaatii niiden prosessien tarkastelemista, joissa kumppanit ovat mukana. Ulkoistamista toteutettaessa halutaan usein nimetä ulkoistamista prosessi tai sen osa. Mikäli ennen ulkoistamishanketta ei ole itselle pystytty prosessimuodossa selvittämään ulkoistettavaa kokonaisuutta, on hyvin vaikeaa sopia ulkoistamisen tapahtuessa menettelytavoista rajapinnoissa ja mitata ulkoistamiselle asetettujen liiketoiminnallisten tavoitteiden saavuttamista.

Prosessit yhteinen kieli eri toimintojen kesken

Toimintaa konkreettisesti kuvaavana prosessit muodostavat yhteisen ymmärrettävän kielen erilaisen taustan ja näkökulman omaavien kehittäjien kesken. Prosessit muodostavat myös konkreettisen kuvaustavan, jossa strategiset tavoitteet voidaan toiminnallisella

tasolla suunnitella käytännön toimenpiteiksi ja työskentelytavoiksi. Strategisten tavoitteiden saavuttaminen ei kuitenkaan ratkea pelkän prosessivirran kuvaamisella ja sen uudelleen suunnittelulla. Vasta prosessiin liittyvän koko prosessiympäristön, kuva 3, tasapainoinen kehittäminen mahdollistaa toiminnallisten tavoitteiden saavuttamisen.

Tavoitteiden konkretisointi toteutuskelpoiseksi suunnitelmaksi vaatii niiden toiminnallisten kokonaisuuksien läpikäymistä, jotka vaikuttavat tavoitteen saavuttamiseen. Nämä toiminnalliset kokonaisuudet rajaavat ne prosessit tai prosessin osat, jotka ovat uudelleen suunnittelun kohteena. Toiminnan kehittämisvaiheessa prosessiympäristön analysointi osa-aluekohtaisesti helpottaa kokonaisuuden hallitsemista ja auttaa sen jäsentämisessä. Seuraavassa on lyhyesti kuvattu prosessiympäristöön kuuluvat osa-alueet.

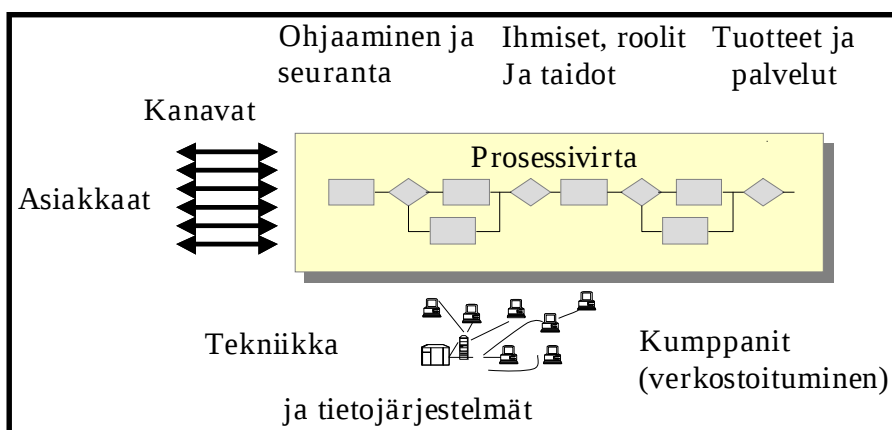
Asiakkaan tarpeiden parempi tyydyttäminen on usein tärkeä toiminnan kehittämisen tavoite. Asiakashan on

useimmiten suorittamassa jotain omaa prosessiaan, johon yrityksen palvelu tai tuote pitää saada optimaalisesti liittyvän. Prosessien kehittämisessä asiakaslähtöisyys ja asiakkaan tarpeiden tunteminen onkin aivan oleellisessa asemassa. Usein tässä yhteydessä asiakkaan prosessiin paneutuminen auttaa ja helpottaa myös oman toiminnan kehittämisessä.

Asiakkaan tarpeet tyydytetään - asiakas kohdataan - jonkun tai joidenkin kanavien kautta. Kanava voi olla esimerkiksi palvelupiste, puhelin, sähköposti, kumppani, itsepalvelupiste tai kirje. Uusien kanavien kehittäminen onkin viime vuosina ollut oleellinen kehittäminen kohde. Esimerkkinä tästä on voimakas itsepalvelun ja sähköisen kanavan kehittäminen tai kaupan piirissä tapahtunut erityyppisten palvelupisteketjujen muodostaminen erilaisine ulkoasuineen ja valikoimineen.

Prosessivirta kuvaa mitä toiminnallisia kokonaisuuksia asiakkaan palvelemiseksi tarvitaan. Tämä tekemisen tapa voi toiminnan tehokkuuden, miellyttävyyden, asiakasystävällisyyden, kustannusrakenteen ja monen muun asia kannalta olla oleellinen tavoitteiden saavuttamisen väline. Prosessivirran kuvaus on myös tärkeä väline uuden toimintatavan suunnittelussa, toiminnan maastouttamisessa ja se helpottaa huomattavasti työhön perehdyttämistä erityisesti sellaisissa organisaatioissa, joissa henkilöiden vaihtuvuus on suuri.

Johtaminen, ohjaaminen ja mittaminen muodostavat oleellisen osan toiminnan hallitsemista. Toimintaa ja pro-



Kuva 3: Prosessiympäristö

sesseja kehitettäessä tulee paneutua siihen, minkälaisia mittareita toiminnan seuraamiseen luodaan ja miten mittarit prosessien yhteyteen toteutetaan. Prosesseja suunniteltaessa tulee vastata kysymykseen miten prosessi saadaan kertomaan riittävästi itsestään. Kun tämä suunnitellaan riittävän aikaisessa vaiheessa, pystytään mittareiden arvojen kerääminen ja erilaiset raportit suunnittelemaan ja toteuttamaan muun toiteutuksen yhteydessä.

Henkilöiden roolit ja kussakin roolissa tarvittavat tiedot ja taidot nousevat usein tärkeään asemaan uuden toimintatavan käytäntöön saattamisessa. Toimintatapojen kehittäminen saattaa vaatia paljonkin henkilöiden rooleihin, osaamiseen, ajattelutapaan ja asenteisiin liittyvää kehittämistä. Tämä on saatava tapahtumaan oikeassa aikataulussa muun kehittämisen kanssa, sillä toiminnallisia hyötyjä voi olla hyvin vaikea muuten saavuttaa. Usein henkilötason muutokset ovat sekä kalenteriajallisesti että työmäärältään mittava kokonaisuus. Henkilömäärät ovat suuria ja koulutus, valmennus ja kehittämisohjelmat joudutaan toteuttamaan normaalin toiminnan ohessa. Kehittämisen vaiheistamisessa ja priorisoinnissa oman organisaation ja henkilöiden asettamat rajoitukset tulee ottaa riittävän realistisesti huomioon. Joitain osa-alueita esimerkiksi järjestelmien osalta voidaan usein viedä hyvinkin nopeasti eteenpäin, mutta jos henkilöiden toimintaa ei pystytä samassa aikataulussa kehittämään, niin tavoitellut hyödyt jäävät usein saavuttamatta ja uusi järjestelmä saatetaan kokea tämän takia epäonnistuneeksi.

Kumppanien osuus toiminnassa ja sen kehittämisessä on viime vuosina korostunut voimakkaasti. Kyse ei ole pelkästään siitä, että kumppanit tekevät osan tehtävistä tai osan prosessista. Kyse voi olla mittavasta prosessien ulkoistamisesta. Tätä tapahtuu nykyään kaikkien prosessien osalta. Esimerkkeinä mainittakoon markkinointi, myynti, valmistus, toimitus, talous, henkilöstö ja tietohallinto. Monissa tapauksissa tässä yhteydessä puhutaan myös verkos-

toitumisesta ja jotkut yritykset perustavatkin koko toimintansa voimakkaasti kumppaneiden varaan.

Uusien tuotteiden ja palveluiden kehittäminen voi strategisten tavoitteiden saavuttamiseksi olla hyvin oleellista. Esimerkkinä tästä on uusien tuotteiden ja uusien palveluiden luominen sähköiseen kanavaan. Osa toiminnallisista kehittämishankkeista taas tapahtuu olemassa olevan tuote- ja palvelukokonaisuuden perusteella ilman, että siihen liitetään tältä osin mitään oleellista kehitystä. Kulloisestakin tilanteesta riippuen tuote- ja palvelukehityksen huomiointi prosessien kehittämisen yhteydessä luonnollisesti vaihtelee.

Tekniikan hyödyntäminen toiminnan kehittämisessä ja kokonaan uusien toimintatapojen mahdollistajana on noussut viime vuosikymmeninä lisääntyvässä määrin uusien toimintatapojen mahdollistajaksi. Erityisesti tietotekniikka on mahdollistanut sellaisia toimintatapoja ja ratkaisumalleja, joita aikaisemmin ei edes pystytty kuvittelemaan. Huolimatta tästä tekniikan voimakkaasta kehitymisestä, tekniikan käyttöönotto tai järjestelmähanke ei yleensä ratkaise toiminnan kehittämistarvetta kokonaisuutena. Järjestelmäprojekti ei vie eteenpäin koko toiminnallista kokonaisuutta. Prosessiympäristö pitää huomioida kokonaisuutena.

Kehittämisen vaiheistus

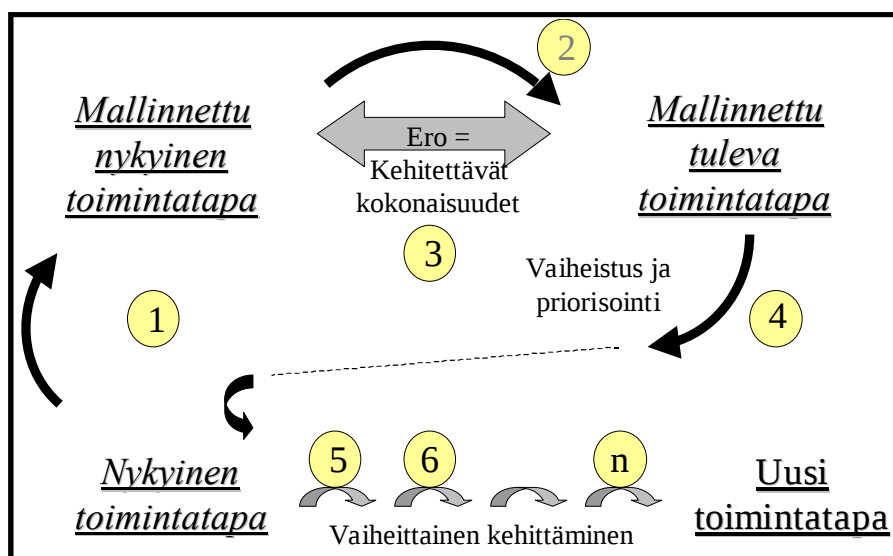
Toiminnan kehittämisessä ja prosessiympäristön läpikäynnissä kannattaa noudattaa jotain kehittämisen perusvaiheistusta, joka voidaan pelkistää kuvan 4 muotoon.

Suunnittelu jaetaan neljään vaiheeseen.

Vaiheessa 1 otetaan haltuun nykyinen toimintatapa. Tähän löytyy organisaatiossa yleensä runsaasti tietämystä, dokumentaatiota, erilaisia menettelytapoja ja muodostuneita käytäntöjä. Oleellista tässä vaiheessa on muodostaa yleisellä tasolla yhteinen näkemys nykytilanteesta ja kirjata siitä oleelliset asiat kuvaukseksi.

Vaiheessa 2 mallinnetaan tuleva toimintatapa. jolloin on hyödyllistä irtautua hetkeksi rajoittavista realiteeteista ja pyrkiä löytämään toimintaan uudenlaisia ratkaisuja. Vaihtoehtojen arvioinnissa ja erilaisten laskelmien tekemisessä realiteetit aikanaan asettuvat varmasti paikalleen.

Vaiheen 3 muodostaa tulevan toimintatavan ja nykyisen toimintatavan erojen analysointi, jolloin pystytään muodostamaan konkreettinen käsitys kehitettävistä kokonaisuuksista.



Kuva 4: Kehittämisen perusvaiheistus

Vaiheessa 4 kehitettävä kokonaisuus priorisoidaan ja vaiheistetaan sekä muodostetaan tarvittavat projektit toteutusta ja seuranta varten. Kuten aikaisemmin jo todettiin, tässä vaiheessa tulee pyrkiä mahdollisimman suureen realismiin oman organisaation suorituskyvyn osalta.

Seuraavat vaiheet, 5-n, ovat suunnitelman toteuttamista, hankkeiden ja niihin kuuluvien projektien läpivientiä.

Prosessiympäristön suunnittelun käytännön työskentelyä voidaan kuvata oheisen kuvan 5 matriisin muodossa, jossa prosessiympäristön osa-alueet ja kehittämisen perusvaiheistus on yhdistetty taulukkomuotoon. Vasemmalla allekkain ovat prosessiympäristön osa-alueet muodostaen jaoteltun kehittämisen läpikäytävälle asiakokonaisuuksille. Ylhäällä kulkee kehittämisen vaiheistus vasemmalta oikealle. Tällä tavalla ositettuna kehitettävät kokonaisuudet saadaan jaoteltua ja prosessiympäristön nykytilanne, tavoitetilanne ja kehittämistarve jäsennettyä vaiheistusta ja priorisointia varten.

Matriisi muodostaa mallin prosessien kehittämisen vaiheistukselle. Prosessin nykytilanne saadaan selville ”täyttämällä” osa-aluekohtaisesti matriisin solut. Vastaavasti tehdään prosessin tavoitetilanteen osalta. Tämän jälkeen pystytään osa-aluekohtaisesti kuvaamaan kehittämistarpeet.

Kokoamalla osa-alueittaiset kehittämistarpeet loogisesti ja ajallisesti toteutettaviksi kokonaisuuksiksi saadaan etenemisen kannalta oleelliset projektit ja vaiheet kuvattua kehittämiskartaksi, kuva 6.

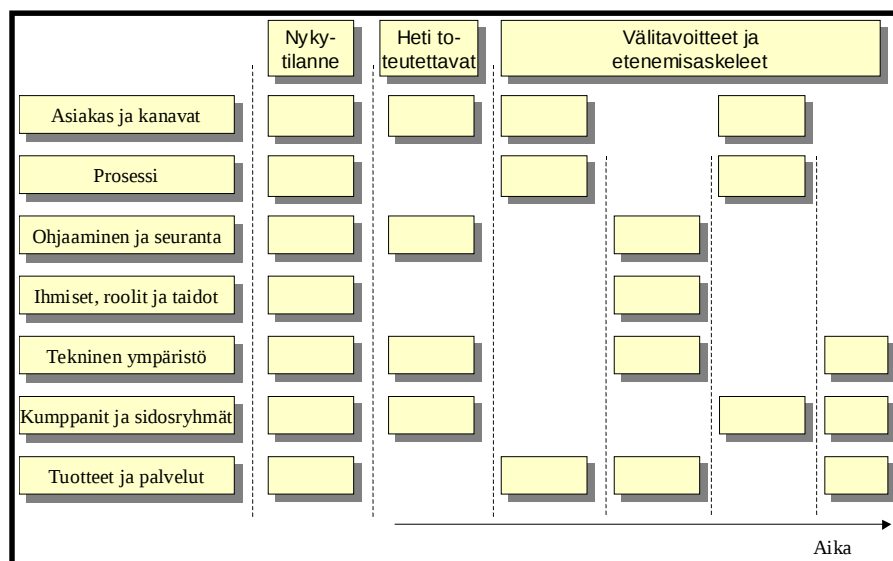
Vaiheistus tulee suunnitella siten, että niistä kukin tuottaa lopputuloksena käytäntöön viedyn toiminnallisen kokonaisuuden, joka omalta osaltaan vie toimintaa kohti asetettuja tavoitteita. Vaiheet tulee suunnitella riittävän lyhyiksi, mielellään noin 4-5 kuukauden mittaisiksi, jolloin saavutetaan nopeasti konkreettisia lopputuloksia ja kehittämisen hallinta helpottuu.

	Prosessin nykytilanne	Prosessin tavoitetilanne	Kehittämistarve	Vaiheistus ja priorisointi	Kehittäminen
Asiakas ja kanavat					
Prosessi					
Ohjaaminen ja seuranta					
Ihmiset, roolit ja taidot					
Tekninen ympäristö					
Kumppanit ja sidosryhmät					
Tuotteet ja palvelut					

Kuva 5: Prosessiympäristön ja kehittämisen vaiheistuksen yhdistäminen osa-alueittaiseksi etenemiseksi.

Eräs kehittämiskartan suurista hyödyistä on, että se havainnollistaa osakokonaisuuksien suhteen toisiinsa sekä ajan että riippuvuuden suhteen. Samalla se konkretisoi sitä, että pelkästään yhden osa-alueen ratkaisulla ei yleensä saavuteta toiminnan kehittämisen tavoitteita. Tarkempi suunnittelu tehdään hankesuunnitelman ja projektisuunnitelmien avulla.

Pekka Särkkinen,
EDS



Kuva 6: Kehittämiskartta.

Surkeuden loppu

Pekka Forselius,
Sttf Oy

“Voi luoja mikä kasa paperia Nakkilan Puljusta on tullut! Näitäkö ne kutsuvat vaatimusmäärittelyksi? Pitääkö näistä tosiaan yrittää tehdä tarjous? Voi ei, ja vielä ylihuomiseen mennessä! Ei tule mitään! Ai niinkö? No, onhan sekin tietysti hyvä näkökulma. No, okei pomo, minä teen minkä voin.” Tämä oli ohjelmistotoimittajan tuotepäällikön osuus ohimennen kuullusta puhelin-keskustelusta.

Kuukautta myöhemmin projektihuoneessa:

“Kylläpä osaavatkin olla surkeita nämä määritykset! Ei näistä hullukaan saisi mitään järkevää aikaan! Taas on kyllä menty tekemään tosi tyhmä tarjous. Tästä ei hyvää seuraa, sanokaa minun sanoneen! Onneksi ei sentään ole mitään laatuvaatimuksiakaan tälle niin sanotulle järjestelmälle. Pakko kai tässä on kuitenkin tehdä jotakin.”

Myöhemmin sitten saamme lukea Helsingin Sanomien kuukausiliitteestä artikkelin “Saatanan tunareista”. Siinä kerrotaan, kuinka Nakkilan Puljun toimitusjohtaja huutaa naama punaisena projektikokouksessa ja ihmettelee, kun ei mitään tule valmiiksi. Toimitus on kuukausikaupalla myöhässä, ja sekin vähä mitä tulevasta ohjelmistosta on nähty, vilisee virheitä ja ei toivottuja “ominaisuuksia”.

Tuo yllä kuultu ja luettu voi monenkin mielestä kuulostaa tutulta. Toivottavasti ei enää muutaman vuoden kuluttua. Ainakaan Suomessa. Eväät parempaan ovat nimittäin nyt olemassa. Huhtikuussa 2002 tuli painosta pieni sinikantinen kirja, jossa on yhteinen toimintamalli järjestelmän tilaajille ja toimittajille. Sen nimi on “TIETOJÄR-

JESTELMÄN HANKINTA - Ohjelmistotoimittajan ja -ratkaisun valinta”.

Kirjaprojekti

Alkuperäinen aloite ohjelmistohankintamenettelyjen ja toimintamallin kehittämiseen tuli TTL:lle asiakkaalta, joka oli vuosia tuskailut erilaisen laajojen tietojärjestelmähankkeidensa kanssa. Jotenkin hankkeet eivät tuntuneet olevan hallinnassa missään vaiheessa, vaikka ne joskus olisivat sattumalta onnistuneetkin. Sekä ohjelmistoprojektien onnistumistutkimukset että kokeneiden konkarien omat havainnot vahvistavat näiden asioiden olevan todellakin melko huonossa hoidossa valitettavan usein.

Pelkkä valittelu ei kuitenkaan auta. Jonkun on todella tartuttava toimeen, jos asioihin halutaan parannusta. Niinpä tämä kirjaprojekti käynnistyi Tietotekniikan liiton kokoonkutsusta keväällä 2001. Mukaan ilmoittautui koko joukko aihepiiriin hyvin tuntevia ja siitä huolestuneita liiton jäseniä, jotka vapaaehtoisesti, ilman palkkaa osallistuivat järjestettyjen ideointityöpajojen suunnitteluun ja toteutukseen. 12-henkisen ydinryhmän lisäksi työpajoihin ja tulosten kommentointiin osallistui noin 40 käytännön työkentän edustajaa. Voidaan sanoa, että ryhmien kokoonpanot olivat hyvin tasapainossa. Sekä toimittaja- että tilaajaorganisaatiot olivat hyvin edustettuina ja lisäksi mukana oli osaava joukko nimenomaan hankintakäytäntöihin liittyvien asioiden kouluttajia ja konsultteja.

Projektin alkuvaiheissa järjestettiin neljä työpajaa, joissa esiteltiin muualla jo aiemmin kehitettyjä huippukäytäntöjä. Alustajia näissä tilaisuuksissa oli sekä Suomesta että muualta Euroopasta. Alustusten pohjalta asioita sitten todella työstiin sopivissa ryh-

missä. Melko tiiviin, mutta huolellisen työskentelyn tuloksena materiaali oli kasassa syksyllä 2001. Sen muokkaaminen kirjan muotoon sekä kaikki laadunvarmistus- ja kommentointikierrokset kestivät vielä muutaman kuukauden. Vuoden loppuun mennessä projekti saatiin käytännössä valmiiksi, minkä jälkeen vastuu siirtyi kustantajalle.

Hankintaoppaan sisältö

Tämän teoksen ehdottomana vahvuutena voidaan pitää käytännöllisyyttä. Siinä ei edes yritetä kertoa kaikkea kaikille, kuten muutamissa muissa hankintaohjeistoissa joihin projektin aikana törmäsimme. Niinpä kirja onnistuttiin pitämään suhteellisen ohuena. Varsinaista tekstiä on vain 72 sivua, jonka lukemisen luulisi onnistuvan jokaiselta, joka tosissaan haluaa onnistua tietojärjestelmän hankintaprojektissa. Koska kirjan tarkoituksena on todella auttaa sellaisissa asioissa kuin tarjouspyyntöjen laadinta ja vertaileminen, siinä on lisäksi 42 sivua liitteitä. Niissä on hyviä esimerkkejä ja menetelmäkuvauksia omassa organisaatiossa käyttöön otettaviksi. Myös kaikki kuvauspohjat sisältävä CD-ROM sisältyy pakettiin. Itse asiassa koko teos on saatavissa organisaation käyttöön myös sähköisenä versiona.

Ehdottomasti keskeisin oppi, jonka saa vain tästä hankintaoppaasta, on 4V-hankintakonsepti. Tietojärjestelmän hankintaa käsitellään nelivaiheisenä kokonaisuutena. Vaiheet ovat VALMISTELU, VALINTA, VALVONTA ja VIIMEISTELY. Hieman rikastettuina ne ovat hankinnan valmistelu, toimittajan ja ratkaisun valinta, toimituksen valvonta sekä hankintaprojektin viimeistely. Kukin vaihe kuvataan kirjassa omassa luvussaan, mutta painopiste on selvästi kahdella

ensin mainitulla “veellä”. Tämä onkin järkevää, koska juuri niissä vaiheissa suurimmat epäonnistumiset on yleensä pohjustettu.

4V-konseptiin kuuluu, että hankinnan edistymistä tarkastellaan jokaisessa vaiheessa kolmella erillisellä tasolla. Ylimpänä tasona 4V:ssä on *hankinnan ohjaus*, joka luonnollisesti saa oman ohjauksensa varsinaisesta liiketoiminnasta. Liiketoiminnan ohjaus kuitenkin on rajattu hankintakonseptin ulkopuolelle. Hankinnan ohjauksen alapuolella on *projektin ohjaus* ja alimpana, joskin todellisuudessa käytännön tulosten kannalta tärkeimpänä *systemityö*, kuten me kaikki sytykeläiset ilman muuta tiedämme.

Erittäin tärkeinä osina 4V-konseptia voidaan pitää huolellista, vaatimusmäärittelyksiin perustuvaa projektin mitoitus- sekä viimeistelyvaiheissa tehtävää keskeisten kokemusten keräämistä. Käytännössä on todettu, että systemaattinen mitoitus- ja tekeminen projektien käynnistyessä eliminoi suuret kustannusylitykset, ja aikataulujen venyminenkin muuttuu huomattavasti harvinaisemmaksi kuin ilman mitoitus- ja kokemusten keräämistä.

lysointi on erityisen tärkeää niille organisaatioille, joilla on toistuvasti uusia tietojärjestelmähankkeita. Tällä tavalla 4V opastaa organisaatioita jatkuvan kehittymisen tielle.

4V-konseptin rakenne esitetään kuvassa 1. Olisi erittäin tärkeää, että sekä tilaajat että toimittajat ottaisivat sen omakseen. Se on nimenomaan tarkoitettu yhteiseksi toimintamalliksi. Suuri osa menneisyyden ongelmista on johtunut nimenomaan siitä, että tilaajilla ei ollut mitään toimintamallia ja jokaisella toimittajalla taas omansa, keskenään erilaiset. Näin esimerkiksi tarjousten vertaileminen on ollut melkolailla arpapeliä, joka toivottavasti saadaan nyt siirtymään historiaan.

Kirjan luettuaan jokainen saa varmasti hyvän kokonaiskäsityksen tietojärjestelmän hankintaprosessista ja sen monipuolisuudesta. Hankinta on monella tavalla huomiota ja taitoa vaativa juttu. Esimerkiksi toimitettavan ohjelmiston laatuksymyksiin on osattava ottaa kantaa kaikissa vaiheissa. Siihenkin oppaasta löytyy jonkin verran tukea, mutta samalle ohjaustasolle kuin investointilaskelmien teossa ei tältä osin kyllä päästä. Kirjan lopussa

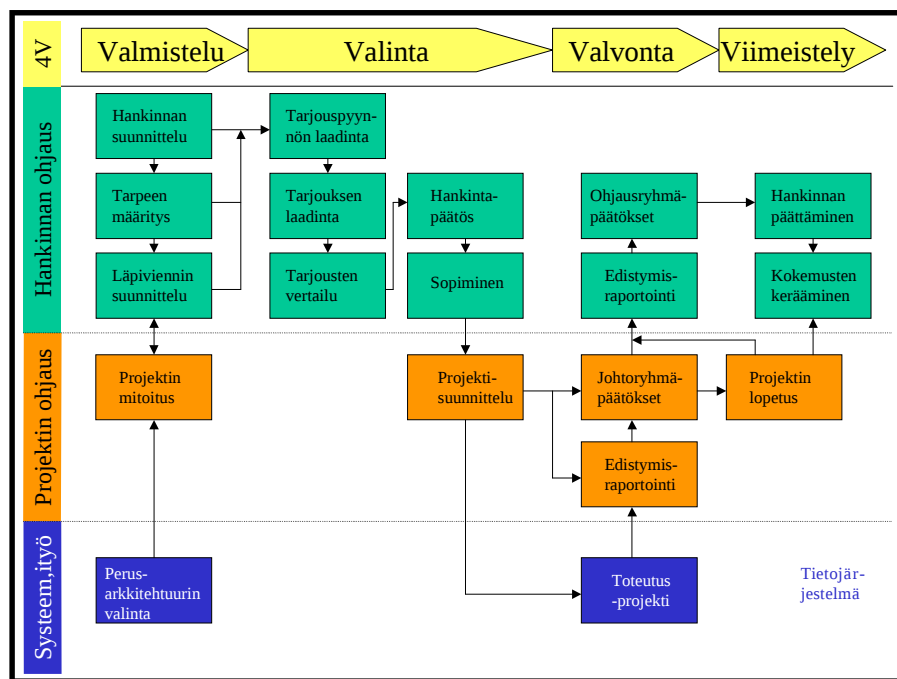
on myös lyhyt suomenkielinen sanasto käsittekarttoineen, ja kunkin termin englanninkielinen vastine. Kuvaukset ovat vain suomeksi.

Kirjan vastaanotto

Ensimmäisen painoksen suuruudeksi määriteltiin 500 kappaletta, koska opas on sen verran arvokas, ettei sitä luultavasti ihan heräteostoksena kukaan tule ottamaan. Sen on tarkoitus olla ennen kaikkea organisaatioiden työväline. Kirjan kysyntä on ylittänyt meidät mukana olleet positii-visesti, koska ensimmäinen painos on jo myyty loppuun ja toinen tullut painosta. Kirjan käyttöönoton tueksi on myös järjestetty yksipäiväisiä kursseja, joiden suosio on koko ajan kasvussa. Ensimmäinen järjestettiin jo maaliskuussa, ennen 1. painoksen ulos tuloa, noin kymmenpäiselle joukolle. Kesäkuussa oli toinen ja lokakuussa kolmas tilaisuus, johon tulijoita olikin yli 30. Koulutustilaisuudessa tulevat kaikki keskeiset ideat läpikäydyiksi ja samalla on mahdollisuus myös keskustella aiheesta kollegoiden kanssa, sekä kysyä lisätietoja eri asioiden taustoista itse kirjan teossa mukana olleilta.

Yleisesti ottaen vastaanotto on siis ollut hyvin myönteinen, eivätkä tuntemani kirjan ostajat tai kurssille osallistuneet ole sanoneet katuvansa kauppiaan. Päin vastoin, he ovat pitäneet tuotetta onnistuneena ja ennen kaikkea hyödyllisenä. Siinä on hieman uusia näkökulmia ja ajattelemisen aihetta jopa niille organisaatioille, joilla hankintakäytännöt on omasta toimesta ohjeistettu jo kauan sitten. Ehdottomasti suositeltavaa lukemista liian kiireen rasittamille tietotyöläisille. Ehkäpä ne turhat kiireet juuri tämän avulla vähenevät ratkaisevasti. Loppuu se surkea ja valittelu vihdoin!

Pekka Forselius,
STTF Oy,
pekkaf@sttf.fi



Kuva 1: Tietojärjestelmän hankintaprosessi

Ostaisin pitävän lupauksen

*Minna Alenius,
Lindström*

Ostaminen

Tämän päivän IT-päätäjän ongelmana on informaatiotulva ja tarjonnan runsaus; yhteydenottoja ja tietoja tulvii kaikkialta; postitse, sähköpostitse ja puhelimitse. Tarjoavan toimittajan puolella kannattaakin huolella miettiä, kuinka ja missä asiassa asiakasta lähestyy, sillä paljon jää pinoon käsittelemättä, menee roskiin, jää huomiotta ja jopa turhaan ärsyttää vastaanottajansa.

Nykyisin miltei kaikilla toimittajilla on jo asiakaslähtöinen toimiala-organisaatio ja periaatteessa asiakkaalle tarjotaan yhden vastuuhenkilön kautta ostamisen helppous. Jos pitäisi äänestää, keneltä on helpointa ostaa, niin parhaita kohtaamiani toimijoita tällä yhden vastuuhenkilön kautta kulkevalla yhteistyöllä ovat kilpailevat saksalainen ja amerikkalainen pakettitoimittaja. Kaikki toimittajat täytyvät kuitenkin osaa; mitä asiakas tekee yhteyshenkilöllä, joka ei yksin pysty asiakkaalle vastaamaan mihinkään kysymykseen? Olen ollut palaverissa, jossa toimittajan edustajia oli kahdeksan ihan varmuuden vuoksi ja asiakkaalta vain kaksi henkilöä. Palaverin jälkeenkin en ymmärtänyt, miksi kaikkien kahdeksan henkilön piti olla paikalla.

On myös rohkeita tuotelähtöisiä yrityksiä. Asia ei sinänsä ole paha, jos kokonaisuuskin hoituisi. Olen kuitenkin törmännyt toimittajiin, joilla on salkussaan vain se tietty tuote ja muutoin myyjää ei kiinnosta auttaa asiakasta vaikka myyjän yrityksen kannalta olisi kysymyksessä suurempi kokonaisuus. Sinnikäs asiakas ei kuitenkaan luovuta; vaikeatakin ostamista kannattaa tehdä, jos kyseessä on asiakkaan kannalta hyvä lopputulos.

Ostamista vaikeuttava tekijä on myös oikean kontaktihenkilön löytäminen myyjäpuolen organisaatiosta. Aikaisempi kokemus it-maailmasta auttaa

ostajaa ja avaa yhteyksiä, mutta kuinka helpottaa esimerkiksi it-vastuullista talousjohtajaa, joka ei it-yritysten organisaatorakenteisiin ole perehtynyt?

Lupaus

Ostajaa on aina vastassa myyjä ja myyjähän ovat lupauksen ammattilaisia. Lupauksen saaminen ei siis lähtökohtaisesti ole ongelma. Ongelmana onkin ko. lupauksen siirtäminen ymmärrettävässä muodossa sopimuspaperille ja sittemmin toimivaan käytäntöön. Toivoisin tapaavani jo lupauksista neuvoteltaessa suoraan ns. lunastusosaston; seuraavan kappaleen pitävyyteen olisi näin huomattavasti helpompi päästä.

Päätöksentekovaiheeseen siirryttäessä ostajan kannattaa todella miettiä, kenen kanssa ja mistä lupauksista keskustelea sekä todentaa, että sovitut asiat tulevat sovitusti paperille. Liian usein unohtuu ja seuraaville henkilöille ei siirry tieto siitä, mistä oikeastaan sovittiin. Arvokkaita ovat myös ne toimittajat, jotka eivät asiakkaan tietämättömyyttä lyhytnäköisesti hyväksikäytä, vaan valistavat asiakasta pitkäkestoisen yhteistyön kannalta olennaisiin asioihin; tässäkin asiassa usein myyjän lunastusosaston tietämystä tarvittaisiin.

Lupauksen pitäminen eli lunastus

Tämä on ostajalle ja asiakkaalle ehdottomasti tärkein asia ja tästä jopa voisi maksaa loistotoimittajan lisää, kunhan sovituksessa pysyttäisiin. Asiakkaan hankintapäätös ja sovitut asiat ovat liiketoiminnasta lähteviä ja siksi suuret poikkeamat vaikuttavat yleensä myös asiakkaan liiketoimintaan.

Kertakauppana tehtävien lisensoitavien pakettituotteiden ja laitteiden osalta tämä ei yleensä ole ongelma, kunhan lupaukset on saatu yhteisesti ymmärrettävässä ja sovituksessa muodossa paperille.

Jatkuvien palveluiden osalta ongelmia syntyy, kun lupaus on pitkäkestoinen ja sen lunastamiseen osallistuvat henkilöt/palvelut voivat ajanmitaan muuttua. Joskus palvelu alkaa sovitulla pelisäännöllä, mutta laajentuaan palvelukokonaisuus karkaa hintoineen kaikkineen asiakkaan käsistä. Joskus taas pitkään jatkuneen palvelun lähtötilanne on kaikille osapuolille jo niin hämärä, että vasta irtisanomalla vanha ja sopimalla uudelleen ko. palvelu, voi selkiyttää tilanteen. Asiakkaan siis kannattaa säilyttää itsellään mahdollisuus (muuttamalla tai irtisanomalla) uusia lupauksen sisältöä tarvittaessa.

Projekti (olipa valmissovelluksen implementointi tai räätälöity sovellys) osalta lupauksen pitämisen ongelma on suurin! Arvostaisin toimittajaa, joka voisi heti referoida aikataulun ja budjetin pitäviä hankkeita; nykyisin ei enää teknologia (olipa minkä lyhenteinen hyvänsä) säväytä yksinään ollenkaan. Projekteissa lupauksen pitävyys korreloi suoraan nimetyn projektipäällikön ammattitaidon kanssa; projektin kokonaisuus ja tehtävät pitää hallita, asioita pitää kyetä ennakoimaan, asiakkaalle pitää jopa osata sanoa tarvittaessa ei – hyvissä projekteissa ei syyllisiä haeta vaan asiat sujuvat. Hyvä projektipäällikkö on helmi, jota asiakaskin vaalii mielellään sellaisen kerran löydettyään.

Lopuksi

Toivoisin siis seuraavan suuren hankintapäätöksen yhteydessä tapaavani IT-toimittajan, jonka valttina on selkeä lupauksen pitäminen. Tällaisen toimittajan mielelläni myös muulle business-johdolle esittelisin; sen verran skeptisiä kommentteja olen IT-projektien läpiviemisestä kuullut. Toki olen havainnut myös, että asiakkaalta yhtälailla vaaditaan sitoutumista ja omalta osaltaan annettujen lupauksen pitämistä.

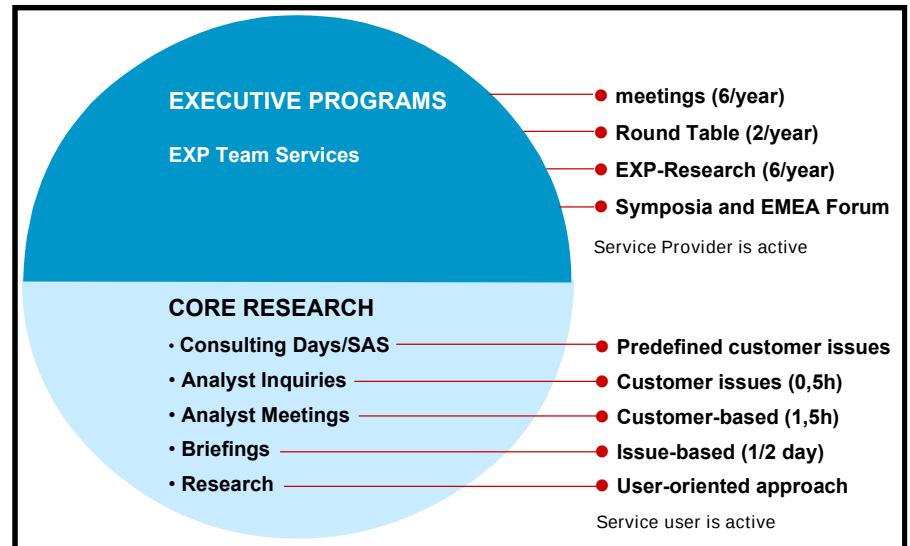
*Minna Alenius, CIO,
Lindström*

Mistä apuvälineitä tietotekniikan ja tietohallinnon johtamiseen

Pekka Sipola,
Market-Visio

Tietotekniikka ja tietoliikenne (ICT) - toimiala on tasaantumassa asteittain viime vuosien e-hypestä. Yritysten liiketoiminta- ja tietohallintojohdon kannalta seuraavien vuosien epävakainen tilanne toimialalla tulee vaatimaan jatkuvaa alan kehityksen seuraamista sekä strategisten ja taktisten päätösten tekemistä eri teknologia-vaihtoehtojen välillä. Gartnerin tutkimuksen mukaan vuosien 2002-2003 aikana tietohallintojohdon tärkeimpiä kehittämisen osa-alueita ovat mm. liiketoiminnan ja tietohallinnon välisen yhteistyön tiivistäminen, tietotekniikan investointien lisäarvon osoittaminen liiketoiminnalle, kustannusten alentaminen ja muutosprojektien parempi hallinta. Teknologian näkökulmasta katsottuna työlistalta löytyy mm. tietoturvan, järjestelmien välisen integraation, verkkoinfrastruktuurin sekä erilaisten portaalien kehittäminen.

Gartner on kehittänyt tietohallintojohdolle analyyttiseen tutkimukseen ja henkilökohtaiseen palveluun perustuvan palvelupaketin, josta käytämme nimeä *Executive Program – EXP*. EXP – palvelun tavoitteena on tarjota kiireisille tietohallintojohtajille alan parasta ja riippumatonta tutkimuspalvelua tärkeimmiksi koetuista aihealueista sekä yksilöllistä palvelua ja tutkimusmateriaaliin perustuvia esityksiä. Palvelu mahdollistaa asiakkaalle en-



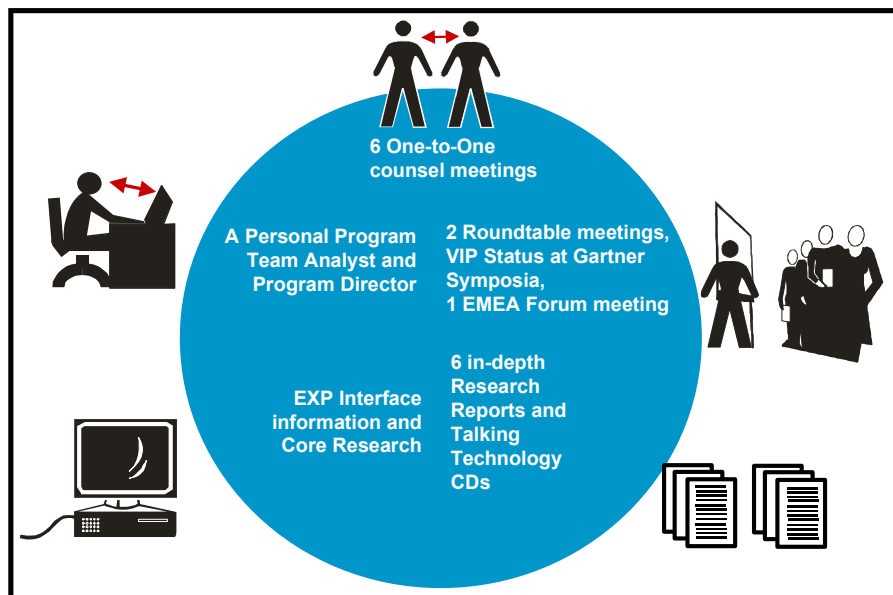
tistä paremmin uusien ideoiden hyödyntämisen, valmiiden toimintamallien käyttöönoton sekä pysymisen ajan tasalla teknologian kehittymisessä.

Gartnerilla on tällä hetkellä maailmanlaajuisesti n. 1700 EXP-asiakasta. Euroopassa asiakkaita on n. 700, Pohjoismaissa yhteensä n. 60 ja Suomessa 15 ”expilaista”. Suomessa asiakkaiden toimialat muodostuvat mm. rahoituksesta, tukkukaupasta, vähittäiskaupasta, teollisuudesta sekä julkisesta sektorista. Toimialojen erilaisuus kuvastaa sitä että useimmilla toimialoilla tietotekniikkaan ja tietohallintoon liittyvät haasteet ja mahdollisuudet ovat samankaltaisia vaikkakin toimialojen sisäiset erilaisuudet vaikuttavat tietotekniikan kehittämisen painopistealueisiin.

Mitä EXP-palveluun kuuluu ja miten se käytännössä toimii?

EXP – palvelun pohjana on Gartnerin tutkimus- ja analysointitieto, johon tietoon EXP-asiakkaalla on pääsy henkilökohtaisella tunnuksella. Tutkimustiedon lisäksi on mahdollisuus keskustella analyytikkojen kanssa joko puhelin - tai videoneuvotteluissa, osallistua analyytikkojen pitämiin Briefing-tilaisuuksiin tai järjestää henkilökohtaisia tapaamisia analyytikkojen kanssa.

Edellämainittujen tutkimustietojen ja analyyttikotapaamisten lisäksi palveluun kuuluu mm. 6-10 EXP - erityisraporttia, jotka käsittelevät eri aihealueita tietotekniikan ja tietohallinnon kehittämisestä. Lisäksi palvelu mahdollistaa osallistumisen kahteen



Suomessa järjestettävään Round Table – tilaisuuteen, joissa esiintyjinä on joko Gartnerin analyytikko tai konsultti, sekä osallistumisen Gartnerin Symposiumiin esim. Firenzessä tai Cannesissa. Asiakas ja palvelujohtaja pitävät vuosittain myös kuusi palaveria, jotka voivat olla joko kahdenkeskisiä palaverieita tai ICT – aiheisia esityksiä pienemmälle kuulijakunnalle.

EXP – palvelun käynnistäminen tapahtuu yleensä ns. Kick-Off-tilaisuuden avulla. Tilaisuudessa mm. kartoitetaan asiakkaan aihealueet, joista tutkimustietoa halutaan etsiä, koulutetaan EXP-palvelun internet-käyttöä sekä sovitaan kommunikointitavoista ja palaverikäytännöistä. Palvelun käynnistämisen jälkeen palvelujohtajan tehtävänä on hyödyntää Gartnerin kansainvälistä EXP – ryhmää Englannissa vastauksien löytämiseksi kysymyksiin,

joita asiakas on määrittänyt mm. Kick-Off-tilaisuudessa. Englannin EXP – ryhmä on vain Gartnerin EXP-asiakkaiden käytettävissä. Vastaukset toimitetaan asiakkaalle joko sähköpostilla, puhelinneuvottelun välityksellä tai palvelujohtaja toimittaa ne asiakkaalle seuraavan EXP-palaverin yhteydessä. EXP – palvelu on tarkoitettu tietohallintojohtajille, jotka toimivat alati muuttuvassa ympäristössä ja joiden aika ei riitä jatkuvaan tiedon hakemiseen tutkimuskannasta tai muista tietolähteistä. Yhteistoimintaan perustuva operatiivinen toimintamalli mahdollistaa myös sen että Market-Vision ja Gartnerin organisaatio sekä Suomessa että muualla maailmassa tuottavat tiedon palveluna tietohallintojohdolle.

Suomessa EXP-toiminta on päässyt hyvään vauhtiin tämän vuoden aikana ja asiakaskunta on kasva-

nut nopeasti. Tyypillisesti asiakkaat haluavat tutkimus- ja analysointitietoja, jotka liittyvät kiinteästi meneillä oleviin tai lähitulevaisuuden ICT- kehitysprojekteihin. Toisaalta halutaan myös nopeasti erilaisia vertailutietoja eri sovellusten ja tekniikoiden toimittajista, ominaisuuksista ja kehitysnäkymistä. Asiakkaat ovat myös hyödyntäneet EXP-yhteisön kansainvälistä verkostoa ja vaihtaneet ajatuksia eri yritysten vastaavien henkilöiden kanssa päivänpolttavista aiheista. Miellyttävää on myös todeta että yhteisön jäsenet ovat omatoimisesti verkostoituneet Suomessa keskenään ja järjestävät yhteisiä tapaamisia tutustuakseen toistensa tietotekniikan ja tietohallinnon ratkaisuihin.

Pekka Sipola

Gartner Inc. on globaali ICT – alan johtava analysointi- ja tutkimusyriyys, jolla on 80 toimipistettä 75 eri maassa ja jonka palveluksessa on 4600 henkilöä. Gartner Inc:n asiakaskunta muodostuu 11.000 asiakasorganisaatiosta ja yli 70.000 henkilöasiakkaasta ja sen liikevaihto oli v. 2001 n. 952 milj. USD. Market-Visiolla on ollut Gartner-palvelujen edustus Suomessa vuodesta 1989.

Kirjoittaja toimii Market-Visiossa johtajana toimialueina Gartner-konsultointi ja EXP-palvelut Suomessa.

Oliomallinnuksesta hyvä relaatiotietokanta

*Riitta Kylmälä,
Kela*

Rela-kerhossa on pohdittu sitä, miten oliomallinnus vaikuttaa relaatiotietokannan syntyyn. Ongelmaa lähestyttiin teorian ja esimerkkien kautta: luokkakaaviosta ja käyttötapauksista levyhakujen kesto.

Kerholaisten kirjoittamissa artikkeleissa annetaan suosituksia kuvaamiseen ja toteuttamiseen.

Oliosuunnittelusta tehokkaaseen tietokantaratkaisuun

Artikkelissa kuvataan oliosuunnittelun ja -ohjelmoinnin sekä relaatiotietokannan suunnittelun välisiä rajapintoja. Käytössä olevat oliosuunnittelumenetelmät eivät näytä suoraan johtavan relaatiotietokannan taulukkaavioihin. Oliomallinnus johtaa tietynlaiseen komponenttiajatteluun, jossa kuhunkin tauluun liittyy omat metodinsa. Usein kuitenkin tehokkaampi ratkaisu on yhdistää asioita samaan tauluun I/O:n säästämiseksi.

Ratkaisumalliksi esitetään perinteisesti käytettyä I/O-moduuliajattelua. Ratkaisulla päästään siihen, että ulospäin tietokanta näyttää noudattavan

loogisen tason kaaviota, mutta takana onkin suorituskykyistä teknisesti erilainen todellinen tietokanta.

Relaatiotietokannan suunnittelu

Tietokannan suunnittelu on oma tehtäväkokonaisuutensa. Lähtökohtana on oliomallinnuksessa syntyvät luokkakaaviot. Artikkelissa esitellään luokkakaavion muunnos relaatiotietokannan kaavioiksi sekä käydään läpi vaihtoehtoisia tapoja muuntaa attribuutteja ja yhteyksiä. Luokkakaavioiden lisäksi tarvitaan informaatiota olioiden käyttäytymisestä ja elämänsärestä.

Artikkelin lopussa on lyhyt katsaus relaatiotietokannan oliolaajennukseen.

Luokkahierarkiasta taulurakenne

Esimerkeissä lähdettiin liikkeelle luokkakaaviosta ja käyttötapauksista ja luotiin taulurakenne.

Taulurakennetta verrattiin suoraan luokkakaavion perusteella laadittuna sekä I/O-käsittely minimoiden. Oleellisena laskentaperusteena taulujen suunnittelussa on I/O-lukumäärät / tapahtuma.

Käytännön kokeilu Rational
Rosella

Teoreettisen lähestymistavan lisäksi kokeiltiin, millainen relaatiotietokanta syntyy sovelluskehittimellä. Kokeilussa käytettiin Rational Rose -kehittäjä.

Kehitin tuotti kolmen taulun ratkaisun, joka esimerkkien perusteella todettiin tehottomimmaksi. Useimmista tapauksista on tarpeen tehostaa tietokantarakennetta. Kuvausmuutokset tehdään tietokantakaavioon ja luokkakaavio jätetään alkuperäiseksi.

Paksujen indeksien jälkeen : paksut taulut

Lopuksi Tapio Lahdenmäki perustelee artikkelissaan paksujen taulujen käyttöä. Hajalevylukujen määrä on oleellinen tehokkuustekijä ellei koko tietokanta ole keskusmuistissa. Lisäksi optimoijaongelmien riski kasvaa kun SQL-lauseessa viitataan useaan tauluun.

*Riitta Kylmälä,
Kela*

Oliosunnittelusta tehokkaaseen relaatiotietokantaratkaisuun

Ari Hovi,
Ari Hovi Oy

Oliosunnittelu vs. relaatiokanta

Tietotekniikan koulutuksessa painotetaan nykyisin melko vahvasti oliosuunnittelua. Toteutettavien sovelusten tietokannat ovat kuitenkin pääosin relaatiokantoja. Puhdaosoppiset oliotietokannat eivät ole saaneet merkittävää suosiota ainakaan valtavirran kaupallis-hallinnollisissa sovelluksissa. Ohjelmointi sen sijaan on yhä useammin oliopohjaista. Ohjelmointia edeltää oliosuunnittelu.

Käytössä olevat oliosuunnittelumenetelmät eivät näytä muodostavan ehyttä polkua relaatiokannan taulukkaavioihin saakka. Nämä menetelmät päättyvät luokka-, toiminto- ja tilakaavioihin, jotka eivät riitä tai suoraan sovi relaatiokannan pohjaksi. Käytännön suunnittelussa usein ensin laaditaan oliomallin mukaisesti erilaiset kaaviot ja sitten hieman erillisenä prosessina tehdään relaatiokannan toteutusta varten käsitelmä (EAR-kaavio). Oliosuunnittelun ja relaatiokantojen väliin näyttää jäävän ”harmaa” alue, jota eivät oliosuunnittelijat hallitse ja jonka relaatiokantasuunnittelijat saatavat täyttää väärillä oletamuksilla.

Oliosunnittelussa tietokantojen suunnittelun osuus näyttää jäävän hyvin vähäiseksi. Aiemmin relaatiokantojen suunnitteluun panostettiin ja tehtyjen virheiden kautta myös opittiin

suunnittelemaan suorituskkyisiä tietokantaratkaisuja. Suorituskyky onkin tärkeää, sillä relaatiokannoilla on hyvin helppo toteuttaa sinänsä toimivia, mutta hitaita ja raskaita järjestelmiä. Usein tietokantaratkaisu, jota yritetään laitteistoa kasvattamalla nopeuttaa, olisikin viritettävissä oikealla taulu- ja indeksisuunnittelulla kertaluokkaa nopeammaksi – ilman laitteistoinvestointeja. Parhaimmillaan tehoa saadaan lisää koskematta lainkaan sovelluskoodiin, muutokset tulevat vain indeksirakenteisiin. Tämä edellyttää kuitenkin asiantuntijatason tietokantasaamista, joka tuntuu nykyisin jääneen oliosuunnittelusuuntauksen jalkoihin.

Palatkaamme takaisin oliosuunnittelun lopputuloksiin ja toisaalta tietokantasuunnittelun edellyttämiin lähtötietoihin, eli tähän ”harmaaseen” alueeseen. Oliosuunnittelun yksi lopputulos, luokkakaavio, sisältää usein periytymistä kuvaavia rakenteita, joita ei ole perinteisessä EAR-kaaviossa ja joita ei voi aivan suoraan toteuttaa relaatiokannoilla. Suoraviivaisin tapa muuntaa luokkakaavio EAR-kaavioksi johtaa helposti hyvin tehottomaan toteutukseen. On suuri vaara, että näin saadaan aikaan hidas ja runsaasti turhia levylukuja aiheuttava ratkaisu. Oliomallinnus johtaa tietynlaiseen komponenttijaotteluun, joka tuntuu toimivan parhaiten toteuttamalla paljon ”pieniä” tauluja, joihin sitten liitetty kuhunkin omat metodinsa. Tyypillisesti tehokkaampi ratkaisu olisi yhdistää sopivasti asioita samaan tauluun

I/O:n säästämiseksi (tämä ei tarkoita denormalisointia). Tämä taas hämärtää komponenttijaottelua kun metodit eivät enää liitykään suoraan tiettyihin tauluihin.

Oliosunnittelusta relaatiotietokantaratkaisuun

Olemme kehittäneet tarkistuslistoja ja ohjeita miten välttää pahimmat sudenkuopat luokkamallia käytettäessä. Ohjeittemme mukaan luokkakaaviosta muodostetaan EAR-kaavio, josta etenemällä relaatiokannan rakennekaavioon ja siitä itse tietokannan toteutukseen päästään suorituskkyisempään ratkaisuun.

Syntyvä relaatiokannan rakenne on erilainen kuin suoraan luokkamallista etenemällä syntyvä kaavio. Tämä on otettava huomioon metodien toteutuksessa; useille samantyyppisille metodeille tuleekin nyt yhteinen taulu.

Monet oliosovellukset ovat tiukasti sidoksissa tietokannan kaavioon. Tämä tekee tietokannan rakenteen muuttamisen hankalaksi, sillä tauluja vastaavia muutoksia pitää usein toteuttaa myös ohjelmalogiikkaan. Tavallisin ratkaisumalli on erityisen sovelluskerroksen rakentaminen sovelluksen ja luokkakaavion ja tietokannan kaavion välille siten, että ainoastaan tämä välikerros on riippuvainen tietokannan rakenteesta. Välikerros siis peittää tietokannan rakenteen. Tietokantaa käsitellään ainoastaan kutsu-

malla välikerrosta, ei suoraan SQL-kielellä.

Välikerros lisää sovelluksen mutkikkuutta, mutta se tarjoaa arvokasta joustavuutta. Erityisesti tilanteissa, joissa joudutaan ulottamaan käsittely useampiin tietokantoihin, saattaa riippumattomuustaso olla ratkaisevan tärkeä.

Yksinkertaisimmillaan välikerros syntyy siten, että kutakin tietokantakäsittelyyn liittyvää sovellusluokkaa

kohti tehdään oma tietokantaluokka, johon viedään kaikki tietokantakäsittely. Tällä ratkaisulla päästään siihen, että ulospäin tietokanta näyttää noudattavan suoraan luokkakaaviota, mutta välikerroksen ”takana” tietokanta onkin suorituskykyisistä hieman erilainen.

Edellä kuvattu ratkaisumalli on sukua ns. I/O –moduliajattelulle, jota on pitkään käytetty relaatiokantoja edeltäneiden verkko- ja hierarkkisten kantojen yhteydessä. Edellä kuvattu

välikerros toteutettiin erillisinä I/O-moduleina, joiden kautta kaikki tietokantakäsittely hoidettiin.

Toivottavasti tämän lehden artikkelit omalta osaltaan edesauttavat parempien ja tehokkaampien relaatiokantaratkaisujen toteuttamista oliosuunnittelun pohjalta.

*Ari Hovi,
Ari Hovi Oy*

Sinulla,

**joka olet kiinnostunut relaatiotietokannoista,
on mahdollisuus tulla mukaan Rela-kerhon toimintaan.**

**Kerho on epämuodollinen keskustelu- ja opintopiiri, joka kokoontuu
noin kerran kuussa.**

**Tässä lehdessä on joukko artikkeleja, jotka syntyivät olioteeman
ympäriältä. Nyt on teemana koosterakenteet:**

- * näkymät,**
- * sql-select lauseen taululauseke,**
- * relaatiotietokantojen oliolaajennukset.**

Kerhon kokouksiin osallistuminen ei velvoita mihinkään.

**Ilmoittaudu sähköpostijakelulistalle lähettämällä viesti
marja.karmeniemi@oracle.com,
niin saat tietoa tulevista tapaamisista.**

Oliosunnittelusta tietokanta

Jaakko Rantanen,
Hämeen ammattikorkeakoulu,
Tietojenkäsittelyn koulutus-
ohjelma

Oletetaan, että tietokantasuunnittelun lähtökohtana on oliomenetelmistään perustuva suunnittelu ja sen tavallinen dokumentaatio. UML (Unified Modeling Language) on saavuttanut muutamassa vuodessa suuren suosion ja onkin erittäin hyvä, että kuvausmenetelmien tavattomaan kirjavuuteen tulee yhtenäisyyttä. Tässä rajoitutaan tarkastelemaan UML-dokumentaatiota.

Tietokantasuunnittelun tärkein lähtökohta on luokkakaavio (class diagram), joka esittää staattiset tietora-

kenteet. Luokat ovat relaatiotietokannan suunnittelijan näkökulmasta tauluehdokkaita. Luokkakaavio ei välttämättä sellaisenaan sovellu tietokantakaavioksi. Luokkakaavion muuntaminen tietokantakaavioksi on menetelmällisesti suoritettavissa.

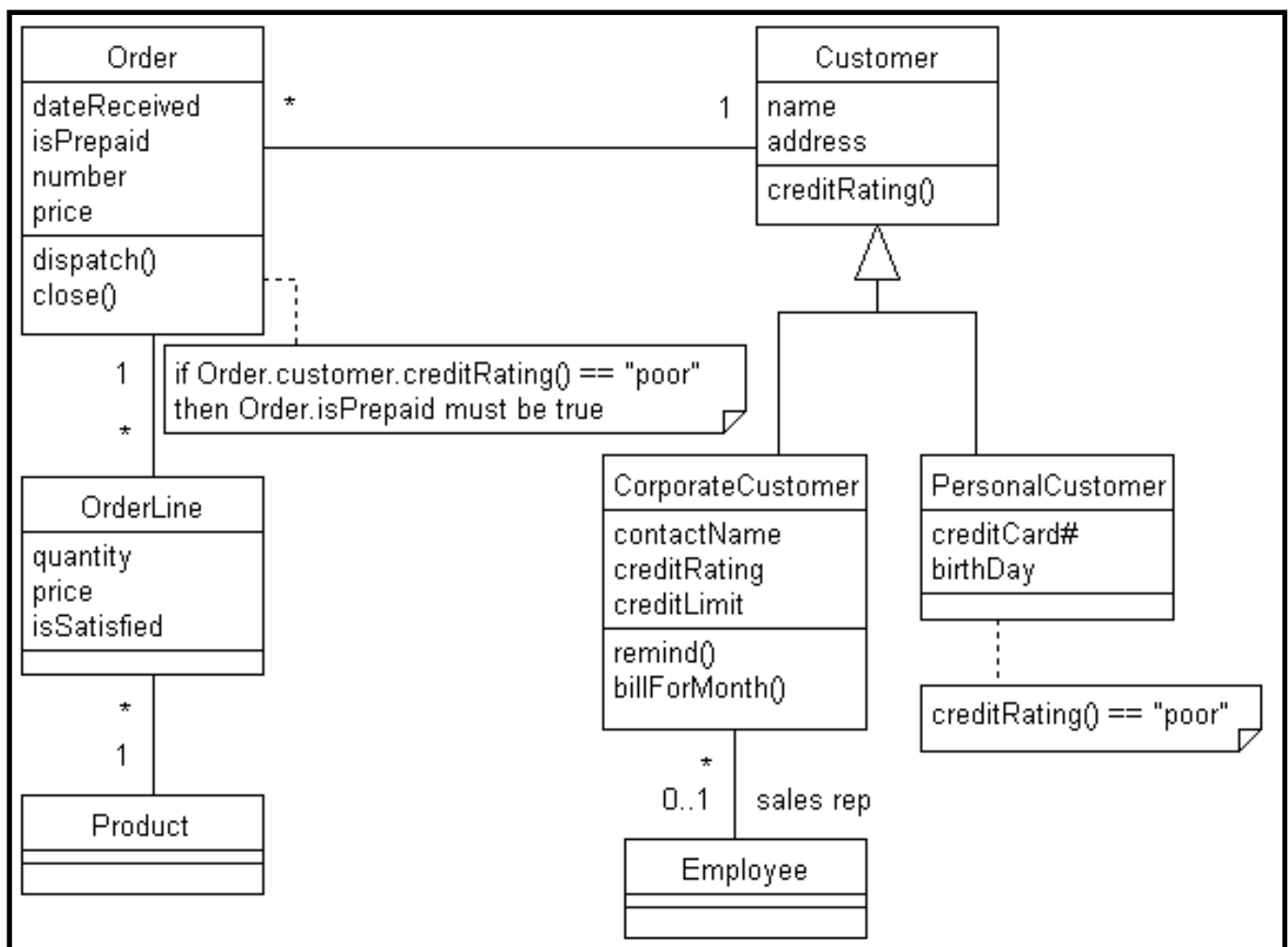
Luokkakaaviosta relaatiotietokannan kaavio

Järkevä luokkakaavion muunnos tauluiksi ei ole mekaaninen suoritus. Tietokantakin kaipaa suunnittelua. Kun otetaan huomioon, että usein tietokannan pitäisi palvella useita sovelluksia, joutuu tietokannan suunnittelija katsomaan asioita yleistävästä näkökulmasta.

Otetaan lähtökohdaksi muunnos, jossa jokaista luokkakaavion olioluokkaa kohti muodostetaan relaatio. Tätä periaatteesta joudutaan tosin poikkeamaan muutamissa tilanteissa.

Oliokäsitteistöä kuvataan parissa esimerkissä ODL-kielellä (Object Definition Language), joka on ODMG-standardin (Object Data Management Group) määrittelykieli oliotietokannoille.

Esimerkeissä käytetään seuraavaa yksinkertaista luokkakaaviota: Kts. kuva alla.



Kuva. Lähde: UML Distilled by M. Fowler, Addison Wesley 1997

Perustyyppiset attribuutit

Jos relaatioesityksessä noudetaan ensimmäisen normaalimuodon asettamia rajoitteita, ts. attribuuttien atomisuusvaatimusta, ne ilmenevät seuraavina:

- Luokan kaikki attribuutit esitetään relaation attribuutteina, ei esimerkiksi yhteyksinä tai metodeina.
- Attribuuttien tyypit ovat perustyyppisiä, eivät rakenteisia tyyppiä.

Perustyyppiset attribuutit muunnetaan taulujen sarakkeiksi seuraavasti:

- Yksilöiviä attribuutteja vastaavat sarakkeet muodostavat taulun perusavaimen.
- Jokaista luokan yksiarvoista attribuuttia vastaa sarakke vastintaulussa.
- Jokaista moniarvoista attribuuttia kohti luodaan oma taulu, johon tulee
 - viiteavain viittaamaan luokkaa vastaavaan tauluun ja
 - sarakke esittämään kyseistä moniarvoista attribuuttia sekä
 - perusavaimeksi kaikki sarakkeet. Esimerkki: henkilöllä saattaa olla useita taitoja, kuten kielitaitoja. Taitojen esittäminen taulun yhdellä sarakkeella `employee (emp_id, skill, ...)` ei tällöin onnistu. Luodaan taitoja esittämään uusi taulu `skill (emp_id, skill)` jossa `emp_id` viittaa tauluun `employee` ja perusavain koostuu molemmista sarakkeista (`emp_id, skill`).
- Tarkistetaan, että luokkien välinen riippuvuus ei ole piilotettuna attribuutiksi. Samaa asiaa ei esi-

tetä sekä attribuuttina että yhteytenä.

- Tarkistetaan, että attribuutteina ei esitetä johdettavissa olevaa tietoa.

Rakenteiset attribuutit

Oliokäsitteistön rakenteiset attribuutit voivat olla esimerkiksi tietoryhmiä (record structure), joukkoja (set), monijoukkoja (bag, multisets), järjestettyjä joukkoja eli listoja (list), tai taulukoita (array). Monijoukko eroaa joukosta siten, että monijoukossa saa sama alkio esiintyä toistuvasti, joukossa ei.

SQL:1999-standardi määrittelee edellisistä tietoryhmän sekä taulukon (array), ei esimerkiksi joukkoa ja monijoukkoa. Kahden viime mainitun esittäminen onnistuu jossakin määrin perinteisin relaatiomallin keinoin.

Seuraavassa on esimerkki `PersonalCustomer`-luokasta, jossa osoite esitetään tietokoosteena. Henkilöllä voi olla joukko puhelinnumeroita.

```
ODL-kielinen luokan kuvaus:
interface PersonalCustomer {
    attribute int cust_id;
    attribute string name;
    attribute Struct Addr {string street,
                           string city} address;
    attribute Set <Struct Phone {string
                                type, int number}> phones;
}
```

Joukko- ja monijoukko ovat esitettävissä keinotekoisella tavalla relaatiotietokannassa ainakin silloin, kun alkiot ovat perustyyppiä. Esimerkin henkilöasiakkaiden puhelinnumerot olisi esitettävä omassa taulussaan, johon pitää luoda lisäksi viiteavain esit-

tämään kenen mikin puhelinnumero on. Jos puhelinnumerot muodostavat joukon, ei puhelinnumero saa toistua. Tästä voi huolehtia yksikäsitteisyysrajoitteella (unique tai primary key). Jos kyseessä olisi monijoukko (bag), yksikäsitteisyyttä ei vaadittaisi.

Tietoryhmän (record structure) esittämisestä SQL:1999 keinoin on esimerkki jäljempänä kappaleessa, joka käsittelee SQL-standardointia. Seuraavassa esimerkissä tarkastellaan yhtä tietoryhmän esittämistapaa, jos joudutaan rajoittumaan ensimmäiseen normaalimuotoon.

Kun luokka muunnetaan tauluksi, tietoryhmän `address` voi esittää kahtena itsenäisenä sarakkeena: `street`, `city`, mutta ryhmän nimi `address` ei sellaisenaan muuntuisi. Ongelmia voisi tällöin koitua samannimisyydestä, jos samaan tauluun tulisi samannimiset sarakkeet kahdesta eri ryhmästä. Tämän estämiseksi ja johdonmukaisuuden vuoksi voisi ryhmän nimen (aina) viedä sarakkeiden osiksi esimerkiksi seuraavasti: `street_address`, `city_address`.

Lista-käsite esiintyy tyypillisimmin käyttöliittymien olio-ohjelmoinnissa esimerkiksi valintalistoina. Listan arvojen säilyttäminen relaatiotietokannassa on luonnollista ja ne haetaan tyypillisesti joukkona. Listojen käsittelyoperaatioissa on ajatus lähempänä käyttäjän tekemisiin reagointia kuin tiedon tallettamista.

Yhteydet

Yhteydet esitetään relaatioiden välillä ainoastaan viiteavaimiin perustuvilla arvoviittauksilla. Viittauksen kohteen on oltava yksikäsitteinen ja aina olemassa. Viiteavain voi muodostua usean sarakkeen yhdistelmästä.

Yhden suhde moneen -yhteydet

Kahden luokan välisen assosiativisen yhteyden voi taulujen välille toteuttaa kahdella perustavalla: joko muodostamalla viiteavain tai luomalla erityinen yhteyttä kuvaava taulu. Lähes aina selvittää viiteavaimella eikä yhteyttä kuvaavaa taulua tarvita.

- Ensin mainitussa tapauksessa lisätään viiteavain siihen tauluun, jota vastaa korkeintaan yksi osapuoli. Esimerkiksi yhteys `order (0..*)—(1..1) customer` toteutetaan viittaamalla `order` -taulusta `customer` -tauluun. Lukumääräisyys 1..1 toteutuu kieltämällä ko. viiteavaimelta tyhjäarvo. Vastaavasti `sales_rep -yhteys corp_cust (0..*)—(0..1) employee` toteutetaan viittaamalla `corp_cust` -taulusta `employee` -tauluun, mutta lukumääräisyys 0..1 toteutuu sallimalla ko. viiteavaimen tyhjäarvo.
- Yhteyttä kuvaavaa taulua saatetaan tarvita esimerkiksi siksi, että yhteyteen halutaan liittää attribuutteja. Käytettävän tietokantajärjestelmän määrittyskielen rajoitukset voivat myös vaatia taulun käyttöä, jos syklisten yhteyksien luonti ei onnistu (so. `alter table` -komennolla ei voi määrätä viiteavainta). Esimerkiksi yhteyden `employee (0..*)—(1..1) department` voisi josakin erityistilanteessa toteuttaa taululla `works(emp_id, dep_id)` missä `emp_id` sekä `dep_id` viittaavat kumpikin omiin tauluihinsa ja perusavaimena on pelkästään `emp_id`.

Kooste-tyyppisessä (aggregate) yhteydessä tarvitaan yleensä tieto kokonaisuudesta koosteen osien tunnistamiseksi, joten osia esittävissä taulussa oleva viiteavain kuuluu perus-

avaimen. Esimerkiksi yhteyden `order (1..1)—(1..*) order_line` toteuttaa taulussa `order_line (order_id, line_no, ...)` oleva viiteavain `order_id`, joka myös sisältyy `order_line` -taulun perusavaimen (`order_id, line_no`).

Yhden suhde yhteen -yhteydet

Yhden suhde yhteen -yhteydellä tarkoitetaan oikeastaan useimmiten ehdollista yhteyttä 1:0 (josta käytetään myös merkintää 1:C). Toinen vaihtoehto on molemmiin puolin pakollinen tapaus eli 1:1.

Lisätään viiteavain yhteyden osapuolia esittävistä siihen tauluun, jonka kannalta yhteys on pakollinen. Jos yhteys on pakollinen molempien kannalta, valitaan kumpi tahansa, mutta tällöin suositeltavinta olisi taulujen yhdistäminen.

Jos perustauluja yhdistetään, vaikka luokkakaaviossa esiintyy eri luokat, voi näkymien käytöstä koitua hyötyä. Esimerkki: Henkilöllä voi olla ajokortti tai sitten ei. Ajokortti on aina tietyn henkilön. Yhteys `person (1..1)—(0..1) drivingLicense` toteutuu helpoiten siten, että taulussa `drivingLicense` viiteavainsarake `pers_id` on yksikäsitteinen (unique) ja se viittaa tauluun `person`. Taulujen yhdistämistä kannattaisi myös harkita. Seuraavan kaltainen tilanne saattaa myös esiintyä: Henkilöllä voi olla enintään yksi PC käytössään, mutta kaikilla henkilöillä ei PC:tä välttämättä ole. Kullakin PC:llä on enintään yksi käyttäjä, siis PC voi olla kokonaan käyttämätönkin. Jos on olemassa eri taulut `person` ja `pc`, niin yhteyden `person (0..1)—(0..1) pc` tilanteessa syntyy tekninen ongelma. Tämä ratkaisu näet edellyttää tietokantajärjestelmältä sellaista harvinaista ominaisuutta, että `pc` -taulus-

sa viiteavaimen `pers_id` on oltava yksikäsitteinen (unique), mutta sallittava silti useita null-arvoja.

Monen suhde moneen -yhteydet

Kahden luokan välistä monen suhde moneen -yhteyttä esittämään muodostetaan oma taulunsa.

- Taulun nimeksi annetaan yhteyden nimi.
- Taulun sarakkeet koostuvat vähintään viiteavaimista ja jos pidetään tarpeellisena, tauluun voi luoda muitakin sarakkeita. Monen suhde moneen -yhteys on luonteeltaan yleistys useista tapauksista, joista jokainen kirjautuu yhteyttä kuvaavaan tauluun omana rivinään. Jokaisesta tapauksesta on mahdollisuus kirjata siihen liittyvää tietoa.
- Perusavaimen voi muodostaa useilla tavoilla.
 - Tavallinen ja yksioikoinen tapa on seuraava: kaikki viiteavainsarakkeet muodostavat yhteyttä kuvaavan taulun perusavaimen.
 - Toinen yleisesti esiintyvä tilanne on sellainen, jossa halutaan käyttää sarjanumerointia (surrogaattia) ja käyttää sitä perusavaimessa. Ks. alla esitetyn esimerkin taulu `order_line` jossa tilausrivit numeroidaan. Taulun `order_line` perusavaimena voi toimia joko yhdistelmä (`order_id, line_number`), jos rivinumeron halutaan juoksevan kunkin tilauksen sisällä. Perusavaimena voi periaatteessa toimia pelkkä `line_number`, jos rivinumeron halutaan juoksevan kaikkien tilausten yli.

Esimerkki: henkilö voi käyttää useaa autoa ja autoa usea henkilö. Tämä monen suhde moneen –yhteys `person (0..*)—(0..*) car` esitetään assosiatiiivisena tauluna `person_use_car` jonka attribuutteja ovat vähintään viiteavaimet (`person_id`, `car_id`), jotka yhdessä muodostavat perusavaimen. Autojen käyttötapauksiin saattaa liittyä talletettavia tietoja, joiden säilytyspaikka tämä taulu olisi. Esimerkiksi yleistetyn yhteyden `order (0..*)—(1..*) product` tarkemman esittämisen tarpeesta johtuen on jo luokkakaavioon kirjattu tilausrivi-luokka `order (1..1)—(1..*) order_line (0..*)—(1..1) product`.

Luokkahierarkia

Toisessa artikkelissa ”Hierarkista taulurakenne” vertaillaan tietokantaratkaisun käyttäytymistä, kun on noudatettu tässä lyhyesti kuvattavia perustapoja muuntaa luokkahierarkia tauluiksi.

Tarkastellaan kahta hierarkiata-soa: ylliluokka ja sen välittömät aliluokat. Yleisesti voi erottaa kolme perustapaa esittää hierarkia tauluina:

1. Luodaan jokaista luokkaa kohti oma taulunsa. Ylliluokkaa vastaavan taulun perusavain periytyy sellaisenaan aliluokkia vastaavien taulujen perusavaimiksi. Aliluokkia vastaavissa tauluissa perusavain toimii myös viiteavaimena.
2. Luodaan vain yksi taulu, jonka sarakkeet koostuvat sekä ylitettävien aliluokkien attribuuteista. Lisäksi on tarpeen tyyppiä esittävä sarake, joka ilmaisee mistä aliluokasta kunkin rivin kohdalla on kyse. Tällöin syntyy tyhjiä sarakkeita mikä sinänsä ei ole resurssiongelma kehittyneiden tietokantajärjestelmien tapauksessa. Tietojen säilyttämisen ristiriidattomana, ts. aliluok-

kien pitäminen erillisinä, on han-kalampi asia. Näkymät (view) tarjoavat mahdollisuuden säilyttää käsittely aliluokkien mukaisena, koska tässä tilanteessa näkymistä voi tehdä päivitettäviä. Sarakkeiden nimeämisestä voi koitua harmia, jos aliluokkia on paljon tai hierarkiatasoja monta. Saman taulun sarakkeiden on oltava yksilöllisesti nimettyjä. Luonnollista olisi sisällyttää aliluokan nimi sarakkeen nimeen, mutta käytännössä sarakenimen maksimipituus voi äkkiä ylittyä.

3. Luodaan vain aliluokkia vastaavat taulut, joista jokaiseen lisätään ylliluokan attribuutteja vastaavat sarakkeet. Ylliluokkaa ei esiintyisi, paitsi (projektoiden yhdisteenä) näkymä- tai tulostaulussa, joka voisi olla vain read-only -tyyppinen.

Tietokantakaavio on syytä piirtää erikseen, se ei yleensä ole samanlainen kuin luokkakaavio.

Luokkakaavion ohella

Laadukkaaseen tietokantasuunnitteluun tarvitaan informaatiota muistakin lähteistä kuin luokkakaaviosta. Olioiden käyttäytymiseen ja elämänsykliin liittyy tietokannan suunnittelijaa kiinnostavaa informaatiota.

Oliosuunnittelija ei välttämättä tule kirjanneeksi kaikkea tietokantasuunnittelijan kaipaamaa oleellista informaatiota. Jos UML-kaaviot oleellista asiaa sisältävätkin, mahdollisesti ongelma koituu siitä, että asiat esiintyvät sangen sirpaleisena.

Mitä sitten kaivattaisiin? Tässä lyhyt toivomuslista:

- Tietomäärät: miten paljon perusdataa on talletettu ja miten paljon sitä syntyy ja muuttuu esimerkiksi vuodessa. Erityisesti kriitti-

sen suuret määrät ovat kiinnostavia ja suuruusluokkien tietäminen riittää.

- Tapahtumatiheydet: miten paljon ja miten tiheästi käsittelytapauksia esiintyy. Taas kriittisen kuormittavat tilanteet olisivat tärkeitä tietää. Tietokantasuunnittelija hauaisi ulottaa ymmärtämisen käyttötapauksista aina tietokannan taulukohtaisen käsittelyn tasolle.
- Mitä attribuutteja tarvitaan kuskakin tapahtumassa ja mikä on käsiteltävä rivimäärä.

Perustiedoista on syytä esittää kertyneet lukumäärät tietyn ajan kuluksi. Tapahtumamäärät kannattaa ilmaista johdonmukaisesti per kiinteä ajanjakso, joka on esimerkiksi vuosi. Taulujen suunnittelijan pitää ajatella toimintoja transaktioina. Raportit ja muut haut, jotka kohdistuvat isoon tietojoukkoon, ovat kiinnostavia, pienet määrät eivät niinkään.

Oliorelationaalisia tekniikoita

Tehdään lopuksi muutamilla esimerkeillä katsaus SQL:1999 –standardin mukaisiin tekniikoihin. SQL:1999 on hyvin laaja. Tässä esitetään vain muutamia seikkoja, jotka liittyvät edellä olleisiin esimerkkeihin.

SQL:1999 on SQL-standardin kolmas päävaiheen toteutuma ja suuri uudistus, joka valmisteluvaiheessa on tunnettu työnimellä SQL3. Edelliset olivat SQL-92 ja SQL-89. Standardointityö jatkuu. Seuraavaa versiota rakennetaan työnimellä SQL:200n.

Taulukko (array) on yksi uusi tyyppi. Seuraavaan esimerkkiin on keksitty taulukko `billedLastMonth`, jonka voi kuvitella esittävän 12 viime kuukauden aikana tehtyjä laskutuksien rahasummia.

```
create table corporate-
Customer (
  custID          integer
                 primary
                 key,
  contactName     varchar
                 (30),
  creditRating    decimal
                 (15,2),
  creditLimit     decimal
                 (15,2),
  billedLastMonth decimal
                 (15,2) array
                 [12]
);
```

Taulukkoalkioita käsitellään normaalin ohjelmointikäytännön mukaisesti alkion järjestysnumeron (taulukkoindeksin) perusteella, esimerkiksi seuraavasti:

```
select custID, billed-
LastMonth[2]
from corporateCustomer
where billedLastMonth[2]
> 10000.0;
```

Tyyppien määrittely

Olioluokan varsinainen vastine SQL:1999ssä perustuu tyyppi-käsitteeseen (UDT, User Defined Type). Tyypin piirteisiin kuuluu paitsi attribuutit myös metodit. Kapselointi (encapsulation) toteutuu, mutta näkyvyys (esim. private/public-määrittelyt) ilmaistaan lupien hallinnalla (grant/revoke) eikä erillisillä näkyvyysmääreillä. Tyyppien välille voi määritellä periytymishierarkian. Monimuotoisuutta (polymorphism) on toteutettu Java-kielen tapaan siten, että metodien nimiä voi kuormittaa.

Tyyppi voi olla yksinkertainen tai rakenteinen. Esimerkkinä yksinkertaisesta tyypestä on seuraavat `euroType` ja `dollarType`:

```
create type euroType as
decimal(15,2) final;
create type dollarType as
decimal(15,2) final;
```

Eri tyypit eivät ole vertailukelpoisia, esimerkiksi seuraava ei olisi sallittu:

```
... where amountEuro =
amountDollar
```

Jos vertaaminen on tarpeen, on tehtävä tyypin muunnos. CAST-operaatiolla saataisiin aikaan laillinen vertailu, esimerkiksi tähän tapaan:

```
... where amountEuro =
CAST(amountDollar as eu-
roType)
```

Taulutyypit

```
create type customerType
as (
  custID integer,
  name varchar(30),
  address varchar(25),
  ref is system generated);
```

Taulutyypistä luodaan tyypitettyjä tauluja (jotka ovat varsinaisia olioluokkien vastineita), esimerkiksi seuraavasti:

```
create table customer of
customerType
ref is customerOID sys-
tem generated
<constraint-määrittely>;
```

Tyypitetty taulu tarvitsee object-id –määrittelyn, joka edellisessä esimerkissä on `customerOID`. Object-id voidaan luoda eri tekniikoilla.

Jos halutaan luoda taulutyypit `customerType`, `corporateCustomerType` sekä `personalCustomerType`, periytymishierarkia määritellään seuraavaan tapaan:

```
create type corporateCus-
tomerType under customer-
Type;
create type personalCus-
tomerType under customer-
Type;
```

Tyypitetyt taulut eivät ole ainoa vaihtoehto näiden tapausten esittämiseen, mutta suositeltavin.

SS S \$SSSS\$SSSSSSSS\$
esittelevää kirjallisuutta

Jim Melton, Alan R. Simon: “SQL:1999 – Understanding Relational Language Components”, Morgan Kaufmann, 2002

Jim Melton: “Advanced SQL:1999 – Understanding Object-Relational and Other Advanced Features”, Morgan Kaufmann, 2002

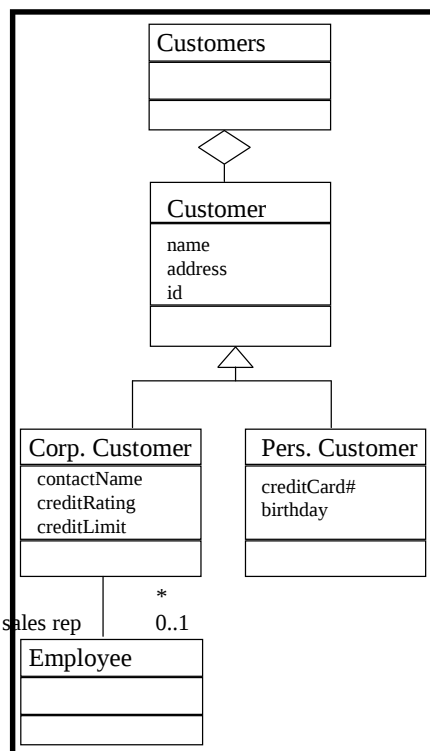
Peter Gultzan, Trudy Pelzer: “SQL-99 complete, really”, R&D Books, 1999

*Jaakko Rantanen,
Hämeen ammattikorkeakoulu,
Tietojenkäsittelyn koulutus-
ohjelma*

Luokkahierarkiasta taulurakenne

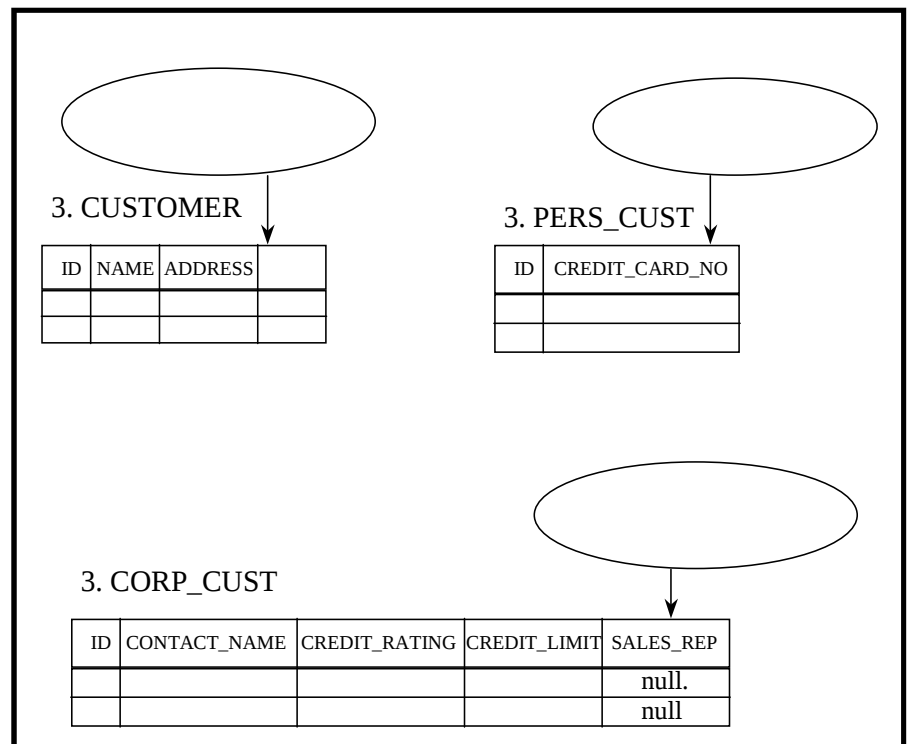
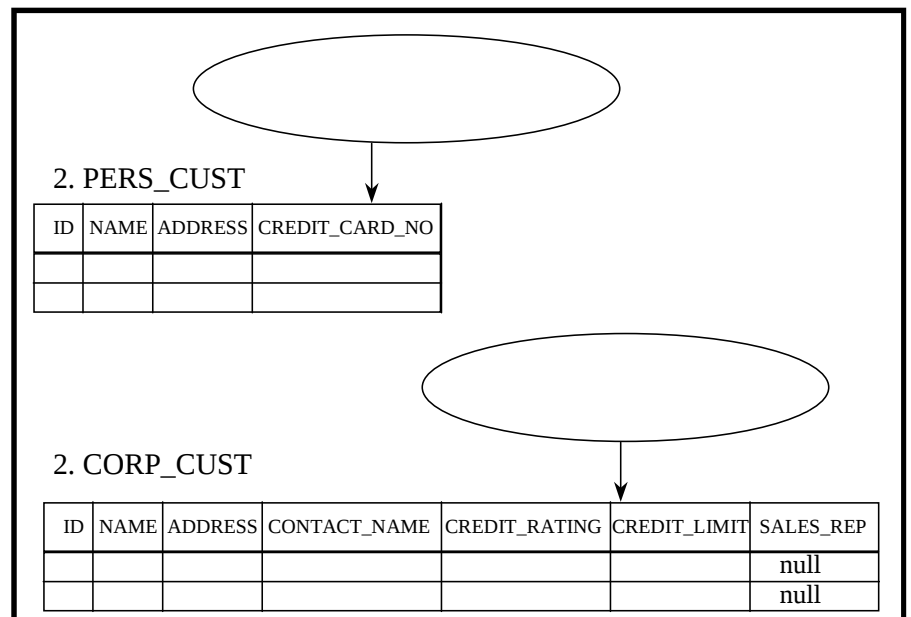
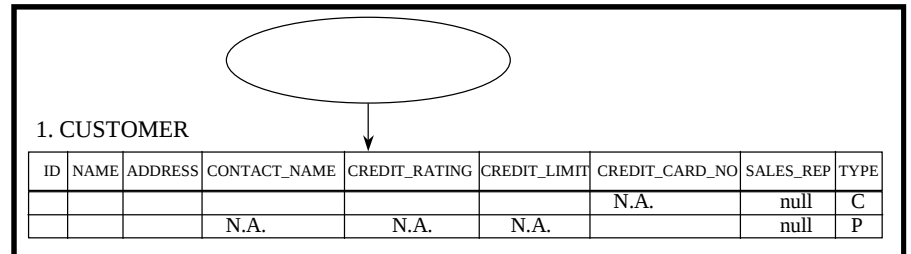
Marja Kärmeniemi,
Oracle Finland/Consulting

Rela-kerhon tämän kertaiseen teemaan, kuinka oliomallinnuksen tuloksena syntyneestä luokkakaaviosta suunnitellaan tehokas relaatiotietokanta, päätimme tarttua yksinkertaisen luokkakaavion avulla.



Tästä luokkakaaviosta voidaan periaatteessa toteuttaa relaatiotietokantaratkaisu, jossa on joko yksi, kaksi tai kolme taulua. N.A merkintä taulukuvissa tarkoittaa, että kyseinen solun arvo on rivillä merkityksetön. Relaatiotietokannassa se toteutetaan joko oletusarvolla (DEFAULT) tai tuntemattomalla arvolla (NULL). Kts. kuvt oikealla.

Päätimme laskea tietokantakutsujen suorittamiseen kuluva aika kuudella erilaisella käyttötapauskella näitä kolmea eri tauluratkaisua verraten. Kaikille kolmelle eri vaihtoehdolle löytyi joku käyttötapaus, jossa juuri

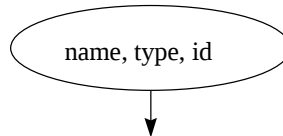


se oli paras tauluratkaisu. Yleispätevin oli ehdottomasti kuitenkin yhden taulun ratkaisu. Tässä artikkelissa on laskelmaesimerkit kahdesta käyttötapauksesta. Tehtävien asettelussa emme keskittyneet oikeaoppiseen oliokuvaamiseen.

Seuraavassa on joukko oletuksia, joita käytimme laskelmissa:

- levyn pyörimisnopus on 10 000 RPM => 6 ms/kierros
 - levyluvun pyörähdysviive on keskimäärin tästä puolet eli 3 ms
 - haku aika (seek) on 6 ms
 - levyjonoaika on 1 ms, kun levy-laite ei ole ylikuormittunut
=> haja-I/O on 3 ms + 6 ms + 1 ms = 10 ms
 - levylaitteen kuormitus saa olla 25%
=> 250 ms/10 ms = 25 eli levy-laite kykenee palvelemaan 25 DIO/s
=> jos tarvitaan 10 DIO/s, niin tapahtumia voi 5 tap/s
=> jos tarvitaan 20 DIO/s, niin tapahtumia voi olla 2,5 tap/s
(DIO on disk I/O eli I/O tapahtuma fyysiselle levylaitteelle)
 - tietokantasivun koko on 4 KB, peräkkäis-I/O aika on 0.35 ms/4KB
=> peräkkäislukunopeus on 0.1 ms/KB eli 10 MB/s
 - Tapahtuman ensimmäinen lukupyyntö indeksin lehtisivulle tai tauluun kohdistuu fyysiseen levyyn, tiedot säilyvät puskurissa tapahtuman loppuun.
 - Lajittelu aika keskusmuistissa on 0,01 ms/rivi
 - indeksin tilantarve on 2 kertaa avaimen pituus kertaa rivien lkm
- Tietomalliin ja käyttötapauksiin liittyviä oletuksia:
- Asiakkaita on 100 000, joista 20 % on yksityisasiakkaita ja 80% on yrityisasiakkaita

1B



1. CUSTOMER

ID	NAME	ADDRESS	CONTACT_NAME	CREDIT_RATING	CREDIT_LIMIT	CREDIT_CARD_NO	SALES_REP	TYPE
						null	null	C
			null	null	null		null	P

saantipolku index only

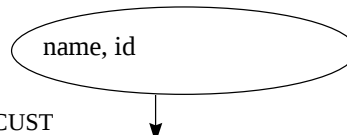
koko hakemiston läpiluku eli 1 haja-I/O ja peräkkäisluku

avain= 50 tavua => indeksin koko $2 \times 100\,000 \times 50 = 10\text{ MB}$

peräkkäislukuaika: $10\text{ MB} / 10\text{ MB/s} = 1\text{ s}$

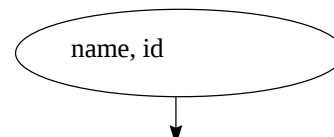
kokonaisaika: $1 \times 10\text{ ms} + 1\text{ s} = 1,01\text{ s}$ jos koko tulosjoukko luetaan kerralla
yhden näytön luku aika on 200 ms.

1B



2. PERS_CUST

ID	NAME	ADDRESS	CREDIT_CARD_NO



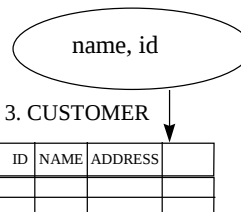
2. CORP_CUST

ID	NAME	ADDRESS	CONTACT_NAME	CREDIT_RATING	CREDIT_LIMIT	SALES_REP
						null
						null

index only molempiin tauluihin eli yhteensä 2 hajalukua + 10 MB peräkkäislukua
eli 1,02 s

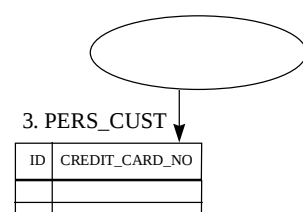
lisäksi tulosjoukon lajittu 0,01 ms per rivi eli $100 \times 0,01 = 1\text{ ms}$
yhteensä 1,021 s

1B



3. CUSTOMER

ID	NAME	ADDRESS



3. PERS_CUST

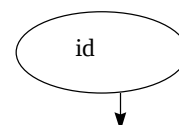
ID	CREDIT_CARD_NO

index only CUSTOMER tauluun, index only CORP_CUST tauluun (PERS_CUST taulua ei tarvita)
1 hajaluku + peräkkäisluku CUSTOMER tauluun indeksin
+ 100 hajalukua CORP_CUST tauluun indeksin

$10\text{ ms} + 1\text{ s} + 100 \times 10\text{ ms} = 2,01\text{ s}$,
näyttö kerrallaan $2,01\text{ s} / 5 = 400\text{ ms}$

3. CORP_CUST

ID	CONTACT_NAME	CREDIT_RATING	CREDIT_LIMIT	SALES_REP
				null
				null



- Tietokanta on optimaallisesti indeksoitu kyseisen käyttötapauksen mukaan
- Osoitetieto (address) on pilkottu useaan sarakkeeseen, jos siitä on hyötyä
- Kolmen taulun tapauksessa ei CUSTOMER tauluun ole suunniteltu TYPE-saraketta, vaan taulut on johdettu suoraan luokkakaaviosta

Ensimmäinen esimerkki:

Asiakkaiden haku nimen osalta siten, että haku-ehdossa on nimen osa. Tulosjoukon oletetaan olevan 100 riviä, jotka tulostetaan 5 osassa näytölle eli 20 riviä kerralla.

INPUT: like '%emi%'

OUTPUT: name, type, id order by name, type

Niemi	P	111
Nieminen	P	222
Nieminen Oy	C	333

Kts. kuvat edellisellä sivulla.

Toinen esimerkki:

Tarralistan tulostus postinumerojärjestyksessä. Osoitteen saa olettaa olevan tauluissa osissa eri erillinen ZIPCODE sarake on olemassa. Postia lähetetään kaikille asiakkaille. Yritysasikille tulostetaan tarralle myös yhteyshenkilön nimi, joka jää tyhjäksi yksityisasiakkaille.

OUTPUT: name, contact name, zipcode, address order by zipcode

Kts. kuvat oikealla.

Johtopäätökset näiden esimerkkien läpilaskemisesta ovat luettavissa muiden Rela-kerholaisten kirjoittamista artikkeleista.

Marja Kärmeniemi,
Oracle Finland/Consulting

2

zipcode, name, contact_name, address

1. CUSTOMER

ID	NAME	ADDRESS	CONTACT_NAME	CREDIT_RATING	CREDIT_LIMIT	CREDIT_CARD_NO	SALES_REP	TYPE
						null	null	C
			null	null	null		null	P

indeksin läpiluku

avain 150 tavua => $2 \times 150 \times 100\,000 = 30\text{ MB}$

$30\text{ MB}/10\text{ MB/s} = 3\text{ s}$

2

zipcode, name, address

2. PERS_CUST

ID	NAME	ADDRESS	CREDIT_CARD_NO

zipcode, name, contact_name, address

2. CORP_CUST

ID	NAME	ADDRESS	CONTACT_NAME	CREDIT_RATING	CREDIT_LIMIT	SALES_REP
						null
						null

molemmat indeksit luetaan läpi $3\text{ s} + \text{lajittelu } 1\text{ s} = 4\text{ s}$

2

zipcode, name, address, id

3. CUSTOMER

ID	NAME	ADDRESS

3. PERS_CUST

ID	CREDIT_CARD_NO

CUSTOMER taulun indeksin läpiluku + hajaluku per rivi CORP_CUST taulun indeksiin

avain 100 tavua => $2\text{ s} + 100\,000 \times 10\text{ ms} = 1002\text{ s}$ eli 16 min

(pessimistinen arvio, sillä sivut luetaan vain kerran, jos riittävä cache eli 1 hajaluku per sivu. Indeksien koko $2 \times 80\,000 \times 50\text{ tavua} = 8\text{ MB}$, jos sivu on 4 KB => $2000\text{ sivua} \times 10\text{ ms} = 20\text{ s}$)

id, contact_name

3. CORP_CUST

ID	CONTACT_NAME	CREDIT_RATING	CREDIT_LIMIT	SALES_REP
				null
				null

fiksun optimoijan tapa: CUSTOMER taulun indeksi lajitellaan id:n mukaan

(2 s läpiluku + lajittelu 1 s), merge scan join toiseen indeksiin, jonka läpiluku $0,8\text{ s}$, lopputulos lajitellaan 1 s => yhteensä $3\text{ s} + 0,8\text{ s} + 1\text{ s} = 4,8\text{ s}$

Käytännön kokeilu Rational Rosella

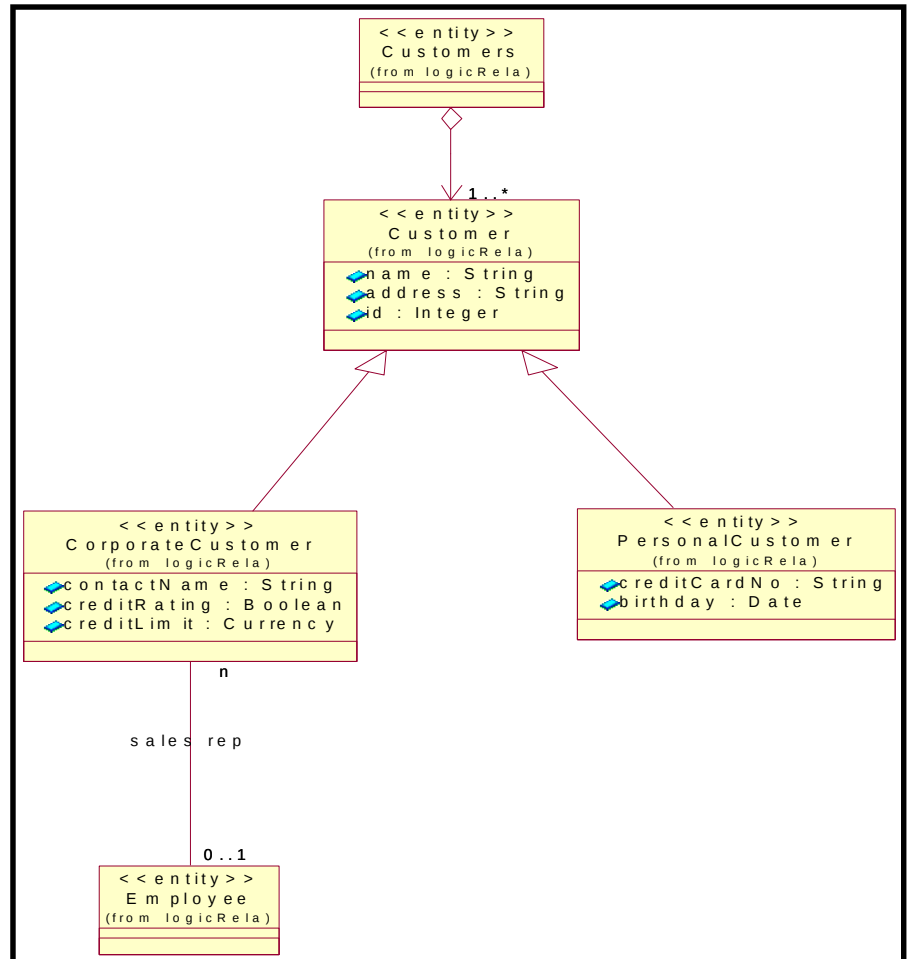
Tuovi Honkanen,
FinanssiData

1 Tausta

Teorian lisäksi halusimme todentaa, että miten oliot muuntautuvat käytännössäkin. Mallinnuksen kohteena on Asiakas. Tehtävästä rajasimme ulos mallissa olevat Tilaustiedot. Ko-keiluvälineeksi valikoitui Rational Rose, koska semmoinen löytyi asennettuna koneesta.

2 Luokkamalli

Ensin tehtiin lähtömallin loogisen mallin mukainen luokkakaavio (class diagram), joka on tietenkin identtinen kirjassa olleen Asiakas-luokan kanssa. Tätä edeltävät kuvaukset jätettiin kuvaamatta (esim käyttötapakaavio (use case diagram)). Kts. kuva oikealla.



3 Tietokantakaavion ja luontilauseiden luonti oletusarvoin

Loogisen mallin perusteella luotiin fyysinen tietokantakaavio.

Jos tarpeellisen aivotyön jättää väliin, tarjoaa ohjelmisto tietokantamielessä tehottoman ratkaisun.

Kts. kuva oikealla yllä.

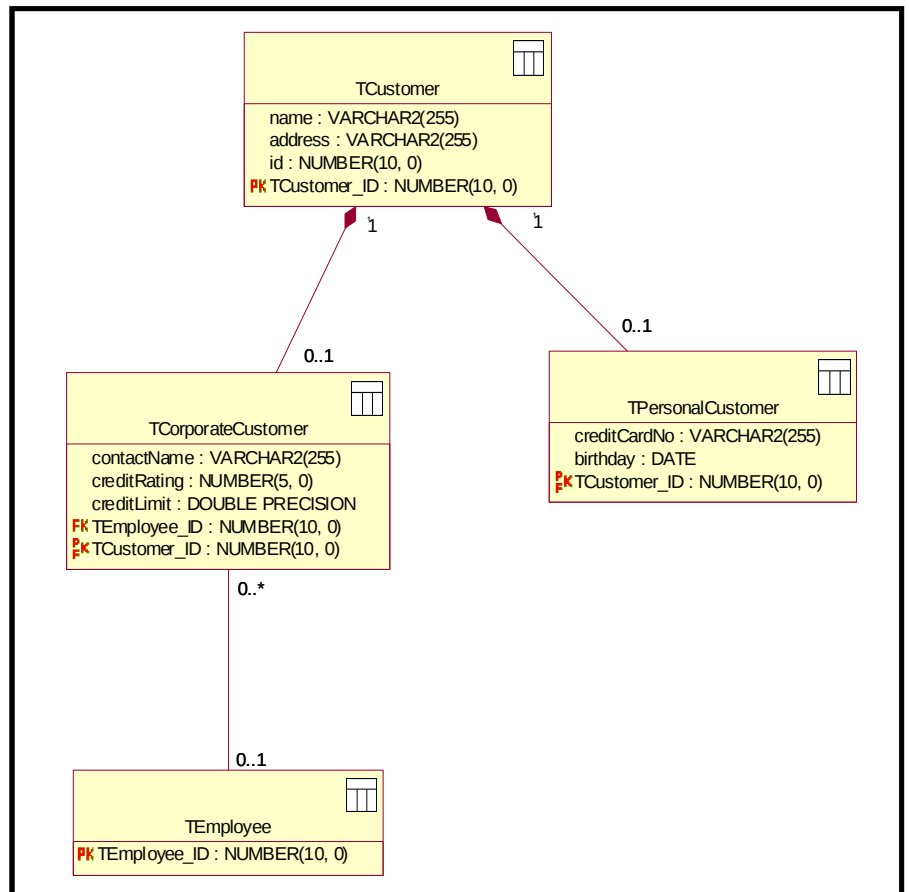
Em. tietokantakaavion perusteella luotiin taulujen luontilauseet.

Kts. kuva oikealla alla.

4 Muutokset tietokantakaavioon

Luokkakaavio kannattaa jättää rauhaan ja muutokset tehdä tietokantakaavioon.

*Tuovi Honkanen,
FinanssiData*



```
CREATE TABLE RELA.PersonalCustomer (
  creditCardNo VARCHAR ( 255 ) NOT NULL,
  birthday DATE NOT NULL,
  id INTEGER NOT NULL,
  PRIMARY KEY (id)
);
CREATE TABLE RELA.Employee (
  Employee_ID INTEGER NOT NULL,
  PRIMARY KEY (Employee_ID)
);
CREATE TABLE RELA.Customer (
  name VARCHAR ( 255 ) NOT NULL,
  address VARCHAR ( 255 ) NOT NULL,
  id INTEGER NOT NULL,
  PRIMARY KEY (id)
);
CREATE TABLE RELA.CorporateCustomer (
  contactName VARCHAR ( 255 ) NOT NULL,
  creditRating SMALLINT NOT NULL,
  creditLimit DOUBLE NOT NULL,
  Employee_ID INTEGER,
  id INTEGER NOT NULL,
  PRIMARY KEY (id)
);
CREATE UNIQUE INDEX PK_Pe40 ON RELA.PersonalCustomer (id);
CREATE UNIQUE INDEX PK_Em38 ON RELA.Employee (Employee_ID);
CREATE UNIQUE INDEX PK_Cu37 ON RELA.Customer (id);
CREATE UNIQUE INDEX PK_Co39 ON RELA.CorporateCustomer (id);
CREATE INDEX RELA.TC_Co81 ON RELA.CorporateCustomer (Employee_ID) PCTFREE 0;
ALTER TABLE RELA.PersonalCustomer ADD FOREIGN KEY FK_Pe17 (id) REFERENCES RELA.Customer (id) ON DELETE NO ACTION;
ALTER TABLE RELA.CorporateCustomer ADD FOREIGN KEY FK_Co16 (id) REFERENCES RELA.Customer (id) ON DELETE NO ACTION;
ALTER TABLE RELA.CorporateCustomer ADD FOREIGN KEY FK_Co15 (Employee_ID) REFERENCES RELA.Employee (Employee_ID) ON DELETE NO ACTION;
```

Indeksien suunnittelu hallinnassa, taulujen suunnittelu ei ihan vielä

*Tapio Lahdenmäki,
IBM*

Parikymmentä vuotta sitä harjoiteltiin, mutta nyt osataan suunnitella relaatiokantojen indeksejä. Algoritmit osoittautuivat lopulta yllättävän yksinkertaisiksi, kun levy- ja keskusmuistin halpeneminen vähensi tärkeiden muuttujien määrää. Operatiivisten kantojen indeksoinnissa on kyse yleensä lähinnä hajalevylukujen määrän säätelystä: indeksin paksuntaminen hidastaa hieman lisäsarakeiden päivitystä mutta nopeuttaa dramaattisesti kyselyjä, joiden tarvitsemat sarakkeet ovat paksussa indeksissä.

Tietokanta-avusteinen tietokannan suunnittelu on jo edennyt indeksien suunnittelusta summataulujen suunnitteluun. Perustaulujen suunnittelussa ollaan kuitenkin vielä taiteen tasolla.

**Puhdasoppisia tauluratkaisuja
on useita**

Kun kohdemallissa on ehdollinen suhde (1:C, esimerkiksi henkilö:ajokortti), on tarjolla kaksi ratkaisuvaihtoehtoa: kaksi taulua tai yksi paksu taulu, jossa on paljon tyhjiä ruu-

tuja. Kumpikin on oikeaoppinen. Arvon puuttuminen ei ole denormalisointirike. Valinta on tehokkuuskysymys: paksu taulu vie yleensä enemmän levytilaa, mutta johtaa pienempään hajalevylukujen määrään.

Hajalevylukujen määrä on tärkeä tehokkuustekijä ellei koko tietokanta ole keskusmuistissa.

Nopeimmillakin nykylaitteilla yhden sivun hajaluku pyöriivältä levyltä vie noin 10 ms.

Paksun taulun vaihtoehdossa koko rivi on normaalisti yhdellä sivulla, kahden taulun ratkaisussa kahdella sivulla. Tämä on ydinkysymys tehokkuuden kannalta.

Käytännössä tehokkuusero on usein suurempi kuin 1:2. Kun SELECTissä viitataan useaan tauluun, optimoijaongelmien riski kasvaa. Joskus lajitellaan tarpeettomasti, joskus taulujen käsittelyjärjestys on väärä. Näitä tehokkuusongelmia ei aina saada hoidetuksi parhaillakaan indekseillä.

Näyttää siltä, että usean taulun ratkaisu valitaan usein joko täysin vaistomaisesti tai sen vuoksi, että tyh-

jät tauluruudut vaikuttavat holtittomalta levytilan tuhlaukselta. Ennen pilkkomisratkaisua kannattaa muuttaa säästö markoiksi esimerkiksi kaavalla 100 euroa per gigatavu kuukaudessa. Tämä laskelma yleensä todistaa levytilasäästön merkityksettömäksi, vaikka tiivistys jätetään huomiotta. Monet tietokantajärjestelmät tiivistävät tehokkaasti taulussa yleisiä merkkijonoja - esimerkiksi toistuvalla 50 blankon jonolle haetaan varmaan lyhyt vastine.

Todistuksen taakka pilkkojalle

Taulujen suunnittelijana lähtisin aina liikkeelle ratkaisusta, jossa yhteydet 1:C ja 1:1 on toteutettu paksulla taululla. Olisin kyllä valmis uskomaan, että joissakin erikoistapauksissa usean taulun ratkaisu on tehokkaampi - jos se numeerisesti todistetaan. Ymmärtäisin myös, että sivun koko ja sarakkeiden maksimimäärä pakottavat ääritapauksissa taulun pilkkomiseen. Voisin ymmärtää sarakkeiden nimeämisongelmiakin. Muuten olisin itsepäinen.

*Tapio Lahdenmäki,
IBM*

Käytännön läheinen syysseminaari

Testaus ja laatu

*Veli Matti Nokso-Koivisto,
SysOpen Oyj*

Sytykeen perinteinen syysseminaari pidettiin 3.9.-5.9.2002. Seminaarin aiheena oli ajankohtainen teema Testaus ja laatu. Testauksen ja laadun ongelmiin ja ratkaisuihin perehtyi 75 tietotekniikan ammattilaista perinteiseksi käyneessä ympäristössä Helsingin ja Tukholman välillä liikennöivällä Silja Serenadella. Ammatillinen osio sisälsi kahdeksan korkeatasoista esitelmää, jotka valottivat teemaa sekä tutkimuksen että käytännön näkökulmista.

Ammatillisen puurtamisen lomassa seminaariväki muisti hyödyntää myös laivan ravintoloiden ruokatarjontaa ja laajentaa henkilökohtaista suhdeverkostoaan tutustumalla muihin alan ammattilaisiin. Seminaarin osallistujia väitetään myös nähdyin jopa laivan yökerhon tanssiparketilla, joten rentoutumisfunktiokin lienee tullut hoidetuksi.



Esitelmäsarjan avasi Compuwaren Paavo Häkkinen esitelmöimällä Suorituskyvyn varmistamisesta sovel-luskehityksen eri vaiheissa. Hän painotti suorituskyvyvaatimusten huomi-oimista kaikissa suunnittelun vaiheis-sa ja kertoi, että 75 % niistä Web-jär-jestelmien kehittäjistä, jotka eivät teh-neet suorituskykytestejä suunnittelun aikana, kokivat skaalautuvuus- ja suo-rituskykyongelmia tuotantovaiheessa. Paavo korosti arkkitehtuurin suoritus-kyvyn arvioimista.



MERCURY INTERACTIVE

Mercuri Interactiven Mika Ger-lander alusti aiheesta QA prosessin hallinnan ja automaattisen testauksen merkityksestä liiketoimintariskinen minimoimisessa uusia palveluita kehi-

tettäessä. Hän painotti riskienhallinnan ja laaduvarmistuksen ajattelun jatku-vuutta. Alustuksen jälkeen Jarkko Augustin ja Christian Jurvanen Pro Agriasta kertoivat maatalouden las-kentajärjestelmien siirrosta Windows 2000 alustalle. He kertoivat automaa-tisoineensa oman järjestelmänsä tes-tauksesta 85 – 90 %. Automaattisen testauksen etuja ovat toistettavuus ja se, että testauksen suorittaminen voi-daan helposti siirtää toiselle henkilöl-le. Kattavien testien tekeminen on no-peampaa työkalun avulla. Ongelma-na ovat alussa testausohjelmistojen te-kemiseen kuluva aika ja sopivan tes-tidatan löytäminen. Yhteenvetona he totesivat, että testauksen automati-sointi on kannattavaa sen ongelmalli-suudesta huolimatta.

Professori Jukka Paakki Helsin-gin yliopiston tietojenkäsittelytieteen laitokselta piti kiinnostavan esitelmän aiheenaan Testaus teoriassa ja käy-tännössä. Hän oli tehnyt kattavan kir-jallisuustutkimuksen testauksesta jul-kaisuista tieteellisistä artikkeleista. Testausta tutkitaan paljon ja siitä on kirjoitettu runsaasti. Testauksen teo-rian tutkimuksen ja teollisuuden käyt-tämien menetelmien välillä on kuiten-kin merkittävä ero. Käytännön ongel-mat ovat liian lyhytnäköisiä, suppeita, teknispainotteisia ja vain tiettyyn jär-jestelmään sidottuja kelvataksaan tut-kijoille tieteellisiksi tutkimuskohteiksi. Tutkijan näkökulmasta käytännön on-gelmat ovat melko tylsiä. Teorian saa-minen teollisuuden hyötykäyttöön kes-tää vähintään 20 vuotta eikä hyödyis-tä ole sitä ennen täyttä varmuutta. Yliopiston tutkija kokee tehtäväksen uusien ideoitten ja uusien tieteellisten tulosten synnyttämisen. Empiirinen tutkimus yhdistää teorian ja käytän-nön. Empiirisellä tutkimuksella selvi-tetään, miten tietty prosessi, menetel-



mä tai työkalu toimii käytännön projekteissa. Empiirinen tutkimus on osoittanut, että mitä laajempi projekti, sitä enemmän siinä on myös virheitä. Projektin laajuus selittää virheitten määrää enemmän kuin sen monimutkaisuus. Yritysten väki oli päättynyt samansuuntaiseen johtopäätökseen kantapään kautta.

CONFORMIQ

Conformiqin Antti Laine herätteli kuulijoitaan katsomaan testausta toisella tavoin. Hän totesi, että elektronikan testausmenetelmät ovat kymmenen vuotta edellä ohjelmistoteollisuutta. Elektroniikkateollisuudessa formaalit menetelmät ovat kehittyneet integroitujen piirien pakkaustiheyden kasvun myötä. Ohjelmistoteollisuudessa ero testausvaatimusten ja käytettävissä olevien työkalujen välillä kasvaa koko ajan. Antti painotti malliin perustuvan testigeneroinnin etuja. Niitä ovat päivityksen helppous, testausmenetelmien itsedokumentoituus, testien parempi kattavuus sekä mahdollisuus korvata massiiviset skriptit mallilla, josta testausmateriaali generoituu automaattisesti. Antti lupaili automaattisella testimateriaalin generoinnilla jopa kymmenkertaista tuottavuuden lisäystä perinteisiin menetelmiin verrattuna vastaavan luotettavuustason saavuttamisessa.

TietoEnator ^{TE}

Anna-Liisa Sihvonen TietoEnatorilta puhui Testitapausten suunnittelusta. Hän esitteli ortogonaalimatriisin, jota hyödynnetään tieteellisessä kokeensuunnittelussa. Sen avulla löydetään pienimmällä mahdollisella koeäärällä kaikki riipuvuudet. Ortogonaalimatriisia voidaan hyödyntää myös ohjelmistotestauksessa testitapausten suunnittelussa. Sillä päästään ad hoc-testausta kattavampaan testaukseen. Anna-Liisa teetti seminaarin osallistujilla mielenkiintoisen

harjoituksen ReissuRepen matkatointon varausjärjestelmän testausmateriaalin rakentamisesta. Seminaarin osallistujat pätkailivät testitapauksia varausjärjestelmälle, joka välittää viikon ja kahden matkoja Timbaktuun ja Ouagadougouhun seikkailujen tai nähtävyyksien merkeissä. Anna-Liisa pelkisti ketterästi kahdeksan testitapausta neljäksi. Seminaariväelle jäi vielä epäselväksi, selviäisikö ReissuRepen varausjärjestelmä sittenkään Sytykeen syysseminaarin varauksista.



Tieturin Ilkka Myllylä esitteli tapoja suorituskyvyn pullonkaulojen löytämiseksi ja suorituskyvyn optimointiin. Pullonkaula voi olla mikä tahansa resurssi, kuten ohjelmisto, laite tai tietoverkon kaistaleveys, joka rajoittaa sovelluksen suoritussopeutta. Reseptinä Ilkka esitteli suorituskyvyn optimoimista riittävän aikaisessa vaiheessa, parhaiten jo määrittelyssä. Keskeisiä pullonkaulojen sijaintipaikkoja web-sovellutuksissa ovat sovelluspalvelin, tietokantapalvelin sekä verkko. Pullonkaula vääristää helposti muiden resurssien kuormitusta ja voi peittää alleen muita pullonkauloja. Profilointityökalut auttavat pullonkaulojen hakemisessa.

Erkki Pöyhönen Nokialta puhui testauksen kohdistamisesta. Hän kertoi testausstrategian tavoitteista, priorisoinnista ja strategiasta. Erkki painotti projektin laatutavoitteitten ymmärtämistä ja niiden mukaista testausstrategian suunnittelua. On löydettävä testit, joilla voidaan varmistaa, että tuote täyttää sen tärkeimmät laatutavoitteet. Tuotteen rakenne on ymmärrettävä, jotta nähdään, mikä osio vaikuttaa mihinkin laatuatribuuttiin. Riskianalyysiin perustuvalla priorisoin-

nilla voidaan löytää keskeisimmät testauskohteet. Testaustarpeet on määriteltävä businessstarpeista lähtien. Etu-riski-matriisin avulla pysytään koko ajan selvillä, mitkä tuotteen edut voidaan varmistaa kullakin testausvaihtoehdolla.

Ennakko-ohjelmassa mainittu Rationalin Olli Mension esitys Java-ohjelmistojen testaamisesta jäi valitettavasti pois ohjelmasta sairastumisen takia.



Sytykeen puheenjohtaja Pekka Forselius STTF:stä puhui aiheenaan Ohjelmiston laadun ja laatuvaatimusten mittaaminen. Ohjelmiston hyvällä ladulla tarkoitetaan sen kykyä vastata odotettuihin laatuvaatimuksiin. Testauksen tarkoitus on todentaa kohteen vaatimustenmukaisuus sekä toiminnallisten että laadullisten vaatimusten osalta. Pekka esitteli myös ohjelmistotuotteen laatustandardin ISO9125, jonka mukaisesti ohjelmiston laatu koostuu kuudesta osa-alueesta: toiminnallinen laatu, luotettavuus, käytettävyys, tuotannollinen tehokkuus, ylläpidettävyys ja siirrettävyys. Pekka totesi, että absoluuttisen laadun mittaaminen lähes mahdotonta. Laatu on käyttäjä- ja käyttötilannekohtainen asia. Pekka perusteli laatustandardien käyttöä sillä, että asioiden systemaattinen tarkastelu ja standardien hyödyntäminen vähentävät riitelemistä ja siten lisäävät onnellisuutta.

Ammatillisen osuuden tultua täyteen seminaariväki poistui saunaan entistä kyvykkäämpä tuottamaan onnea sekä asiakkailleen että itselleen.

Suurta kiitosta sai Heini Holopaisesta koostuneen yhden hengen ris-teilytoimikunnan aktiivinen työskentely

onnistuneen seminaarin järjestämisessä. Osanottajat olivat näkemäänsä ja kuulemaansa hyvin tyytyväisiä ja tunsivat saaneensa uusia eväitä päivittäisen työhönsä. Keskustelua herätti eniten seminaarin ajankäytön organisointi. Tukholman osuuteen kaivattiin ammatillista sisältöä esimerkiksi esitelmien tai yritysvierailujen muodossa.

Seminaarissa kulki kuuma huhu Sytykeen entisen puheenjohtajan, Siljan, pimeistä asehankinnoista Tukholman vanhan kaupungin hämärillä kujilla. Huhu sai Sytykeen johtokunnan pelkäämään entisen puheenjohtajan suunnittelevan aseellista vallankumousta. Johtokunnan lähettämä tiedustelija löysi Siljan matkatavaroista kypäroitä ja miekkoja. Materiaali muistutti harmaata terästä, mutta kysymyksessä saattoi olla jopa aito muovi. Silja yritti selittää asiaa pienten sukulaispoikien mielihaluilla, mutta peitetarina ei vakuuttanut johtokuntaa, joka jäi suunnittelemaan kilpavarustelukierrosta seuraavan seminaarin yhteydessä.

Muistiin merkitä Systeemityö-lehden risteilevä reportteri,

Veli Matti Nokso-Koivisto, SysOpen Oyj

PS. Osa esitelmien materiaalista on ladattavissa kiinnostuneiden käyttöön Sytykkeen verkkosivuilta.

Esitelmäitsijät:

Paavo Häkkinen,
Compuware,
paavo.hakkinen@compuware.com

Mika Gerlander,
Mercury Interactive,
m.gerlander@mercury-eur.com

Jarkko Augustin ja
Christian Jurvanen,
Pro Agria,
jarkko.augustin@mloy.fi ja
christian.jurvanen@mloy.fi

Jukka Paakki,
Helsingin Yliopisto,
jukka.paakki@helsinki.fi

Antti Laine,
Conformiq,
antti.laine@conformic.com

Anna-Liisa Sihvonon,
TietoEnator Oyj
anna-liisa.sihvonon@tietoenator.com

Ilkka Myllylä,
Tieturi Oy,
ilkka.myllyla@tieturi.fi

Erkki Pöyhönen,
Nokia Research Center,
erkki.poyhonen@nokia.com

Pekka Forselius,
STTF Oy,
pekkaf@sttf.fi

