

Väitöskirja: Joustavampia menetelmiä systeemityöhön

Juha-Pekka Tolvanen,
MetaCase Consulting Oy

Tietojärjestelmien suunnittelun ja rakentamisen tueksi on kehitetty erilaisia systeemityön menetelmiä. Kuten muillakin insinöörialoilla menetelmillä pyritään ohjaamaan ja systematisoimaan kehitystyötä sen tuottavuuden ja laadun parantamiseksi. Nämä tavoitteet ovat systeemityössä niin toivottuja että erilaisia menetelmiä ja niiden laajennoksia on kehitetty useita satoja. Esimerkiksi vasta 90-luvulla yleistyneitä olio-menetelmiä on arvioitu olevan jo yli sata erilaista ja niitä kehitetään yhä lisää. Menetelmiä ja systeemityömalleja on myös pyritty standardisoimaan (esim. IDEF¹, SSADM², UML³) mutta tulokset ovat toistaiseksi olleet varsin heikot. Esimerkiksi viime vuosina standardoinnin tuloksena syntynyttä UML-menetelmää on jo käytössä kolmea erilaista versiota ja uusia laajennoksia esitetään lähes jokaisessa UML-menetelmää käsittelevässä konferenssissa.

Menetelmien käytöstä

Menetelmien käytön osalta vallitsee kuitenkin paradoksaalinen tilanne: Vaikka menetelmiä on kehitetty lukuisia ja niillä saavutettavat hyödyt tunnustetaan menetelmiä käytetään yllättävän vähän. Viimeaikaiset empiiriset tutkimukset⁴ ovat osoittaneet että alle puolessa kaikista systeemityöhankkeista ylipäänsä käytetään menetelmiä. Arvioidaanko yrityksissä menetelmien hyödyt haittoja pienemmiksi? Ainakin menetelmät koetaan tutkimusten mukaan systeemityötä rajoittaviksi ja hidastaviksi. "Ei päästä heti ohjelmoimaan kun pitää ensin määritellä", on varsin tavallinen kommentti menetelmien haitoista. Muita syitä menetelmien vähäiseen käyttöön

ovat niiden oppimisen vaikeus, vaadittavan alkuinvestoinnin suuruus, sekä menetelmien sopimattomuus yrityksen tarpeisiin.

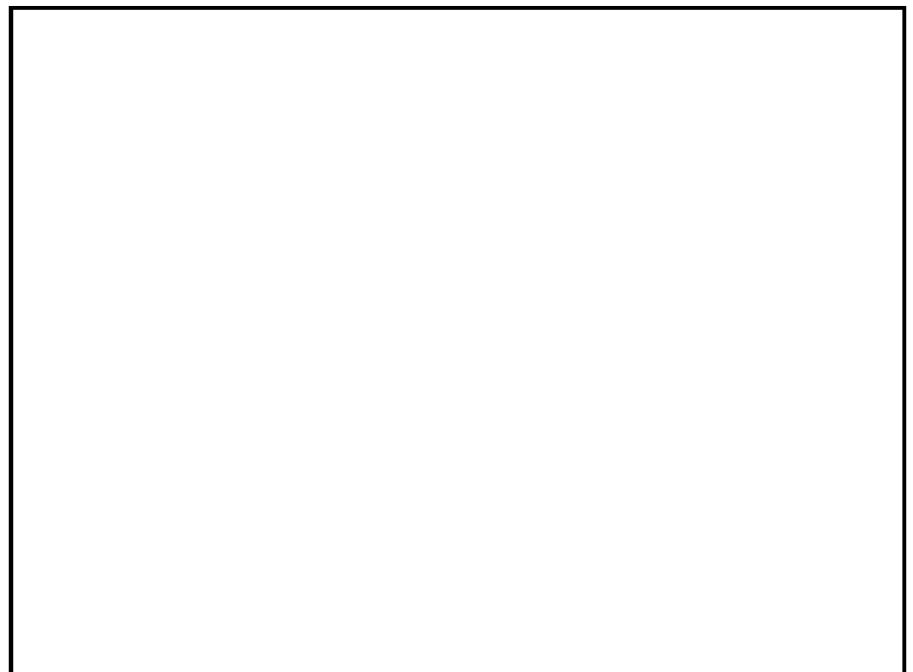
Menetelmien tilannekohtaisuus

Toinen systeemityön menetelmien erikoisuus on niiden tilannekohtaisuus. Erilaisiin kehitysympäristöihin ja erilaisten tietojärjestelmien suunnitteluun soveltuvat erityyppiset menetelmät: esimerkiksi matkapuhelimien suunnittelussa menetelmätarpeet ovat erilaisia kuin suunniteltaessa www-sovelluksia tai logistisia prosesseja tukevia tietojärjestelmiä. Tilannekohtaisten menetelmien tarpeellisuutta korostavat lisäksi uusien tietojärjestelmätyyppien ja käytettävissä olevien teknisten ratkaisuiden lisääntyminen. Empiiristen tutkimusten mukaan onkin varsin tavallista että menetelmiä käyttävät organisaatiot ja yksittäiset kehitysprojektit muokkaavat menetelmiä omia käyttötilanteitaan varten^{5,6}. Yksi käytännön esimerkki Suomesta: Nokia Telecommunications va-

litsi 90-luvun alussa OMT-menetelmän GSM-verkonhallintajärjestelmän kehityshankkeisiin. Varsin nopeasti kuitenkin havaittiin menetelmän puutteet esimerkiksi laajojen järjestelmien, interaktiivisuuden ja käyttöliittymien kehittämisessä. Tämä johti useisiin menetelmämuutoksiin ja -versioihin (OMT+, OMT++) joilla tavoiteltiin parempia työtapoja ja menetelmiä. Vastaavanlaisia paikallisia menetelmäkehityshankkeita on satoja, joista Ari Jaaksin väitöskirja⁷ antaa hyvän kuvauksen OMT++:n osalta.

Menetelmätutkimus

Menetelmien verrattain vähäinen käyttö sekä niiden tilannekohtaisuus olivat pääsyyinä väitöskirjatyölleni. Voimakkaasti kärjistäen voisi sanoa, että valtaosa yrityksistä ei käytä menetelmiä ja nekin jotka käyttävät muokkaavat niitä omiin tarkoituksiinsa. Kaikista menetelmästandardeista (esim. UML) huolimatta suuntaus yrityksissä on universaalien



Kuva 1. Esimerkki menetelmän tukemisesta CASE-välineessä.

menetelmien sijasta kohti tilannekohtaisia menetelmiä. Vaikka mallien symbolit voivat näyttää samanlaisilta eri yrityksissä niillä kuvataan ja tarkastellaan erilaisia asioita eri tarkkuus-tasoilla. Tästä kehityksestä OMT + on yksi hyvä esimerkki.

Tutkimuksen kannalta on tällöin mielenkiintoista sen miten yritysten suorittamaa menetelmäkehitystä voidaan tukea. Väitöskirjatyössäni olen tarkastellut tätä ongelmaa toisaalta menetelmien kuvaamisen ja toisaalta organisaation oppimisen kannalta. Menetelmiä tarkastellaan osana tietokoneavusteisia suunnitteluohjelmistoja, joita ovat esimerkiksi CASE-välineet (Computer Aided Systems Engineering) ja liiketoimintaprosessien suunnitteluohjelmistot. Suunnitteluohjelmistojen tarkastelu menetelmien yhteydessä on tärkeää koska pelkän kynän ja paperin avulla ei laajoja kehityshankkeista voida toteuttaa. Suunnitteluohjelmistojen ansioista tietojärjestelmän määrittämisestä voidaan laatia, jakaa ja ylläpitää tehokkaasti sekä määrittämisestä voidaan generoida esimerkiksi prototyyppisiä, koodia ja testiaineistoa. Kuvassa 1 esitetään esimerkki muutamasta CASE-välineen toiminnallisuudesta: luokkakaavion laatimisesta sekä CORBA IDL:n mukaisen rajapintakuvauksen generoinnista.

Perinteiset välineet kuitenkin rajoittavat tukemaan vain yhtä menetelmää ja määrättyä ohjelmointikieltä. Tällöin yksi menetelmä ja väline on hyvä sulatettujen järjestelmien suunnittelussa, toinen www-sovellusten suunnittelussa ja kolmas vaikka CORBA IDL-rajapintojen suunnittelussa ja niiden generoinnissa. Erityisesti on huomattava, että mikäli yrityksellä on jotain omia tarpeita CASE-väline ei pysty tukemaan niitä.

Ideaaliratkaisu välineiden ja menetelmien joustamattomuuteen olisi se, että käyttäjät saavat itse määrittää heidän työtapoihin ja sovellusalueille sopivat työkalut. MetaCASE-välineet automatisoivat juuri tämän prosessin. Ne tarjoavat välineet, joilla voidaan

kuvata systeemyön menetelmä ja generoida menetelmäkuvausten perusteella CASE-väline. MetaCASE-väline on siis eräänlainen CASE-kehitin (vertaa metakääntäjä). Menetelmämäärittäykset voivat sisältää systeemyössä käytettävät käsitteet, säännöt, notaation, koodin generaattorit ja dokumentointikäytännöt.

Menetelmien kuvaaminen

Menetelmien kuten minkä tahansa muunkin järjestelmän kehittämisessä määrittelytavat ovat keskeisessä asemassa. Menetelmän käsitteet, säännöt, ohjeistukset, notaatio jne, pitää pystyä kuvaamaan mahdollimman täydellisesti. Menetelmän määrittämisestä käytetään yleisesti termiä metamalli, eli malli mallista. Menetelmien mallinnuksesta käytetään puolestaan termiä metamallinnus. Täydellisyyden lisäksi metamallien pitää olla riittävän formaaleja CASE-työkalujen generoimiseksi sekä metamallintamisen kustannustehokasta: oman menetelmän ja CASE-työkalun tekemisen pitää olla mahdollisimman helppoa ja nopeaa. Esimerkiksi kuvassa 1 esitetyn luokkamallin ja sitä tukevan välineen määrittely ja toteutus ei pitäisi kestää muutamaa tuntia kauempaa.

Väitöskirjatyössä esitetään käsitteitä

menetelmien kuvaamiseksi, joita voidaan käyttää siten myös metamallinnuskielten ominaisuuksina. Tarpeellisia (meta)käsitteitä etsitään induktiivisesti analysoimalla joukkoa suunnittelumenetelmiä, kuvaamalla ne metamallinnuskielten avulla ja sovittamalla menetelmät muokattaviin CASE-välineisiin. Rakennettujen CASE-välineiden avulla voidaan testata metamallien oikeellisuus. Kuvassa 2 esitetään esimerkki UML-menetelmän 'luokan' määrittämisestä MetaEdit + metaCASE-välineessä. Määrittämisessä luokalla on joukko ominaisuuksia kuten luokan nimi ja sen operaatiot, sääntöjä kuten mahdollinen yksiperintä sekä symbolimäärittäminen. Näiden määrittäysten kautta on generoitu CASE-väline jonka käyttöä esitetään kuvassa 1.

Menetelmäkehityksen inkrementaalisuus

Menetelmien kehittämisen lisäksi on tärkeää arvioida ovatko ne toimineet käytännössä toivotulla tavalla. Mikäli perinteisellä CASE-välineellä tuettu menetelmä ei sovellu yrityksen tarpeisiin sen käyttö lopetetaan tai käyttöä jatketaan puutteista huolimatta. MetaCASE-teknologiaa käyttävällä yrityksellä on vaihtoehtona myös menetelmän jatkokehitys. Menetelmätietämyksen ylläpitoa on väitöskirjassa



Kuva 2. Esimerkki menetelmän määrittämisestä metaCASE-välineessä.

tarkasteltu organisaation oppimisen kannalta: menetelmät kertovat miten tietojärjestelmiä pitäisi rakentaa ja ovat näin ollen jokaisen ohjelmistotalon tarkeinta omaisuutta.

Väitöskirjatyössä on esitetty periaatteita menetelmien soveltuvuuden arvioimiseksi. Esitetyt periaatteet perustuvat tietojärjestelmiä kuvaavien mallien ja menetelmiä kuvaavien metamallien väliseen vertailuun. Mallit kuvaavat miten menetelmää on käytetty ja metamallit miten menetelmää oli kehittäjän mielestä tarkoitus käyttää. Kyse on siis tarkoituksen ja sen toteutumisen vertaamisesta. Vertaamisen tuloksena syntyy joukko muutosvaatimuksia joilla voidaan osoittaa juuri määrättyihin kohtiin menetelmän määrityksissä, siis metamallissa. Metamallia muuttamalla voidaan menetelmää parantaa ja "päivitetty" menetelmä on metaCASE-työkalun kautta jaettavissa edelleen kaikille systeemityöhön osallistuville. Tämä päättymätön ketju menetelmän käytöstä, arvioinnista ja kehittämisestä on nimetty työssä inkrementaaliseksi menetelmäkehitykseksi.

Inkrementaalisen lähestymistavan käyttökelpoisuutta on testattu useissa käytännön hankkeissa. Väitöskirjatyössä havainnollistetaan nämä periaatteet kuvaamalla niiden käyttöä kahdessa tietojärjestelmien kehityshankkeessa: tukkukaupan ja metsäteollisuuden logististen tietojärjestelmien suunnittelussa. Molemmissa tapaus-tutkimuksissa käytettyjen suunnittelu-menetelmien soveltuvuutta pystyttiin parantamaan työssä esitettyjen periaatteiden avulla.

Tulosten hyödynnettävyys

Väitöskirjatyön tuloksia voivat hyödyntää kaikki suunnittelumenetelmiä käyttävät organisaatiot menetelmien soveltuvuuden parantamiseksi ja menetelmäosaamisen kehittämiseksi. Tulosten hyödynnettävyyttä voi kuvata ehkä parhaiten toteamalla että kehitettyjä menetelmiä että metamallinnuksen periaatteita sovelletaan jo osana MetaEdit+ -ohjelmistoa yli 30 maassa. MetaEdit+ :lla tuettuja tilannekohtaisia menetelmiä on tehty muun muassa matkaviestimien suunnitteluun, voimalaitostoimitusten simulointiin, paikkatietojärjestelmien

kehittämiseen, erikoistuneeseen tilamallintamiseen, yrityskohtaisten oliomenetelmien tukeen ja jopa kemiallisten prosessien mallintamiseen.

Tutkimustulokset johtavat lähes aina jatkotutkimukseen. Niin myös inkrementaalisen menetelmäkehityksen osalta. TEKESin rahoittamassa kolmevuotisessa (1999-2001) RAMSES-tutkimusprojektissa selvitetään menetelmäkomponenttien hallintaa yhdessä Jyväskylän yliopiston, Nokia Mobile Phonesin ja MetaCase Consulting Oy:n kanssa. Lisäksi kansainvälisessä EMPEROR-hankkeessa selvitetään yhdessä muiden eurooppalaisten yliopistojen ja tutkimuslaitosten kanssa vaatimusmäärittelyä ja sen menetelmien hallintaa metamallintamisen avulla.

Juha-Pekka Tolvanen
MetaCase Consulting Oy
Ylistönmäentie 31
40500 Jyväskylä
<http://www.metacase.com>
e-mail: jpt@metacase.com

¹ FIPS, Integration definition for function modeling (IDEF0), *Federal Information Processing Standards Publication*, 183, 1993.

² CCTA (Central Computer and Telecommunication agency) *SSADM4+: reference manual*, NCC Blackwell, 1995.

³ Booch, G., Rumbaugh, J., Jacobson, I., *Unified Modeling Language- User's Guide*, Addison-Wesley, 1999.

⁴ Fitzgerald, B., (1995) *The use of system development methods: a survey*. Paper ref 9/95, University College Cork.

⁵ Russo, N., Wynekoop, J., Walz, D., The use and adaptation of system development methodologies. *Procs of International Conference of IRMA*, Atlanta, May 21-14, 1995.

⁶ Hardy, C., Thompson, J., Edwards, H., The use, limitations and customization of structured systems development methods in the UK. *Information and Software Technology*, 37 (9), 1995.

⁷ Jaaksi, A., (1997) *Object-oriented development of interactive systems*, Dissertation, Tampere University of Technology, Publications 201, Tampere, Finland.