

University of Joensuu
Department of Computer Science
International Proceedings Series 5

Tietojenkäsittelytieteen päivät 2004

24.-26.5.2004, Joensuun Tiedepuisto

Pasi Fränti ja Esko Marjomaa (toim.)

Joensuu 2004

Joensuun yliopisto
Tietojenkäsittelytieteen laitos
PL 111, 80101 Joensuu

ISBN 952-458-480-8
ISSN 1457-9448

Joensuun Yliopistopaino
Joensuu 2004

Esipuhe

Järjestyksessä seitsemännet tietojenkäsittelytieteen päivät järjestetään 24.-26.5.2004 Joensuun tiedepuistossa. Päivien ohjelma koostuu kutsuttujen puhujien esitelmistä, paneelista, kolmesta workshopista sekä seitsemästä opiskelijaesitelmästä, jotka valittiin 19 ehdokkaan joukosta kriteereinä kiinnostavuus, yleistajuisuus ja töiden tieteellinen taso. Tähän julkaisuun on koostettu päivien ohjelma, opiskelijaesitelmien tiivistelmät sekä yhteenvedot kutsuesitelmistä. Kiitän kaikkia päivien järjestelyihin osallistuneita.

Joensuussa 14. toukokuuta 2004

Pasi Fränti
Ohjelmatoimikunnan puheenjohtaja

Ohjelmatoimikunta

Pasi Fränti (pj), Joensuun yliopisto
Lea Kutvonen, Helsingin yliopisto
Esko Marjomaa, Joensuun yliopisto
Mauno Rönkkö, Kuopion yliopisto
Tapio Salakoski, Turun yliopisto
Jorma Tarhio, Teknillinen korkeakoulu

Järjestelyorganisaatio

Roman Bednarik (weppi-tiimi)
Pasi Fränti (koordinointi, workshop)
Juha Hakkarainen (tekniikka)
Markku Hauta-Kasari (TiPu-kierros)
Wilhelmiina Hämäläinen (workshop)
Ilkka Jormanainen (weppi-tiimi)
Oili Kohonen (majoitus, tilavaraukset)
Marja-Liisa Makkonen (emäntä)
Esko Marjomaa (minikonferenssi, proceedings)
Jussi Nuutinen (torvisoittokunta)
Jussi Parkkinen (PR)
Sirpa Torvinen (tiedotus)

Sisällysluettelo

Esipuhe
Sisällysluettelo

Kutsuesitelmät

Suomalaisen tietojenkäsittelytieteen identiteetti

Katsaus tietojenkäsittelytieteen tutkimukseen huippuyksikköohjelman valossa, Erkki Oja (TKK)..... 6

Esimerkki modernista sovellushakuisesta teoriasta: vuorovaikuttavat tilakoneet, Antti Valmari (TTY) 7

Teollisuuden puheenvuoro: Mitä tietojenkäsittelytieteeltä odotetaan?

Tuomo Kauranne (Arbonaut)..... 13

Tietojenkäsittelytieteen kansallinen strategia - onko sitä?

Mitä tutkimonuudistuksen yhteydessä tehty ydinainesanalyysi kertoo koulutusstrategiasta?

Jyrki Nummenmaa (TaY)..... 14

Etiikka tieteessä ja tietojenkäsittelytieteessä, Kai Kimppa (TY) 16

Minikonferenssi

Research into roles of variables, Pauli Byckling, Petri Gerdt & Seppo Nevalainen (JoY) 21

Emootiot esille! Marika Tähti & Leena Arhippainen (OY) 25

Neural network modelling of speech production and its problems, Antti Järvelin (TaY) 29

Comparison of heterogeneous proximity functions and euclidean distance, Janne Lumijärvi (TaY)..... 33

Kamada-Kawai -algoritmin muunnelma dynaamisten verkkojen tiiviiseen aseteluun, Otto Seppälä (TKK) .. 39

Konferenssi-protokollan automaattinen testaus, Antti Kervinen & Pablo Virolainen (TTY)..... 44

Todistuskompleksisuudesta Boolean piirien toteutuvuustarkastuksessa, Matti Jarvisalo (TKK)..... 49

Paneeli

Minne menee tietojenkäsittelytiede? Erkki Sutinen (JoY) pj..... 55

Workshopit

Tutkijakoulujen yhteistoiminnan avaaminen

Tietojenkäsittelytieteen tutkijakoulut 2001-2003, Pasi Fränti (JoY)..... 56

Elämänkaari-workshop

Tohtorius Naisille Lasikatto? Tarja Tiainen (TaY) 61

Miksi tietojenkäsittelytieteessä on niin vähän naisia? Wilhelmiina Hämäläinen (JoY)..... 64

Kutsuesitelmät

Katsaus tietojenkäsittelytieteen tutkimukseen huippuyksikköohjelman valossa
Erkki Oja, Teknillinen korkeakoulu

Esimerkki modernista sovellushakuisesta teoriasta: vuorovaikuttavat tilakoneet
Antti Valmari, Tampereen teknillinen yliopisto

Teollisuuden puheenvuoro: Mitä tietojenkäsittelyltä odotetaan?
Tuomo Kauranne, Arbonaut
Aimo Maanavilja, Elisa

Akatemian huippuyksikköstrategiat
Kaisa Sere, Åbo Akademi

Mitä tutkinnonuudistuksen yhteydessä tehty ydinainesanalyysi kertoo koulutusstrategiasta?
Jyrki Nummenmaa, Tampereen yliopisto

Etiikka tieteessä ja tietojenkäsittelytieteessä
Kai Kimppa, Turun yliopisto

Miten huippuyksikkö tehdään ja miten siellä työskennellään?
Ralph Back, Åbo Akademi

Katsaus tietojenkäsittelytieteen tutkimukseen

huippuyksikköohjelman valossa

Erkki Oja

TKK

Biography

Erkki Oja is Director of the *Neural Networks Research Centre* and Professor of Computer Science at the *Laboratory of Computer and Information Science*, Helsinki University of Technology, Finland. He received his Dr.Sc. degree in 1977. He has been research associate at Brown University, Providence, RI, and visiting professor at Tokyo Institute of Technology. Dr. Oja is the author or coauthor of more than 270 articles and book chapters on pattern recognition, computer vision, and neural computing, as well as three books: "Subspace Methods of Pattern Recognition" (RSP and J.Wiley, 1983), which has been translated into Chinese and Japanese, "Kohonen Maps" (Elsevier, 1999), and "Independent Component

Analysis" (J. Wiley, 2001). His research interests are in the study of principal components, independent components, self-organization, statistical pattern recognition, and applying machine learning algorithms to computer vision and signal processing. Dr. Oja is member of the editorial boards of several journals and program committees of several recent and upcoming conferences including ICANN, IJCNN, and ICONIP. He is member of the Finnish Academy of Sciences, Fellow of the IEEE, Founding Fellow of the International Association of Pattern Recognition (IAPR), and President of the European Neural Network Society (ENNS).

Esimerkki modernista sovellushakuisesta teoriasta: vuorovaikuttavat tilakoneet

Antti Valmari

Tampereen teknillinen yliopisto, Ohjelmistotekniikan laitos

1 Johdanto

Tilakoneet ovat laajalti käytetty abstraktio ohjelmistojen ja järjestelmien suunnittelussa. Tilakone on luonnollinen malli moneen tilanteeseen, ja sen perusidea on helppo ymmärtää. Kuitenkin, ehkä yllättäen, tilakoneen vuorovaikutus muiden tilakoneiden tai järjestelmän ympäristön kanssa on erittäin vaikeasti ymmärrettävä asia. Monissa suosituissa tilakoneformalismeissa sitä joko ei ole yritetty käsitellä ollenkaan, tai se on määritelty sangen epätarkoituksemukaisella tavalla (alkuperäinen SDL).

Tietojenkäsittelyteoreettinen tutkimus on noin 25 viime vuoden aikana tuottanut runsaasti tuloksia, joiden avulla vuorovaikuttavien tilakoneiden käyttäytymistä koskevia ilmiöitä voi jäsentää. Tulokset ovat hajallaan, mutta niissä on jo selvästi nähtävissä vuorovaikuttavien tilakoneiden yleisen teorian runko. Tässä esitelmässä kuvaan, minkälainen tuo runko on; kerron, miten tulokset liittyvät aikaisempiin tietojenkäsittelyteorian tuloksiin; sekä esitelen teorian näköpiirissä olevia käytännön sovelluksia.

2 Tilakoneiden tyypillinen rakenne

Tilakoneen perustana on joukko *tiloja* ja niiden välisiä *tilasiirtymiä*. Kuvissa tilat piirretään yleensä ympyröinä, soikioina tai laatikoina, ja tilasiirtymät ovat niiden välisiä nuolia. Tavallisesti yksi tai useampi tila on osoitettu erityisek-

si *alkutilaksi* esimerkiksi tyhjästä alkavalla lyhyellä nuolella. Tilasiirtymien varrelle on yleensä kirjoitettu lisäinformaatiota, ja joissakin tilakoneformalismeissa myös tiloissa sekä tilakaavion ulkopuolella voi olla lisäinformaatiota.

Ajatuksena on, että tilakone aloittaa toimintansa jostakin alkutilastaan. Kun sopiva ehto toteutuu, tilakone siirtyy tilasta toiseen niitä yhdistävää tilasiirtymää pitkin. Tilasiirtymän varrelle kirjoitettu lisäinformaatio määrittelee, millä ehdoilla siirtyminen on mahdollista, ja mitä viestintää ja muita toimenpiteitä tilakone suorittaa siirtymisen aikana. Saatuaan tilasiirtymän kokonaan suoritettua tilakone jää sen lopputilaan odottamaan, että pääsisi eteenpäin jotakin siitä tilasta alkavaa tilasiirtymää pitkin. Näin tilat esittävät hetkellisiä tilanteita, joissa tilakoneella mallinnettu kohde voi olla, ja tilasiirtymät esittävät kohteen suorittamaa laskentaa.

Ensi vilkaisulta tilakone muistuttaa hyvin paljon tietojenkäsittelyteoriasta tuttua äärellistä automaattia. Niillä on kuitenkin olennaisia eroja. Äärellisistä automaateista tuttuja lopputiloja ei tilakoneissa yleensä ole. Vaikka olisikin, tilakoneiden käyttötarkoitus on toinen kuin äärellisten automaattien, ja niinpä myös tilakoneiden matematiikka on erilaista. Tilakoneen ei liioin tarvitse olla äärellinen.

Monet tilakoneformalismit sallivat muuttujien käytön ohjelmointikielistä tuttuun tapaan (tämäkin on ero äärellisiin automaatteihin). Muuttujat määritellään usein näkymään koko tilakoneessa (esimerkki tilakaavion ulkopuolisesta informaatiosta), mutta joskus on parempi määritellä kullekin tilalle omat muuttujat (esimerkki

tilakohtaisesta informaatiosta).

Tilasiirtymään liittyy tyypillisesti *vahti*, eli looginen lauseke, joka viittaa muuttujien arvoihin tilasiirtymän alkupään tilassa. Tilasiirtymä voidaan suorittaa vain, jos vahdin arvo on **True**. Muuttujien arvoja alkupään tilassa voidaan käyttää myös laskettaessa tilasiirtymän aikana tilakoneesta ulospäin lähetettävää tietoa sekä niitä arvoja, jotka tilasiirtymän loppupään tilan muuttajat saavat tilasiirtymän valmistuttua. Loppupään tilan muuttujia voidaan käyttää myös tallettamaan tilakoneen ulkopuolelta tilasiirtymän aikana tulevaa tietoa. Tilasiirtymän suorittamisen ehtona on vahdin lisäksi se, että ulkomaailma on valmis tilasiirtymässä määriteltyyn viestintään.

Tilasiirtymät oletetaan yleensä *atomisiksi*. Se tarkoittaa, että tilasiirtymän suoritusta ei voi jättää kesken eikä jakaa pätkiksi. Kun tilasiirtymä on aloitettu, se suoritetaan loppuun asti, ja kaikki se mihin tilasiirtymä koskee on tilasiirtymän yksityisomistuksessa niin kauan kunnes tilasiirtymä on valmis. Jos kohteen toiminnan esittäminen vaatii epäatomisuutta, niin tilasiirtymä on korvattava usean peräkkäisen tilasiirtymän ja niiden välissä olevien tilojen ketjulla.

3 Yleinen vuorovaikutus

Samanaikaisesti suoritettavien ohjelmanosien väliseen viestintään on kehitetty lukuisia, toisistaan kovasti poikkeavia mekanismeja: yhteinen muisti, opastimet, sanomajonot jne. Eräs tutkimuksen tuottama tärkeä uusi näkemys on, että

kaikki tämä moninaisuus on ymmärrettävissä yhden viestintämuodon, synkronisen vuorovaikutuksen ilmentymiksi.

Esimerkiksi sanomajonon välityksellä tapahtuva viestintä mallinnetaan siten, että myös sanomajono tulkitaan tilakoneeksi, viestin lähettäminen esitetään synkronisena vuorovaikutuksena lähettävän tilakoneen ja sanomajonon välillä, ja vastaanottotapahtuma esitetään sanomajonon synkronointina vastaanottavan tilakoneen

kanssa. Käytännön ohjelmoinnissa esiintyvistä käsitteistä lähimmäksi synkronista vuorovaikutusta pääsee Adan kohtaamis- eli rendez-vous-mekanismi. Se ei kuitenkaan tavoita synkronisen vuorovaikutuksen täyttää yleisyyttä.

Synkronisessa vuorovaikutuksessa kaksi tai useampia tilakoneita suorittaa tilasiirtymän yhtäaikaan. Tätä (mutta ei pelkästään tätä) kutsutaan *näkyväksi tapahtumaksi* (englanniksi *visible action*). Suoritettavien tilasiirtymien on oltava tietyllä tavalla yhteensopivia. Yksinkertaisin tapa määritellä asia on määritellä jokaiselle tilakoneelle joukko *portteja* (*gate*). Sanotaan, että tilakone on *kytkeytynyt* portteihinsa. Monta eri tilakonetta voi olla kytkeytyneenä samaan porttiin (eri tilakoneiden portit ovat samat, jos ja vain jos niillä on sama nimi). Näkyvä tapahtuma liittyy aina johonkin porttiin, ja siihen osallistuvat täsmälleen ne tilakoneet, jotka ovat kytkeytyneet kyseiseen porttiin.

Tilasiirtymässä ilmoitetaan, minkä portin kautta tilakone haluaa vuorovaikuttaa tilasiirtymää vastaavan näkyvän tapahtuman aikana. Lisäksi tilasiirtymässä kerrotaan, kuinka monta ja minkälaista data-arvoa eli *tapahtuman parametria* tilakone haluaa viestiä tapahtuman aikana. Näkyvä tapahtuma on mahdollinen vain, jos sen osapuolet ovat samaa mieltä parametrien määrästä ja arvoista. Näkyvän tapahtuman nimi esitetään matemaattisesti muodossa $g\langle p_1, \dots, p_k \rangle$, missä g on portti, ja p_1, \dots, p_k ovat tapahtuman parametrien arvot.

Jos esimerkiksi tilakone A haluaa lähettää portin g kautta tilakoneelle B 8-bittisen luvun 3 ja B on valmis vastaanottamaan sen, niin sekä A että B ilmoittavat parametrien määräksi yksi. A :n mielestä kyseinen parametri saa olla vain 3, mutta B :n mielestä se saa olla mikä tahansa 8-bittinen luku. B :n tilasiirtymään voidaan kirjoittaa, että B tallettaa tapahtuman parametrin johonkin muuttujaansa myöhempää käyttöä varten. A :n asettaman vaatimuksen vuoksi vain sellainen tapahtuma voi tapahtua, jonka parametrin arvo on 3. Näin A määräsi parametrin arvon ja on siis *lähettäjä*. B on *vastaanottaja*, koska sille olisi kelvannut mikä arvo hyvänsä.

Tästä esimerkistä syntyvän tapahtuman nimi on $g\langle 3 \rangle$.

Esimerkki havainnollistaa, että synkronisella vuorovaikutuksella voi esittää tutut lähettämisen ja vastaanottamisen. Sillä voi esittää myös monimutkaisempia tilanteita. Samassa tapahtumassa voidaan välittää kaksi parametria, niin että A on ensimmäisen tapahtuman suhteen lähettäjän ja toisen suhteen vastaanottajan roolissa, ja B :lle päinvastoin. Yhdenkin parametrin tapauksessa lähettäjän ja vastaanottajan roolit voivat mennä sekaisin. Esimerkiksi voi olla, että A :lle kelpaa parametriksi mikä tahansa luku joka on pienempi kuin 5, mutta B vaatii, että sen on oltava parillinen ja väliltä $0, \dots, 10$. Silloin mahdollisia tapahtuman nimiä ovat (vain) $g\langle 0 \rangle$, $g\langle 2 \rangle$ ja $g\langle 4 \rangle$. Tapahtumaan voi osallistua myös enemmän kuin kaksi tilakoneita.

Jos tapahtuman parametrien arvoille on useita kaikille osapuolille kelpaavia vaihtoehtoja, niin tilakoneiden teoria ei ota kantaa siihen, mikä valitaan. Teorian näkökulmasta jokainen on yhtä lailla mahdollinen. Valinta on *epädeterministinen*. Teoria ei myöskään murehdi sitä, miten osapuolet löytävät kaikille sopivat parametrien arvot tai toteavat, että sellaisia ei ole olemassa. Tässä suhteessa synkroninen vuorovaikutus on puhtaasti teoreettinen käsite: se tuntuu vaikealta tai mahdottomalta toteuttaa käytännössä täydessä yleisyydessään, mutta sen avulla voidaan helposti esittää kaikki käytännössä esiintyvät vuorovaikutusmekanismit ja näin ollen tuoda ne yhtenäisen teorian piiriin.

Teoria sallii myös sellaiset näkyvät tapahtumat, joihin osallistuu vain yksi tilakone. Silloin vain yksi tilakone on kytkeytynyt kyseiseen porttiin, ja tapahtuman mahdollisuus tai mahdottomuus määräytyy pelkästään kyseisen tilakoneen asettamista ehdoista. Lisäksi on olemassa *näkymättömiä tapahtumia*. Näkymättömässä tapahtumassa portin paikalla on symboli τ , eikä dataparametreja ole. Tilakone suorittaa näkymättömän tapahtuman yksinään.

Samasta tilasta saa alkaa useita tilasiirtymiä. Silloin tilakone on valmiina useaan vaih-

toehtoiseen vuorovaikutukseen ja/tai näkymättömään tapahtumaan.

4 Tilasiirtymärelaatiot

Tilakoneformalismeissa esitetään vahteja, muuttujien arvoilla tapahtuvaa laskentaa ja muuttujiin sijoittamista yleensä ohjelmointikielistä tai pseudokoodista tutuin merkinnöin. Tapahtuman parametrin lähettäminen esitetään tyyliin *!lauseke* ja vastaanotto *?muuttuja*. Monimutkaisempia vuorovaikutusmuotoja rakennetaan esimerkiksi lisäämällä vastaanotettuja muuttujien arvoja rajaavia ehtoja hakasulkeissa.

Esitystavan yksityiskohdista riippumatta vain kolme tekijää on olennaisia: muuttujien arvot tilasiirtymän alkupään tilassa (merkitään jatkossa \bar{v}), tapahtuman parametrien arvot (\bar{p}), ja muuttujien arvot tilasiirtymän loppupään tilassa (\bar{v}'). Pseudokoodilla, huuto- ja kysymysmerkeillä sekä hakasulkeilla kirjoitettu teksti määrittelee loppujen lopuksi vain sen, mitkä $(\bar{v}, \bar{p}, \bar{v}')$ -yhdistelmät ovat mahdollisia. Tilasiirtymässä tapahtuva laskenta voidaan siis pelkistää relaatioksi muotoa $R(\bar{v}, \bar{p}, \bar{v}')$.

Tämä on kutakuinkin sama abstraktio kuin peräkkäisohjelman pelkistys syötteen ja tuloksen väliseksi funktioksi tai relaatioksi. Funktio riittää silloin kun peräkkäisohjelma halutaan nähdä deterministisenä. Peräkkäisohjelmakin voi olla epädeterministinen, ja silloin sen merkitys on relaatio. Tilakoneiden käytännössä epädeterministisyys on niin olennaista, että teorian on sallittava se.

Tämä abstraktio on erittäin hyödyllinen siksi, että se muodostaa selvän roolijaon ja rajapinnan tilakoneiden teorian ja ohjelmointikielten teorian välille. Tilakoneformalismin suunnittelija saa valita tilasiirtymärelaatioiden ilmaisemiseen käytettävät merkinnät miten haluaa ilman, että sillä on mitään vaikutusta tilakoneiden teoriaan. Merkintöjen merkitys — eli minkä relaation ne ilmaisevat — on puhtaasti ohjelmointikielten teorian piiriin kuuluva asia. Näin tilakoneiden teoria saadaan vapautettua sille

epäolennaisista asioista.

5 Tila-avaruus

Jos yksittäisen tilakoneen käyttäytymistä halutaan simuloida, on pidettävä kirjaa siitä, missä tilassa tilakone on, mitkä arvot sen muuttujilla on, ja minkänimisen näkymättömän tai näkyvän tapahtuman se milloinkin suorittaa ja millä parametrien arvoilla. Tilakoneen tilaa ja muuttujien arvoja yhdessä voidaan kutsua *kokonaistilaksi*, ja tilasiirtymää kokonaistilasta toiseen täydennettynä vastaavan tapahtuman parametrien arvoilla *kokonaistilasiirtymäksi*. Näin saadaan rakenne, jolla on neljä osaa: niiden kokonaistilojen joukko, joihin tilakone voi joutua; tilakoneen porttien nimet täydennettynä kaikilla periaatteessa mahdollisilla parametrien arvoyhdistelmillä; kokonaistilasiirtymät; ja alkutiloja vastaavien kokonaistilojen joukko.

Tämä rakenne sisältää kaiken olennaisen tiedon tilakoneen käyttäytymisestä. Sitä kutsutaan näkökulmasta riippuen *tila-avaruudeksi* (*state space*) tai *nimetyksi siirtymäjärjestelmäksi* (*labelled transition system*). Tila-avaruuden muodostaminen yhdestä tilakoneesta on olennaisesti sama toimitus kuin Petri-verkkojen maailmassa tunnettu *aukikelaus* (*unfolding*), ja samasta on kyse myös silloin, kun laskettavuuden teoriassa sanotaan, että jokin tieto talletetaan Turingin koneen äärelliseen tilaan.

Tila-avaruus voidaan muodostaa myös useasta vuorovaikuttavasta eli *rinnankytketystä* tilakoneesta. Tällöin kokonaistilaan otetaan kaikkien tilakoneiden tilat ja muuttujien arvot, ja kokonaistilasiirtymiä muodostettaessa otetaan huomioon synkronisen vuorovaikutuksen säännöt. Nytkin tila-avaruus sisältää kaiken olennaisen tiedon järjestelmän käyttäytymisestä.

Tila-avaruus on itse asiassa tilakone, jossa ei ole muuttujia. Niinpä voidaan sanoa, että

Tilakoneen käyttäytyminen on muuttujaton tilakone.

Tästä seuraa, että myös tila-avaruuksia voidaan kytkeä rinnan. Järjestelmän käyttäytymisen voi siis muodostaa kahdella eri tavalla: suoraan rinnankytkettyjen tilakoneiden käyttäytymisenä; tai muodostamalla kullekin osatilakoneelle käyttäytyminen ja kytkemällä ne rinnan. Voidaan varsin helposti osoittaa matemaattisesti, että molempia reittejä saadaan sama tulos.

Rinnankytkettyjen tilakoneiden käyttäytyminen voidaan laskea muodostamalla kunkin tilakoneen käyttäytyminen ja kytkemällä ne rinnan.

6 Kätkentä ja kutistus

On tavallista, että järjestelmään kuuluvilla tilakoneilla on keskinäisiä vuorovaikutustapahtumia, jotka eivät itsessään ole kiinnostavia tarkasteltaessa järjestelmää kokonaisuutena, vaikka niillä saattaa olla kiinnostavia seurauksia. Esimerkiksi pankin ja pankkiautomaatin välillä esiintyy lukuisia viestintätapahtumia, joista pankkiautomaattia käyttävä asiakas ei ole kiinnostunut. Hänelle on tärkeää vain se, saako hän rahat ja veloitetaanko tiliä oikein.

Tätä tarkoitusta varten tilakoneiden teoriasa on *kätkentänä* (*hiding*) tunnettu operaatio. Portin g kätkeminen tarkoittaa sitä, että kyseisessä portissa esiintyviä tapahtumia (joiden nimet ovat muotoa $g\langle p_1, \dots, p_k \rangle$) ei näytetä ulospäin sellaisenaan, vaan niiden tilalla näytetään τ .

Kätkentä ei yksinään muuta järjestelmän tila-avaruutta paljoakaan: se vain muuttaa kokonaistilasiirtymien nimiä τ ksi. Tilakoneiden teoriassa tunnetaan kuitenkin myös *kutistusoperaatioita*, joilla tila-avaruuksista voi poistaa τ -kaaria. Kutistus saattaa pienentää tila-avaruutta valtavasti.

Kutistettukin tila-avaruus on muodollisesti tilakone. Siitä seuraa, että kätkentää ja kutistusta voidaan soveltaa hierarkkisesti. Jaetaan järjestelmä osajärjestelmiin, muodostetaan kullekin osajärjestelmälle tila-avaruus, kytketään ne

rinnan, lasketaan niiden yhdistetty tila-avaruus, kätketään siitä ne portit joita ei tarvita osajärjestelmän ulkopuolella, ja kutistetaan. Osajärjestelmän tila-avaruus voidaan muodostaa samalla tavalla pienempien osa-osajärjestelmien tila-avaruuksista ja niin edelleen. Toistuvien kutistusten ansiosta tämä tekniikka tekee mahdolliseksi hallittavan kokoisen kutistetun tila-avaruuden muodostamisen lukuisille sellaisille järjestelmille, joiden tila-avaruus on aivan liian suuri suoraan muodostettavaksi.

Kuten aiemmin, tämänkin tekniikan perustelemiseksi tarvitaan teoreema, joka sanoo, että lopputulos on riippumaton siitä, kutistetaan ko kerralla vai vaiheittain. Tämä tulos ei kuitenkaan ole ollenkaan itsestään selvä. Päinvastoin, sen tavoittelu on johtanut suureen määrään monimutkaista matematiikkaa. Paljastuu, että on otettava kantaa siihen, mitkä tilakoneen käyttäytymisen piirteet katsotaan tärkeiksi ja mitkä ei.

Osa kutistusalgoritmeista on hyvin monimutkaisia. Niiden suunnittelussa on hyödynnetty sekä algoritmiikan että kompleksisuusteorian tuloksia.

7 Abstraktit semantiikat

Kutistaminen tuhoaa ainakin jotakin tietoa järjestelmän käyttäytymisestä. Tämän vuoksi on tärkeää, että määritellään, mitä tietoja saa muuttaa ja mitä ei. Tällaista määritelmää kutsutaan *abstraktiksi semantiikaksi*.

Olisi mukavaa, jos olisi olemassa yksi "ainoa oikea" abstrakti semantiikka, mutta niin ei ikävä kyllä ole. Erilaisia abstrakteja semantiikkoja on kehitetty kymmenittäin, ehkä jopa sadoittain. Osittain kyse on siitä, että kun ei heti osattu kehittää kaikki toiveet täyttävää semantiikkaa, niin tehtiin erilaisia välivaiheita. Abstraktista semantiikasta ei ole helppoa tehdä sellaista, että edellä kuvattu vaiheittaisen kutistamisen tekniikka johtaisi aina ja joka tilanteessa oikeaan lopputulokseen. Toinen syy on se, että järjestelmän kokoamiseksi osistaan voidaan käyttää muutakin kuin rinnankytkentää ja kätken-

tää, ja parhaan semantiikan valinta riippuu siitä, mitä muuta otetaan mukaan. Vielä yksi syy on näkemuserot siitä, halutaanko jokin käyttäytymispiirre säilyttää kutistuksissa vai ei.

Tutkijat ovat jokseenkin yksimielisiä siitä, että ainakin on tarpeen säilyttää kaikki näkyvien tapahtumien jonot, jotka voivat syntyä kun järjestelmää suoritetaan alkutilasta lähtien nolla tai useampia askelia. Tässä sallitaan, että suoritus katkaistaan keinotekoisesti milloin tahansa, koska tarkasteltavana olevan ominaisuuden kannalta ei yleensä ole tarpeen jatkaa suoritusta niin pitkälle kuin mahdollista. Esimerkiksi pankkiautomaatti voi suorittaa jonon *kortti sisään, nostopyyntö, anna rahat, veloita tiliä, kortti ulos*, jonka jälkeen se jatkaa tavalla joka ei juuri nyt kiinnosta. Jono *kortti sisään, nostopyyntö, veloita tiliä, kortti ulos* olisi virheellinen, koska siinä asiakas jäi rahatta vaikka tiliä veloitettiin.

Pelkästään tällaiset jonot eli *jäljet* säilyttävä semantiikka tunnetaan *jälkisemantiikkana* (*trace semantics*). Se on melko (vaan ei ihan) tarkasti sama asia kuin äärellisen automaatin hyväksymä kieli. Jälkisemantiikka ei kuitenkaan säilytä esimerkiksi tietoa siitä, voiko pankkiautomaatti jumiutua kesken kaiken (paitsi jos se tekee sen joka kerta).

Enemmän tietoa säilyttävät semantiikat voidaan jakaa kahteen ryhmään sen mukaan, perustuvatko ne *lineaariseen* vai *haarautuvaan* aikakäsitykseen. Niiden eroa voi karkeasti luonnehtia sanomalla, että lineaariset semantiikat hukkaavat tiedon siitä, mitä vaihtoehtoja toteutuneelle tapahtumien kululle olisi sen eri vaiheissa ollut. Lineaarinen semantiikka näkee vain, että työtoverisi ottaa joskus kahvia ja joskus teetä; mutta haarautuva semantiikka näkee myös, tekikö hän valinnan kahvin ja teen välillä ennen kuin astui kahvioon vai sen jälkeen.

Kuitenkin lineaarisetkin semantiikat säilyttävät tietoa siitä, mitkä ovat vaihtoehdot välitömästi seuraavaksi näkyväksi tapahtumaksi, kun annettu jälki on suoritettu. Tämä johtuu siitä, että se on välttämätöntä *lukkiutumien* eli sellaisten tilanteiden paljastamiseksi, jossa jär-

jestelmän mikään tilakone ei voi edetä. Lineaarinenkin semantiikka näkee, jos työkaverisi kahvin ollessa loppu jää odottamaan sen sijaan että muuttaisi mielensä ja ottaisi teetä.

Toinen tärkeä jakoperuste on semantiikan suhtautuminen *pillastumaan* (*divergence*), eli tilanteeseen, jossa järjestelmä jostakin hetkestä alkaen touhuu loputtomasti saamatta mitään näkyvää aikaan. Toisessa ääripäässä pillastumia ei noteerata mitenkään, toisessa pillastuma katsotaan niin pahaksi virheeksi, että jos järjestelmä voi jonkin jäljen jälkeen pillastua, niin ei tarvitse tietää, mitä muuta se voi saman jäljen jälkeen tehdä.

Vaikka semantiikkoja on kaikkiaan paljon, teorian soveltamisen kannalta mielenkiintoisia on aika vähän. Useimmissa käytännön tapauksissa sopiva semantiikka määräytyy yksikäsitteisesti, kun on ratkaistu suhtautuminen ajan haarautumiseen ja pillastumiin sekä päätetty järjestelmän kokoamisessa sallittavien operaattoreiden joukko. Tiedetään, että jos kaikesta huolimatta jäljelle jää useita vaihtoehtoja, niistä kannattaa valita se, joka säilyttää vähiten informaatiota.

Uusin tutkimus viittaa siihen, että jos järjestelmät ovat deterministisiä eräässä luontevassa mielessä, niin kaikki abstraktit semantiikat romahtavat jälkisemantiikan kanssa samaksi. Jos tämä tulos vahvistuu, niin se selittää, miksi vuorovaikuttavien tilakoneiden yhteydessä esiintyy lukuisia semantiikkoja, kun esimerkiksi abstraktien tietotyyppien yhteydessä — jotka ovat tilakoneita nekin — selvittää yhdellä. Abstraktit tietotyypit suunnitellaan tarkoituksella deterministisiksi, mutta vuorovaikuttavien tilakoneiden tapauksessa epädeterminismin välttäminen on mahdotonta useastakin syystä.

8 Teorian sovelluksia

Ehkä eniten käytetty tilakoneiden teorian käytännön sovellus on järjestelmien algoritmien verifiointi eli automaattinen toiminnallinen tarkastus. Kutistetun tila-avaruuden hierarkkinen

muodostaminen sekä monet muut matemaattisesti syvälliset tekniikat, joihin ei tässä voida mennä, tekevät varsin monimutkaisesti käytettyjen järjestelmien verifiointin mahdolliseksi. On tavallista, että perusteellisestikin testatusta järjestelmästä löytyy virheitä, kun sitä tutkitaan verifiointimenetelmillä. Ikävä kyllä nykyisten verifiointimenetelmien kapasiteetti on vielä riittämätön ja käyttö liian vaikeaa laajamittaiseen soveltamiseen teollisuudessa. Pienehköjen kriittisten osien tarkastamiseen nykyiset menetelmät pystyisivät, mutta vain, jos niitä käytämässä on erikoiskoulutettua henkilökuntaa.

Tilakoneiden teorian soveltamista järjestelmien testaamiseen on tutkittu kasvavassa määrin viime vuosina sekä Suomessa että ulkomailla. Testaus voi hyvin harvoin olla täydellistä. Tämän vuoksi testauksessa pyritään ainoastaan suuntaamaan käytettävissä olevat voimavarat järkevästi; toisin sanoen löytämään mahdollisimman paljon virheitä mahdollisimman vähällä työllä. Tilakoneiden teoriasta haetaan apua tähän.

Koska tilakoneen käyttäytyminen on tilakone ja sopivaa semantiikkaa käytettäessä kutistaminen on sallittua milloin tahansa järjestelmää rakennettaessa, voi tilakoneilla laskea hyvin vapaasti. Tunnetaan muun muassa teoreema, joka sanoo, miten muuttujallisia tilakoneita voi yhdistää yhdeksi muuttujalliseksi tilakoneeksi, jonka tila-avaruus on sama kuin alkuperäisten tilakoneiden tila-avaruuksien rinnankytkentä. Tämä tekee periaatteessa mahdolliseksi monenlaisten järjestelmien uudelleenorganisointitehtävien ratkaisemisen mekaanisesti laskemalla. Se vaatii kuitenkin niin paljon matemaattista osaamista, että sitä ei liene juurikaan sovellettu käytännössä.

Ehkä kaikkein tärkein tilakoneiden teorian käytännön hyöty on sen synnyttämissä näkemyksissä. Tilakoneiden teoriaa tunteva ymmärtää rinnakkaisohjelman sekä rinnakkaisuutta esittävän formalismin suunnittelun vaikeudet, ja osaa siksi suhtautua asiaan sen vaatimalla huolellisuudella.

Mitä tietojenkäsittelytieteeltä odotetaan?

Tuomo Kauranne

President and Chairman of Arbonaut Ltd.

Biography

Arbonaut is a Location Based Services company based in Joensuu, Finland, with teams in other European countries. Arbonaut focuses on Location Messaging – messaging amended with location information. Location Messaging is a technology that allows very fast creation of valuable business-to-business location based services, such as Fleet Management, Personnel and Asset Security and Tracking services.

Tuomo Kauranne founded Arbonaut in 1994, while conducting research in signal processing at the University of Joensuu. During the 1990's, he and his company participated in several EU funded research and development projects in information technology and telematics. One of these designed the first combined GPS and GSM handset in the world, produced by Benefon, a Finnish handset maker. Dr. Kauranne is a co-founder of several innovative start-up companies that include three call center service and technology companies.

Dr. Kauranne currently also conducts research and teaches at Lappeenranta University of Technology in Applied Mathematics. Dr. Kauranne has consulted companies and research establishments on scalable computing technology in the United Kingdom, Germany, Italy and the U. S. for over ten years. He has been an evaluator for the European Commission on several occasions in projects related to High performance computing and medical information technology.

Remote sensing applications developed by Dr. Kauranne are now world technology leaders in forest measurement technology from aerial and satellite images, in the United States and in Europe, sold through Arbonaut's U. S. based joint venture Falcon Informatics. He has participated in the management of many European Union funded research projects in the area of Information Technology in public services and telemedicine.

Prior to moving back to his native Finland in 1993, Dr. Kauranne was working at the European Centre for Medium-range Weather forecasts in Reading, UK from 1988 to 1992 as a team leader in parallel computing. The parallel computing technologies developed by Dr. Kauranne's team have been adopted in all major operational weather forecasting centers since the early 1990's and also run on the world's fastest supercomputer The Earth Simulator in Yokohama that features more than 5000 ultrafast processors.

Before moving to the United Kingdom, Mr. Kauranne was a project manager at the Center for Scientific Computing in Finland, in charge of the procurement of the first supercomputer to Finland in 1988.

Mr. Kauranne holds a doctorate in Applied Mathematics from Lappeenranta University of Technology, an Lic Phil Degree from the University of Jyväskylä in Information technology, and a MSc in Computer Science at the University of Helsinki. He is the co-author of four international scientific monographs.

Mitä tutkinnonuudistuksen yhteydessä tehty ydinainesanalyysi

kertoo koulutusstrategiasta?

Jyrki Nummenmaa

Tampereen yliopisto

jyrki.nummenmaa@uta.fi

Tutkinnonuudistuksen yhteydessä yliopistoja ja sitä myöten laitoksia on patistettu tekemään ydinainesanalyysejä. Asia saattaa tavallaan tuntua uudelta, mutta se ei sitä kuitenkaan voine täysin olla – laitokset ovat jollain perusteella aikaisemminkin valinneet ne ainekset, jotka ovat olleet opetuksen kulmakivi. Ohjelmoinnin opetus eri tavoin on ollut tyypillisintä tietojenkäsittelyopin ydinainesta. Alan kehittyessä myös ydinaines on kehittynyt. Uuden tutkintojärjestelmän myötä luonnontieteellisellä alalla pyritään koordinoimaan tutkintojen ydinaineskehitystä.

Tietojenkäsittelyn alalla työtä samanaikaisesti helpottaa ja vaikeuttaa ACM:n ja IEEE:n ydinainestyö, josta löytyy tietoa verkosta (katso esim.

<http://www.computer.org/education/cc2001/report/>). Helpotusta tuo toki se, että alalla on kansainvälisiä suosituksia joita on ilmeisesti/toivottavasti pohdittu hieman laajemmalla rintamalla. Ehdotuksia ei kuitenkaan ole yhtä ainoaa, vaan niitä on erikseen ainakin ”computer science”, ”information systems” ja ”software engineering” –aloille. Suomessa ei kuitenkaan koulutus ole yleisesti ottaen segmentoitunut näin, vaikka esimerkiksi erityisesti tietojärjestelmäalalle suuntautuneita laitoksia onkin. Tämä tarkoittaa sitä, että ydinainesanalyysejä varten suosituksia pitäisi valikoiden yhdistellä.

Pääsääntöisesti laitokset ovat ilmeisesti tähän asti tehneet omaa analyysiaan

”computer science” –ytimen pohjalta. Tämä on ilmeisesti hyvä valinta, vaikka esim. mallintamista saattaa suomalaisessa koulutuksessa tyypillisesti olla enemmän kuin tuossa suosituksessa, ja vastaavasti jotain saattaa olla vähemmänkin.

Toinen ongelma on se, kuinka kurssit oikeastaan pitäisi kuvata. ”Ydinaines” terminä kuulostaa opetussisältöpohjaiselta. Kuitenkin tietojenkäsittelyssä oikeastaan ratkaisevaa on se, mitä opiskelijat osaavat opintojensa jälkeen tehdä, enemmänkin kuin se, mistä he ovat kuulleet. ACM:n kuvaukset ovat osittain askel oikeaan suuntaan, mutta pitemmällekin voitaisiin mennä. Kurssikuvaukset on perinteisesti opinto-oppaissa kuvattu varsin lyhyesti ja askeettisesti. Jonkinlaisena toisena ääripäänä voitaneen pitää koko opetuksen ja oppimateriaalien kuvaamista. Toisaalta tähänkin sisältyy oikeastaan toive siitä, että opiskelijat perehtyvät heitä varten tarkoitettuun kurssimateriaaliin. Oikeastaan osaamista arvioitaessa ratkaisevaa lienee se, mitä opiskelijoilta vaaditaan kurssisuoritusta varten – siis millaisia tehtäviä heidän pitää ratkaista ja minkä tasoisia ratkaisuita heiltä edellytetään.

Vaikka äsken tavallaan vähäteltiin kurssisisältöjen kuvausta, niin toki nekin ovat informatiivisia ja tärkeitä. Yhteinen opetusmateriaalipankki voisi olla myös hyödyllinen väline sekä ydinainesinformaatiota varten että muuten laitosten väliseen opetusyhteistyöhön.

Etiikka tieteessä ja tietojenkäsittelytieteessä

Kai K. Kimppa

Yliassistentti, Turun yliopisto, Informaatioteknologian laitos

E-mail: kakimppa@it.utu.fi

Puhelin: +358 2 333 8665

Johdanto

Tietojenkäsittelytieteet vaikuttavat yhteiskuntamme toimintaan monin eri tavoin. Monet näistä vaikuttamisen muodoista ovat positiivisia; Internet tarjoaa meille uuden tavan kommunikoida ja jakaa informaatiota, tietokoneet mahdollistavat uudenlaisen tavan kirjoittaa, tehdä työtä, viettää vapaa-aikaa ja ratkaista ongelmia, jotka aiemmin olivat liian työläitä ratkaista. Mutta tietojenkäsittelytieteet mahdollistavat myös sellaisia asioita joiden on — ainakin perinteisesti — ajateltu heikentävän ihmisten hyvinvointia. Tietotekniikkaa voidaan käyttää uudenlaiseen valvontaan digitaalisen hahmontunnistuksen muodossa, tietokannoista voidaan louhia markkinointi-informaatiota jolla suoramainonnasta voidaan tehdä entistä tunkeilevampaa, informaation kerääminen henkilöistä agenttien avulla mahdollistaa henkilökohtaisen ja julkisen yhdistämisen tavalla, joka saattaa haitata vaikka työhön pyrkimistä. Tällaisten, enemmän tai vähemmän selkeästi moraalisesti ongelmallisten kysymysten lisäksi tietojenkäsittelytieteet mahdollistavat sellaisten asioiden tekemisen, jotka ovat moraalisesti uusia, ratkaisemattomia kysymyksiä. Täsmäpommien kehitys ja niiden käyttö ei olisi mahdollista ilman tietotekniikkaa. Onko, kuten Richard De George (2003) ajattelee, valtioilla moraalinen velvollisuus

kehittää täsmäpommeja jotta sodan aikana siviiliuhrien määrä voidaan minimoida? Vai onko Andy Bissett (2003) oikeassa sanoessaan, että sodassa tapetaan aina niin paljon kuin voidaan, eivätkä täsmäpommit oikeasti vähennä uhreja vaan ennemminkin niitä voidaan käyttää tekosyynä lähteä sotiin joita ei muuten ollenkaan tulisi? Tulisiko bioinformatiikkaa käyttää geneettisen muuntelun apuna; geneettisen muuntelun, joka nyky muodossaan (ja tulevissa muodoissaan, mitä ne lienevätkään) ei olisi lainkaan mahdollista ilman tietojenkäsittelytieteitä?

IT-ammattilaisen vastuu

Deborah Johnson (2001) lähtee pohtimaan IT-ammattilaisen (on hän sitten tutkija tai työntekijä) moraalista vastuuta IT-ammattilaisilla vallasta ja siten myös vastuusta käsin. Hän vertailee IT-ammattilaista perinteisiin ammattilaisryhmiin joilla on katsottu olevan suoraa valtaa ja joilla siksi on ollut myös erityisen selkeät eettiset velvoitteet, jopa kirjoitetut eettiset säännöt. Näitä ryhmiä ovat muiden muassa lääkärit ja juristit. Vaikka IT-ammattilaisten työ ei ole samoin erikoistunut kuin vaikka juristien, on joukossamme paljon henkilöitä, joilla on erikoisasemansa takia valtaa ja vastuuta.

Myös IT-ammattilaisille on kirjoitettu eettisiä säännöstöjä. Näistä

ehkä tunnetuin on ACM:n (Association for Computing Machinery) Code of Ethics (ACM, 2001). Vastaavanlaisia säännöstöjä on kirjoitettu myös muiden tietojenkäsittely- ja tietotekniikan liittojen toimesta. Suomessakin TTL (Tietotekniikan liitto) ry (TTL ry, 2002) on toteuttanut suomalaisille IT-ammattilaisille suunnatun eettisen säännöston. Myös IFIP (International Federation for Information Processing) on yrittänyt luoda eettistä, kansainvälistä säännöstöä. Kulttuurirelativistisista syistä tällaisen yleisen ja yhteisen säännöston kirjoittaminen ei kuitenkaan ole onnistunut. Niinpä IFIP:n SIG (Special Interest Group) 9.2.2 Framework on Ethics keskittyykin nyt luomaan ohjeistusta eettisten säännöstöjen kirjoittamisesta kansallisille liitoille joilla eettistä säännöstöä ei vielä ole.

Johnson (2001) kysyy ovatko tällaiset säännöstit tietojenkäsittelytieteiden tekijöille ja ammattilaisille tarpeen? Mitä sellaista tietojenkäsittelytieteet sisältävät, joka aiheuttaisi tällaisen tarpeen? Hänen vastauksensa on, että vaikka IT-ammattilaiset eivät tyypillisesti käytäkään samanlaista suoraa valtaa kuin esimerkiksi juuri juristit (huonon juristin työn seuraukset ovat asiakkaalle välittömästi merkitseviä) tai lääkärit (jos lääkäri ei hoida potilaitaan hyvin, seuraukset ovat myös välittömiä), niin tietojenkäsittelytieteiden harjoittajilla on merkittävästi välillistä valtaa; ja vallasta seuraa vastuu. Esimerkiksi suunniteltaessa terveydenhuollon järjestelmiä ja laitteita, on vastuu potilaan terveydestä osittain myös järjestelmän tai laitteen suunnittelijan harteilla. Kun tehdään kulunvalvontajärjestelmiä, käyttää järjestelmän kehittäjä valtaa sulkea

jotkut pois järjestelmän sallimista tiloista. Kehitettäessä tiedonlouhintamenetelmiä helpotetaan informaation löytämistä—informaation löytämistä mistä, tai *kenestä*, ja miksi? Esimerkkejä tietojenkäsittelytieteitä koskevista kysymyksistä ja kysymysalueista katsotaan seuraavassa luvussa.

Tietojenkäsittelytieteitä koskevia eettisiä kysymysalueita

Monet tietojenkäsittelytieteitä koskevista eettisistä kysymysalueista ovat toistensa kanssa päällekkäisiä. Tyypillisesti esimerkiksi tietoon pääsyä koskevat kysymykset leikkaavat immateriaalioikeuskysymyksiä ja autorisoimatonta kopioimista. Vastaavasti valvonnan tehostamiseen liittyvät kysymykset ja yksityisyyskysymykset koskevat usein toisiaan. Seuraavassa esitelen kuitenkin kunkin alueen erikseen, olkoonkin, että tämän tyyppisiä päällekkäisyyksiä on varmasti havaittavissa käytetyissä esimerkeissä.

Pääsy tietoon ja ilmaisuuspaus

Informaatio- ja kommunikaatioteknologialla voidaan edesauttaa tiedon leviämistä yhteiskuntaan. Kysymykseksi nousee kenellä on oikeus mihinkin tietoon? Vaikka Internet voitaisiin nähdä ns. "creative commonsina" (viite), se ei ole muotoutunut sellaiseksi ja viimeaikoina informaatioon käsikiappääsy on heikentynyt entisestään, erityisesti lainsäädännöllisistä, mutta myös teknisistä esteistä johtuen. Erilaisia näkemyksiä löytyy tietoanarkiasta täydelliseen informaation omistusoikeuteen saakka.

Tekoäly

Vaikka keinoälyteknologiat eivät ole ainakaan vielä näkyvissä olevassa tulevaisuudessa kykeneviä itsetajuisuuteen, on toisinaan syytä pohtia myös tulevaa. Jos tulevaisuuden ongelmiin ei varauduta ajoissa, saattaa syntyä saman kaltainen tilanne kuin Suomessa on nyt keinohedelmöityksen suhteen. Ensimmäinen keinohedelmöitetty henkilö syntyi 1984, ja jo tänä vuonna, 2004, Suomen eduskunta on pohtinut lakia siitä, kenellä on oikeus valtion avittamaan keinohedelmöitykseen ja kenellä ei.

Työ- ja bisnesetiikka

Mitä velvollisuuksia ja oikeuksia tietojenkäsittelytieteen ammattilaisilla on työnantajaansa, asiakastaan, yhteiskuntaa tai ihmiskuntaa kohtaan? Onko syytä olettaa, että myös työssään tietojenkäsittelytieteiden ammattilaiset noudattavat moraalisia toimintatapoja, silloinkin, kun niistä olisi työn välittömän suorittamisen kannalta haittaa? Työstä erottamisen uhallakin?

Tiedon louhinta

Tiedon louhinta (data mining) tarjoaa meille mahdollisuuksia kartoittaa mm. sairauksia, mutta samalla se tarjoaa myös mahdollisuuksia vaikka entistä kohdistetumpaan mainontaan—siitä riippumatta haluavatko asiakkaat sitä vai eivät.

Päätöksentekoa algoritmit

Liiallinen luottaminen päätöksentekoa algoritmien toimivuuteen ja manuaalisen tai älyllisen tarkastamisen tekemättä jättäminen saattaa johtaa jopa katastrofeihin. Tyypillinen esimerkki—hyvin alhaisen teknologisen asteen—automatisoidusta päätöksenteosta olisi esimerkiksi järjestelmä joka tarkkailee

potilaan lääkintätasoa. Jo pilkkuvirhe ohjelmassa saattaa aiheuttaa vakavia seurauksia.

Järjestelmä ja käyttöliittymäsuunnittelu

On esitetty teorioita, että käytännöllä järjestelmä- ja käyttöliittymäsuunnittelussa hyväksi IT-ammattilaisten eettisiä säännöstöjä saataisiin aikaan ohjelmistoja, jotka olisivat paremmin käyttäjien tarpeisiin vastaavia. Ongelmaksi muodostuu usein yritysten tarve saada voittoa joka saattaa määrätä kehityssuuntia jotka eivät ole käyttäjän kannalta optimaalisia.

Digitaalinen tunnistus

Digitaalinen tunnistus saattaa olla mainio keino varmistaa kulkijoiden oikeus olla alueella. Toisaalta meillä ei ole takeita siitä mihin näillä kerättyä informaatiota—erityisesti yhdistettynä muuhun informaatioon—käytetään. Tai jos tunniste, esimerkiksi digitaalinen allekirjoitus, onnistutaan väärentämään on sen väärennökseksi todistaminen hankalampaa kuin perinteisen allekirjoituksen.

eTerveys, eHallinto, eÄänestys, eOpetus, e?

Kaikissa näissä on yhteisenä ongelmana informaation paikkaansa-pitävyyden kysymys. Miten tahansa teknisesti yritetäänkin varmistaa sen oikeellisuus, aina löytyy keinoja kiertää tunnisteet.

Pelietiikka

Ovatko tietokonepelit pahasta pelaajille? Jos, niin miksi? Auttaako Carmageddonin kaltainen peli meitä kehittämään itseämme eettisemmiksi toimijoiksi (tuskin?) vai vain purkamaan aggressioitamme ”viattomasti”?

Sukupuoli- ja muut syrjintäkysymykset

Miksi tietojenkäsittelytieteiden tekijät ovat pääosin miehiä? Johtuuko tämä tietojenkäsittelytieteiden luonteesta vai sen tekijöistä? Miksei tietojenkäsittelytieteiden sovelluksia kehitetä kaikkien, myös eri tavoin rajoitteisten, käytettävissä oleviksi?

Immateriaalioikeuskysymykset

Tyypillisesti on ajateltu, että tietentekijöiden tarkoituksena on luoda uutta tietoa. Nykyisten immateriaalioikeuksien vallitessa tämän tiedon levittäminen on kuitenkin entistä rajatumpaa. Onko ohjelmiston koodi suojattavissa olevaa tietoa vai ei? Tyypillisesti on ajateltu, että matemaattiset algoritmit (joihin ohjelmistotkin kuitenkin kuuluvat) ovat tieteellisiä ideoita, joita ei voi suojata. Tämä ajattelu on kuitenkin ohjelmistojen kohdalla väistyvää.

Sodankäynti

Sodassa käytetään yhä enenevässä määrin tietojenkäsittelytieteiden tarjoamia mahdollisuuksia. ”Fiksuja” taistelupukuja, joiden avulla sotilaat pystyvät kommunikoimaan keskenään ja jotka seuraavat taistelukentän tapahtumia käytetään yhä enenevässä määrin tehostamaan sodankäyntiä.

Paikannusteknologiat

Jokainen jolla on mukanaan matkapuhelin on paikannettavissa. Se, että lailla tällainen toiminta on kiellettyä (ellei siihen erikseen ole saatu poikkeusta), ei takaa sitä etteikö tällaista paikannusta silti tehtäisi. Vastaavan tyyppinen esimerkki on vaikka jo tapahtunut puhelutietojen tarkkailu. Missä on

mahdollisuus, siellä voi tapahtua väärinkäytöksiä.

Yksityisyys

Kenellä on oikeus saada tietoa yksilöistä? Kenellä on oikeus hallinnoida tietoa yksilöistä? Internetin haulla löytää itsestään helposti paljon digitaalisia jälkiä, joiden yhteenkerääminen on yhä helpompaa ja helpompaa mutta joiden hävittäminen, jos ne ovat haitallisia tai väärä on yhä vaikeampaa ja vaikeampaa.

Tietoturva

Miten tietoa voidaan turvata? Onko ongelma tekninen vai inhimillinen? Vaikuttaa siltä, että vaikka kehitettäisiin millaisia teknisiä tietoturva-menetelmiä, niin ”minkä ihminen saa lukkoon, sen ihminen saa auki”.

Lähteet

ACM (2001). Professional Standards, <http://www.acm.org/serving/ethics.html>

Bissett, Andy K. (2003). Carl von Clausewitz and high technology war, proceedings of CEPE 2003, The Fifth International Conference on Computer Ethics – Philosophical Enquiry, Boston College, Chestnut Hill, MA, June 25-27, 2003, pp. 14–26.

De George, Richard T. (2003). Post 9/11: Computers, ethics and war, keynote address in CEPE 2003, The Fifth International Conference on Computer Ethics – Philosophical Enquiry, Boston College, Chestnut Hill, MA, June 25-27, 2003.

Johnson, Deborah G. (2001). Computer Ethics (3rd edition). Prentice Hall, New Jersey, USA.

TTL ry (2002). Tietotekniikan ammattiettiikka <http://www.tt-tori.fi/pls/ttl/docs/F1864587862/Eettisetetusivuv23.htm>

Minikonferenssi

Research into roles of variables

Pauli Byckling, Petri Gerdt & Seppo Nevalainen
Joensuun yliopisto

Emootiot esille!

Marika Tähti & Leena Arhippainen
Oulun yliopisto

Neural network modelling of speech production and its problems

Antti Järvelin
Tampereen yliopisto

Comparison of heterogeneous proximity functions and euclidean distance

Janne Lumijärvi
Tampereen yliopisto

Kamada-Kawai -algoritmin muunnelma dynaamisten verkkojen tiiviiseen asetteluun

Otto Seppälä
Teknillinen korkeakoulu

Konferenssiprotokollan automaattinen testaus

Antti Kervinen & Pablo Virolainen
Tampereen teknillinen yliopisto

Todistuskompleksisuudesta Boolean piirien toteutuvuustarkastuksessa

Matti Järvisalo
Teknillinen korkeakoulu

Research into Roles of Variables

Pauli Byckling, Petri Gerdt, Seppo Nevalainen

University of Joensuu, Department of Computer Science

Pauli.Byckling@cs.joensuu.fi

Petri.Gerdt@cs.joensuu.fi

Seppo.Nevalainen@cs.joensuu.fi

1. Background

Roles of variables [13] are a new concept that can be utilised, e.g., in teaching programming to novices. Roles represent programming knowledge on a higher level than simple programming language knowledge. Role knowledge has traditionally been tacit but it can be made explicit and thus help students to understand the ways variables are used in programs. Roles are easy to adopt in teaching, too: in one study, computer science teachers learned in less than an hour to recognise roles in their typical uses with 90 % accuracy [2].

The research into roles of variables consists of several different perspectives. In this paper we present research fields of three postgraduate students. The rest of the paper is structured as follows: chapter 2 describes the research into effects of the role concept in strategic programming knowledge. In chapter 3 the research concerning attention and creation of program knowledge is presented. Finally chapter 4 outlines automatic role detection from program code.

2. Roles and Strategic Programming Knowledge

Roles of variables can be utilised in many ways, but current research activities concentrate mainly on how they can be utilised in teaching programming to novices. The role concept is a unique way in structuring programming knowledge and one of our main interests is to find out how it can be best used in teaching novice level programming. To do that we will have to reveal actual influences of the role concept as a teaching method in action, i.e., to find out the effects of the role concept in gaining strategic programming knowledge when compared to more traditional teaching methods.

2.1 Objectives

One primary objective in the research is to study whether roles of variables will eventually help novices to gain deeper comprehension in programming especially in the beginning of their studies. The research will try to find answers for questions like "how does the role concept effect in the progression of programming methods and comprehension strategies in programming at the novice level". Another research question is to study the effects

of the role concept in strategic programming knowledge among expert programmers.

2.2 Research Work and Progress

The role concept has been utilised in a classroom experiment conducted in fall 2002 [14]. In this experiment the subjects were divided into three groups: one receiving normal lectures and exercises (the traditional group), one attending lectures with systematic use of variable roles throughout the course (the roles group), and one attending the same lectures as the roles group but using role-based animator in exercises (the animation group).

The classroom experiment comprised two kinds of protocol tasks: individual program comprehension tasks and program creation tasks made in pairs. Therefore the outcome of the tasks in the experiment consists also of two types: program summaries, and program construction speak-aloud protocols. The analysis of these protocols is the primary ongoing task in this study. As the whole idea of roles is quite new, the first stage in the research work has been discovering and validating appropriate methods for the analysis.

For the purpose of analysing program summaries we have tried out Good's program summary analysis scheme [4]. We have performed an inter-rater reliability analysis in order to see the applicability of Good's scheme for our purposes [3]. As this work is now in its final stage we will next concentrate on analysing the program summaries with Good's method. This method will at the very least reveal possible differences in the level of program information between students in the three groups.

Research methods concerning programming protocol tasks are still mainly at abstract level. One idea we have thought is to apply Rist's model

[11] of focal expansion and plans in programming. The main goal is to find if there are any differences in variable plan knowledge between the groups.

The methods we have adopted this far are measuring programming knowledge as a whole but there are still open questions about how accurately these methods measure those details we are interested in. So the process to find appropriate research methods still continues.

3. Attention and Creation of Program Knowledge

When teaching programming to novices, visualizations may be used to make both programming language constructs and program constructs more comprehensible [5, 6]. Petre and Blackwell [9] note that visualizations should not work in the programming language level because within-paradigm visualizations, i.e., those dealing with programming language constructs are uninformative. Hence visualization of higher-level program constructs should be preferred [15].

Roles represent programming knowledge on a higher level than simple programming language knowledge. PlanAni [15] is a role-based program animation system, where each role has a visualization–role image—that is used for all variables of the role.

In our research we will study questions related to the content of visualizations, the focus of attention, and creation of program knowledge.

3.1 Objectives

In our research, we are interested in finding answers to the following central questions:

To what extent and in what way does the content of a visualization influence

the focus of attention and the creation of program knowledge?

The goal of our research is to collect detailed information about the way different role images, different organization of the images, and different animation of the images affect the focus of attention. We will observe, how these possible differences in the focus of attention affect the creation of program knowledge.

3.2 Research Work And Progress

In our research, we are using PlanAni with different visual appearances as a stimulus. Eye tracking camera is used to gather point of gaze data that provides information of subjects' attention.

During January 2004 we carried out an experiment in which we compared the applicability of three different eye tracking devices in psychology of programming research [7]. The purpose of this experiment was to collect information about the use of eye tracking in general and to test the applicability of this method to psychology of programming research.

The first experiments in which we study the effect of different visual appearances of PlanAni to the creation of program knowledge is scheduled to autumn 2004.

4. Automatic Detection of Variable Roles

In previous sections we have stated that variable roles are cognitive constructs, which represent programming knowledge. The automatic analysis of variable roles processes computer programs and tries to assign roles to the variables. The purpose of the automatic analysis is to find a connection between the program code and the cognitive structures of the programmer. The fact that human cogni-

tion is not exact whereas program code is exact makes this task a challenging one.

4.1 Objectives

Our main goal is to find a way to automatically detect variable roles from programs written in procedural languages with context-free grammar. The primary objective can be divided into two sub-goals: gathering variable characteristics from a source program, and mapping a set of variable characteristics to a certain variable role.

The first sub-goal requires customization of compiler and data-flow analysis methods and techniques [1, 8] to extract information about how variables are used in programs.

The second sub-goal includes the creation of a database, which includes mapping information between roles and data-flow analysis derived variable characteristics. This mapping can be done by letting the automatic role analyst to extract information from programs, for which human analysts have assigned variable roles to. Then the characteristics and human defined role information are combined. This is a kind of a machine learning task.

4.2 Research Work And Progress

Roles must be defined in computer terms when detecting variable roles automatically. However, roles are cognitive constructs. As such they may not have strict characteristics but rather a loose definition of behavior - which may differ somewhat from person to person. We are experimenting with defining roles as a set of constraints set upon a set of data-flow analysis derived variable characteristics.

We are implementing an initial prototype of the automatic role analyzer. The initial prototype will analyze Pas-

cal programs and deal with only three roles: fixed value, stepper, and most-recent holder. These three roles covered the majority of variables (81.7 - 90.3 %) in a study of three Pascal programming textbooks [12]. The implementation is done with TclTk and based on Yeti [10], a Yacc-like compiler-compiler [1].

Acknowledgments

This work was supported by the Academy of Finland under grant number 206574.

References

- [1] A. V. Aho, R. Sethi, and J. D. Ullman. *Compilers, Principles, Techniques, and Tools*. Addison-Wesley Publishing Company, 1988.
- [2] M. Ben-Ari and J. Sajaniemi. Roles of Variables as Seen by CS Educators. Accepted to the 9th Annual Conference on Innovation and Technology in Computer Science Education (ITiCSE 2004).
- [3] P. Byckling, M. Kuittinen, S. Nevalainen, and J. Sajaniemi. An Inter-rater Reliability Analysis of Good's Program Summary Analysis Scheme. Accepted to the 16th Annual Psychology of Programming Interest Group Workshop (PPIG'04).
- [4] J. Good. *Programming Paradigms, Information Types and Graphical Representations: Empirical Investigations of Novice Program Comprehension*. PhD thesis, University of Edinburgh, 1999.
- [5] C. D. Hundhausen, S. A. Douglas, and J. T. Stasko. A meta-study of algorithm visualization effectiveness. *Journal of Visual Languages and Computing*, 13:259–290, 2002.
- [6] P. Mulholland. A principled approach to the evaluation of SV: A case study in Prolog. In J. Stasko, J. Domingue, M. H. Brown, and B. A. Price, editors, *Software Visualization - Programming as a Multimedia Experience*, pages 439–451. The MIT Press, 1998.
- [7] S. Nevalainen and J. Sajaniemi. Comparison of Three Eye Tracking Devices in Psychology of Programming Research. Accepted to the 16th Annual Psychology of Programming Interest Group Workshop (PPIG'04).
- [8] F. Nielsen, H. R. Nielsen, and C. Hankin. *Principles of Program Analysis*. Springer-Verlag, Heidelberg, 1998.
- [9] M. Petre and A. F. Blackwell. Mental imagery in program design and visual programming. *International Journal of Human-Computer Studies*, 51(1):7–30, 1999.
- [10] F. Pilhofer. YETI - Yet another Tcl Interpreter. Internet WWW-page, URL: <http://www.fpx.de/fp/Software/Yeti/>, 2002. (March, 2004).
- [11] R. S. Rist. Schema Creation in Programming. *Cognitive Science*, 13:389–414, 1989.
- [12] J. Sajaniemi. An Empirical Analysis of Roles of Variables in Novice-Level Procedural Programs. In *Proceedings of IEEE 2002 Symposia on Human Centric Computing Languages and Environments (HCC'02)*, pages 37–39. IEEE Computer Society, 2002.
- [13] J. Sajaniemi. The Roles of Variables Home Page. Internet WWW-page, URL: http://www.cs.joensuu.fi/~saja/var_roles/, 2003. (March, 2002).
- [14] J. Sajaniemi and M. Kuittinen. An Experiment on Using Roles of Variables in Teaching Introductory Programming. *Computer Science Education*, in press.
- [15] J. Sajaniemi and M. Kuittinen. Program Animation Based on the Roles of Variables. In *Proceedings of the ACM 2003 Symposium on Software Visualization (SoftVis 2003)*, pages 7–16. Association for Computing Machinery, 2003.

Emootiot esille!

Marika Tähti, Leena Arhippainen

Oulun yliopisto, Tietojenkäsittelytieteiden laitos

{marika.tahti, leena.arhippainen}@oulu.fi

1. Johdanto

Nykyisin tuotteiden kehittäminen on yhä käyttäjälähtoisempää eli käyttäjän tarpeet ja toiveet huomioidaan suunnittelussa. Myös tuotteen käytettävyyteen panostetaan entistä enemmän. Kuitenkaan nykypäivän vaativille kuluttajille pelkkä käytettävyys ei enää riitä vaan tuotteen täytyy tarjota kokemuksia ja nautintoja [11]. Käyttäjäkokemustutkimuksessa keskitytään laajemmin käyttäjän tarpeiden ja toiveiden selvittämiseen. Käyttäjän ja tuotteen vuorovaikutuksessa muodostuvaan kokemukseen vaikuttavat muun muassa käyttäjän aiemmat kokemukset, oletukset ja emootiot. Myös tuotteen ominaisuudet ja käyttöympäristö vaikuttavat käyttäjälle syntyviin emootioihin ja kokemuksiin. Etenkin kannettavien (mobile) ja mukautuvien (adaptive) sovellusten kehittäminen tuo uusia piirteitä käyttäjäkokeuksen ja emootioiden tutkimiseen muun muassa vaihtuvan ympäristön ja kenttätestiasetelmien vuoksi.

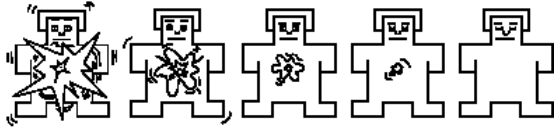


Yleisesti emootiolla tarkoitetaan käyttäjän omakohtaisia tunteita. Termiä käytetään usein, kun puhutaan ihmisen käyttäytymisestä, jolle ei ole riittävää ulkoista syytä tai tarkoitusta [10]. Vaikka emootio yleisesti ymmärretäänkin, ei sille ole löydetty yhtä yksiselitteistä tieteellistä määritelmää. Esimerkiksi Cabanac [5] esittää Kleingin ja Kleingin tutkimuksen, johon on listattu 92 eri määritelmää emootioille. Leventhal [14] määrittelee emotion kokemukseksi, joka on todellinen, mutta jota ei voida suoraan havaita.

Aluksi esittelemme lyhyesti olemassa olevia emootioiden mittaamenetelmiä. Kerromme myös, mitä ongelmia olemme havainneet nykyisissä sanattomissa itseraportointimenetelmissä. Lopuksi esittelemme kehitysvaiheessa olevan emootioiden ja kokemusten keräämiseen tarkoitetun 3E-menetelmän (Expressing Emotions & Experiences).

2. Emootioiden mittausmenetelmät

Emootioita on perinteisesti mitattu psykologian ja sosiologian aloilla. Viime aikoina kiinnostus emootioihin on kuitenkin lisääntynyt muun muassa markkinoinnissa ja teollisessa suunnittelussa. [8] Myös tietotekniikassa tutkitaan emootioita, esimerkiksi tunne-elektronikan (Affective Computing) [20] alueella, missä kehitetään tietokoneita, jotka osaavat paremmin tunnistaa käyttäjän käytöksen ja mukauttaa toimintansa siihen [8]. Esimerkkejä tunne-elektronikassa käytetyistä tekniikoista ovat mm. eleiden ja ilmeiden tunnistus, puheen analysointi sekä elektrofysiologiset mittaukset, kuten kasvojen lihassähkötutkimukset. [17]

Emootioiden tutkiminen ja hyödyntäminen käyttöliittymien suunnittelussa on tullut tärkeäksi myös HCI-alan (Human Computer Interaction) ammattilaisten keskuudessa [16]

Menetelmä	Kuvaotos tai animaatio	Selite
SAM		Vireystila (Arousal) Asteikolla innostunut - unelias
Emocards		Innostuneen neutraali (Excited neutral)
PrEmo	Animaatiossa kädet nousevat vartalon vierestä hieman korkealle. Äännähdys: ”Aaah” 	Pettymys (Disappointment)

Kuva 1. Esimerkkiotokset SAM-, Emocards- ja PrEmo-itseraportointimenetelmistä.

Emootioita mittaavat menetelmät vaihtelevat kynä-paperi -menetelmistä huipputeknologisiin mittauslaitteisiin. Nämä menetelmät mittaavat vain yhtä emootioiden osa-aluetta kerrallaan, kuten erilaisia ilmaisuja, fysiologisia reaktioita ja omakohtaisia tuntemuksia. [8]

Ilmaisut ja niiden tutkimusmenetelmät voidaan jakaa kahteen pääryhmään: kasvojen ilmeet [9], [12] ja ääni-ilmaisut [15], [19]. Mitattaessa kasvojen ilmeitä, tutkimuksen kohteena ovat kasvat ja ilmeiden muuttuminen. Ääni-ilmaisuja tutkittaessa kiinnostus on äänen ja puheen muutoksissa. [8]

Emootioihin liittyy paljon erilaisia fysiologisia reaktioita, joita voidaan mitata muun muassa iholta [3], pupilleista [17] ja sydämenlyönneistä [4].

Omakohtaisia tuntemuksia voidaan mitata sanallisilla, sanattomilla ja jatkuvilla [1] itseraportointimenetelmillä. Esimerkkejä sanallisista menetelmistä ovat muun muassa erilaiset haastattelut, kyselyt ja päiväkirjat [18] sekä ESM-menetelmä (Experience Sampling Method) [6], joka yhdistää haastattelun, kyselyn sekä päiväkirjan. Sanattomia itseraportointimenetelmiä ovat muun muassa (Kuva 1):

- SAM (Self-Assessment Manikin) [13], jossa käyttäjä valitsee ihmiskuvasarjoista kolme asiaa: tyytyväisyys, vireystila, hallinnantunne.
- Emocards [7] koostuu 16 sarjakuvakasvosta (8 naista, 8 miestä), joista käyttäjä valitsee tuntemustaan kuvaavan ilmeen.

Sanallisen ja sanattoman ilmaisun yhdistelmämenetelmä on PrEmo (Product Emotion Measurement) [8], joka koostuu 14 animaatiokuvasta ja niihin liitetystä äännähdyksistä.

3. 3E-menetelmän kehitys

Kannettavien laitteiden käytön lisääntyessä ja mukautuvien sovellusten kehittyessä muuttuva käyttöympäristö tuo haasteita käyttäjäkokemuksen ja emootioiden tutkimiseen. Käytettävien menetelmien tulisi olla helppokäyttöisiä ja soveltua kenttätestaukseen sekä samalla mahdollistaa pitkäaikaisen käytön tutkiminen. Käytettävien tutkimusmenetelmien mahdollinen vaikutus käyttäjäkokemukseen tulisi minimoida. Käyttäjään kiinnitettävät mittauslaitteet voivat vääristää testituloksia, esimerkiksi laite voi aiheuttaa fyysisistä rasitetta tai jännitystä testitilanteessa, millä puolestaan voi olla erilaisia vaikutuksia käyttäjäkokemuksen syntyyn.

Usein ihmisten on vaikea ilmaista tunteitaan sanallisesti [8]. Havaitimme tämän myös käyttäjäkokemustestien yhteydessä, jolloin haastattelimme ja havainnoimme käyttäjiä [2]. Haastatteluissa käyttäjillä ilmeni vaikeuksia kuvata tunteitaan sanallisessa muodossa. Havainnoiteja oli myös vaikea tulkita yksiselitteisesti, sillä käyttäjän ilmeet ja sanalliset ilmaisut olivat hyvin ristiriitaisia. Näimmekin tärkeäksi tehtäväksi löytää emootioiden keräämiseen soveltuva menetelmä kannettavien ja mukautuvien sovellusten käyttäjäkokemuksen tutkimiseksi. Koska menetelmän tuli olla helppokäyttöinen ja tukea emootioiden ilmaisemista, valitsimme käyttäjäko-

kemustutkimuksiimme sanattomat itseraportointimenetelmät (Emocards ja SAM).

Emocards-menetelmää sovelsimme kannettavan laitteen kenttäkokeissa, joissa Emocards-vastauksia saimme 61 testihenkilöltä. Havaitimme, että menetelmä on kyllä toimiva, mutta sarjakuvakasvon välittämä tunne voi jäädä käyttäjälle epäselväksi.

SAM-menetelmää sovelsimme käyttäjäkokemusaineiston (N=6) keräämisessä paikkatietoon mukautuvasta kannettavasta sovelluksesta. Tässäkin menetelmässä käyttäjillä ilmeni vaikeuksia valita tunnetaan kuvaava vaihtoehto.

Käyttäjäkokemustestiemme perusteella voimme todeta, että olemassa olevilla sanattomilla menetelmillä on hyvin vaikeaa saada selville ihmisten todellisia emootioita ja niihin liittyviä kokemuksia.

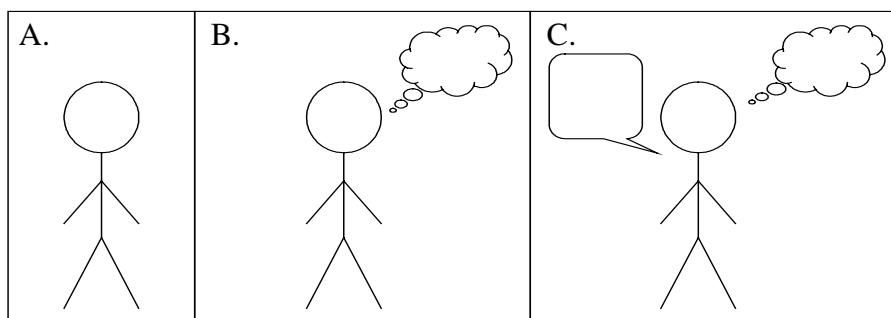
Edellä esitetyissä menetelmissä (Kuva 1) annetaan käyttäjälle valmiit kuvat, joista hän valitsee tuntemustaan vastaavan. 3E-menetelmässä tutkimme emootioiden ilmaisemista toisesta näkökulmasta, eli emme anna käyttäjälle valmista tunnetta tai kuvaa, vaan hän saa itse ilmaista tuntemuksiaan piirtämällä ja kirjoittamalla.

Aluksi hahmotelimme paperille tyhjän pallon kuvaamaan ihmisen päätä. Koska

tämä pallo ei tuntunut riittävän kuvaavalta, lisäsimme myös pelkistetyn vartalon, jotta käyttäjät hahmottaisivat, että kyse on ihmisestä, käyttäjästä itsestään (Kuva 2 A). Pelkkä ihmishahmo käyttäjän lisäämine ilmeineen on vaikea tulkita, sillä se ei ilmaise riittävästi yhteyttä käyttäjän tunteiden ja niiden syntyvän välillä.

Emootioiden tulkinnan tueksi lisäsimme kuvaan ajatuskuplan, johon käyttäjä voisi ilmaista ajatuksensa liittyen tunteeseen, jota ilme kuvastaa (Kuva 2 B). Puhekuplat mahdollistavat sanallisen kuvauksen tuntemuksista, joten lisäsimme kuvaan tyhjän puhekuplan käyttäjäkokemustiedon keräämiseksi (Kuva 2 C). Uskomme, että näin saamme kerättyä tunteita ilmepiirustuksesta sekä tunteisiin liittyviä kokemuksia ajatus- ja puhekuplista. Teimme pilotitestin kehitteillä olevan menetelmän kolmelle eri versiolle (Kuva 2).

Viiden otoksen testi osoitti, että ihmiset vastaavat eri tavalla kuhunkin kuvaan. Etenkin versioiden A ja C välillä oli eroja. Esimerkiksi kuvan C-versioon kukaan ei piirtänyt pään kohdalle mitään, vaan kaikki kirjoittivat vain kupliin. Sen sijaan, versiolle A oli piirretty ilmeet ja kirjoitettu joitakin ajatuksia. Jatkossa testaamme versioita A ja C laajemmalla testijoukolla ja sen myötä kehitämme menetelmää eteenpäin.



Kuva 2. 3E-menetelmän kehitysvaiheita (A-, B- ja C-versiot).

4. Yhteenveto

Alussa motivoimme emootiotutkimuksen tarvetta tietojenkäsittelytieteiden näkökulmasta, erityisesti kannettavien ja mukautuvien laitteiden kehityksessä. Esittelemme myös olemassa olevia emootioiden

mittausmenetelmiä ja havaitsemiamme puutteita sanattomissa itseraportointimenetelmissä. Lopussa kerromme myös uudesta kehitysvaiheesta olevasta 3E-menetelmästä, jota kehitämme tuodaksemme kattavan ja joustavan lähestymis-

tavan emootioiden ja kokemusten tutkimiseen.

Lähteet

- [1] P.V. Abeele ja Maclachlan. Process Tracing of Emotional Responses to TV Ads: Revisiting the Warmth Monitor. *Journal of Consumer Research*, 20 (4), s. 586-600. 1994.
- [2] L. Arhippainen ja M. Tähti. Empirical Evaluation of User Experience in Two Adaptive Mobile Application Prototypes. *MUM 2003*, s. 27-34. 2003.
- [3] W. Ark, D.C. Dryer ja D.J. Lu. The Emotion Mouse. *HCI international '99*. Munich Germany, August. 1999.
- [4] J.F. Brosschot ja J.F. Thayer. Heart rate response is longer after negative emotions than after positive emotions, *International Journal of Psychophysiology*, 50 (3), s. 181-187. 2003.
- [5] M. Cabanac. What is emotion? *Behavioural Processes*, 60 (2), 69-83. 2002.
- [6] M. Csikszentmihalyi ja R. Larson. Validity and Reliability of the Experience-Sampling Method. *The Journal of Nervous and Mental Disease*, 175 (9), s. 526-536. 1987.
- [7] P.M.A. Desmet, C.J. Overbeeke ja S.J.E.T. Tax. Designing Products with Added Emotional Value; Development and Application of an Approach for Research through Design. *The Design Journal*, 4(1), s. 32-47. 2001.
- [8] P. Desmet. *Designing Emotions*. Väitöskirja. Delft University of Technology. 2002.
- [9] P. Ekman ja W.V. Friesen. *Facial Action Coding System*. Consulting Psychologists Press, Palo Alto, California. 1978.
- [10] N.H. Frijda. *The Emotions*. Cambridge University Press. 1986.
- [11] P.W. Jordan. *Designing Pleasurable Products, An Introduction to the New Human Factors*, Taylor & Francis, New York. 2000.
- [12] S. Kaiser ja Wehrle. Emotion research and AI: Some theoretical and technical issues. *Geneva Studies in Emotion and Communication*, 8 (2), s. 1-16. 1994.
- [13] P.J. Lang. *Behavioral Treatment and Bio-behavioral Assessment: Computer Applications*. Teoksessa Sidowski, J.B., Johnson, J.H. ja Williams, T.A. (Eds.) *Technology in Mental Health Care Delivery Systems*. Norwood. NJ: Albex, s. 119-139. 1980.
- [14] H.A. Leventhal. *Perceptual Motor Theory of Emotion*. Teoksessa Scherer, K. ja Ekman, P. Lawrence (Eds.) *Approaches to Emotion*. Erlbaum Associates, USA, s. 271-291. 1984.
- [15] D. Litman ja K. Forbes. Recognizing emotions from student speech in tutoring dialogues. *Automatic Speech Recognition and Understanding Workshop (ASRU 2003)*. 2003.
- [16] A. Marcus. The Emotion Commotion. *Interactions*, 10(6), s. 28-34. 2003.
- [17] T. Partala, M. Jokiniemi ja V. Surakka. Pupillary Responses to Emotionally Proactive Stimuli. *Eye Tracking Research & Applications Symposium*, s. 123-129. 2000.
- [18] J.P. Robinson. Microbehavioural Approaches to Monitoring Human Experience. *The Journal of Nervous and Mental Disease*, 175 (9), s. 514-518. 1987.
- [19] J. Toivanen, T. Seppänen ja E. Väyrynen. Automatic recognition of emotions in spoken Finnish: preliminary results and applications. *Prosodic Interfaces 2003*, s. 85-89. 2003.
- [20] J. Vanhala, J. Rantanen ja M. Iltanen. *Apostolinkyydillä tietoverkoissa, NetWalk*. Proessori, marraskuu, s. 43-48. 1999.

Neural Network Modelling of Speech Production and its Problems

Antti Järvelin¹

University of Tampere, Department of Computer Sciences,
antti.jarvelin@cs.uta.fi

Abstract. Naming errors and word finding difficulties occur commonly as slips of the tongue in healthy subjects, but even more often in patients suffering from brain damage. Indeed, these errors are common concomitant of aphasia (a language disorder caused by brain damage) and dementia patients. Such errors are of theoretical interest, since they may reveal the inner structure of the language production system. Computational models have been applied to study the dynamics and causes of this disorder. Current paper briefly describes a connectionist model "Learning Slipnet" for the simulation of word production errors in the Finnish language. First the architecture of the model is described. Then the model is shown to be able to simulate word production errors of heterogeneous group of aphasic and dementia patients by briefly presenting the simulation results of selected patient cases.

Keywords and phrases:

Neural network modelling, word production problems, aphasia, dementia.

1. Introduction

Word-finding difficulty (anomia) is the most common linguistic deficit in aphasia and dementia. This difficulty is often investigated by a picture naming task, because naming a picture includes all major processing stages of word production: activation of a target concept, retrieval of lexical-semantic information on that concept, retrieval of the corresponding word

form, and finally articulation. The errors made during the picture naming task are of particular interest, since they might reveal the underlying reasons of a patient's anomia. Common error types are e.g. semantic errors (giraffe → Africa), formal errors (giraffe → carafe), neologistic (non-word) errors (giraffe → arff) and omission errors (patient says nothing or says "I don't know" etc.).

Excluding the visual perception of a picture and the articulation of the corresponding word, naming of a picture can be abstracted to the process of *lexicalization*. Lexicalization is a two-stage process [1, 3] in which a speaker translates a semantic representation (meaning) of a target word into its phonological representation (sound structure). In the first phase an intermediate form of the word (lemma) is retrieved. Lemma is specified by syntactic and semantic properties of the target word but does not include any phonological information about the target. During the second phase the retrieved lemma is translated into the corresponding phonological word form (lexeme).

Anomic naming errors have extensively been studied by neural network models of lexicalization (see e.g. [2, 3, 8, 10]). Such models are needed to better understand the underlying word production system and, in particular, how the system should be organized in order to give rise to a specific symptomatology during malfunction. A common feature of these models is that they are static, i.e., non-learning. This means that certain prop-

¹Work supported by Tampere Graduate School in Information Science and Engineering (TISE).

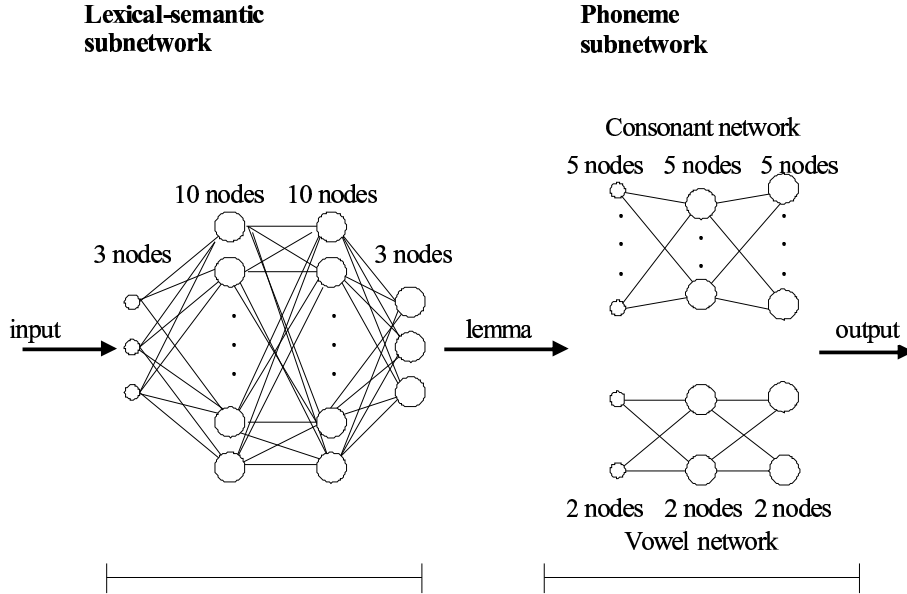


Figure 1: The architecture of the lexicalization model: lexical-semantic network and two phoneme networks.

erties, like word learning in children or re-acquisition of linguistic abilities after brain damage, cannot be modelled with the models. Therefore learning models of lexicalization should be developed and studied.

This paper reviews our suggestion for a learning model of lexicalization called "Learning Slipnet" and briefly presents some simulation results. The main difference between our model and the previous models of lexicalization is the fact that it is implemented using Multi-Layer-Perceptron (MLP) neural networks, which renders the model learning. Detailed description of the model and performed simulations is available in [5–7].

2. The model

The architecture of Learning Slipnet is shown in Figure 1. An input for the model is a semantic representation of a target word. The input is passed to the lexical-semantic subnetwork, which simulates lemma access. The result of the lexical-semantic subnetwork (lemma) is an input to the phoneme subnetwork, which

sequentially produces the corresponding phonemes one at a time either in consonant or vowel network. The phoneme subnetwork was divided into two separate networks, because even in disordered speech there exist no consonant–vowel substitutions. All three MLP neural networks of the model (lexical-semantic subnetwork and two phoneme networks) are autoassociative, i.e., their target output equals to their input.

To simulate disturbed word production a random noise parameter αe_{ij} was added to every weight w_{ij} between the nodes i and j in the network. The parameter e_{ij} is derived from normal distribution $N(0,1)$ and α is a positive scaling factor. The total input v_j for node j is calculated with

$$v_j = \sum_{i=0}^n (w_{ij} + \alpha e_{ij}) x_i, \quad (1)$$

where x_i is an output of node i connected to the node j and n is the number of the nodes in the current layer ($x_0 = 1$ is a bias scalar). Otherwise standard computation rules and equations of MLP neural networks were applied to calculate the results of the subnetworks (see e.g. [4]). A separate scaling factor was used for

Patient	Correct %		Semantic %		Omission %		Other %	
	P	M	P	M	P	M	P	M
A	51.7	53.3	4.2	4.4	44.1	42.4	0.0	0.0
B	77.1	77.5	3.0	2.5	19.3	19.4	0.6	0.6
C	65.7	66.4	3.6	3.2	7.2	7.6	23.5	22.8
W	42.9	44.2	13.9	13.9	21.1	21.0	22.2	20.9
AD	37.8	38.7	53.3	53.0	0.0	0.2	8.9	8.1
VaD	54.1	56.5	35.1	35.5	0.0	0.1	10.8	8.0

Table 1: Example cases of simulated naming error data. P corresponds to patient’s and M model’s performance.

both subnetworks (α_L for lexical-semantic and α_P for phoneme subnetwork) to allow the simulations of mainly semantically or phonologically based disorders.

Since during picture naming tests patients do not necessarily provide answers for every picture (i.e. they ”produce” omissions), a threshold value τ was added between the subnetworks. The threshold τ discards output vectors \mathbf{o} too far from any word taught to it. Let \mathbf{n} be the nearest vector (in Euclidean sense) to the output \mathbf{o} taught to the model and d an Euclidean distance between vectors \mathbf{n} and \mathbf{o} . The thresholded result \mathbf{r} of the lexical-semantic network is calculated with

$$\mathbf{r} = \begin{cases} \mathbf{n}, & \text{if } d = 0 \\ \mathbf{n}, & \text{if } d > 0 \text{ and } \tau \leq \frac{1}{c \cdot d} \\ \text{otherwise undefined} \end{cases}, \quad (2)$$

where c is a scaling factor. During the simulations we set $c = 100$.

The result of the model is calculated by first calculating the result of the lexical-semantic subnetwork. This result is thresholded with (2). If the output vector \mathbf{r} is undefined the result of the model is an omission. Otherwise the phoneme subnetwork sequentially produces the phonemes corresponding to the vector \mathbf{r} . The concatenation of the produced phonemes is the result of the model. This result can be either correct word or an error. The result was classified as a semantic error if the output of the lexical-semantic subnetwork differed from its input and the result of the phoneme subnetwork corresponded

to the result of the lexical-semantic subnetwork. If the result of the phoneme subnetwork differed from the result of the lexical-semantic subnetwork, the output was classified as other error. Other errors therefore included pure phonological errors, i.e., errors that occurred in the phoneme subnetwork and so called *mixed errors*, where errors occurred in both the networks.

3. Simulation results

The model was tested by fitting it to the naming data of 10 aphasic and 22 dementia patients described in [8, 9]. The results of these simulations are briefly addressed here. Table 1 shows the selected patient cases from the patient data set and corresponding performance of the model.

Cases A, B, C, and W are patients with aphasia and cases AD and VaD are patients with dementia. Main difficulties of cases A (anomic aphasic) and B (Broca’s aphasic) were omissions. With case A (44.1 % omissions) the retrieval deficit was considerably worse than with B (19.3 % omissions). Cases C (conduction aphasic) and W (Wernicke’s aphasic) had biggest problems with phonological processing which is exemplified by high rates of other errors (23.5 % and 22.2 %). Case W had also high rates of omissions and semantic errors. Naming performances of dementia patient cases AD (probable Alzheimer’s disease) and VaD (vascular

disease) were characterized by semantic errors (53.3 % and 35.1 %).

As Table 1 suggests, the model was successful in simulating the different naming patterns of this heterogeneous group of patients. Examples of the simulation parameters used in the simulations are presented in Table 2.

Patient	α_L	τ	α_P
A	0.005	0.253	0.000
B	0.004	0.216	0.188
C	0.004	0.162	0.418
W	0.008	0.127	0.437
AD	0.016	0.010	0.295
VaD	0.011	0.012	0.291

Table 2: Examples of the simulation parameters found for the patients.

4. Conclusion

In this paper we presented a model for simulating the naming errors of aphasic and dementia patients. The simulation results briefly highlighted in the previous section suggested that the model was successful in simulating the naming errors of this heterogeneous group of patients. However, in the future the error classification of the model has to be improved, because current classification can be criticized to be too vague to provide reliable results. Furthermore the model should be developed so that the rehabilitation studies could be carried out. This may result major changes to the current model, e.g. changing the underlying perceptron based architecture to some other neural network type.

References

[1] D. Chilant, A. Costa, and A. Caramazza. Models of naming. In A. E. Hillis, editor, *The Handbook of Adult Language Disorders*, pages 123–142. Psychology Press, New York, 2002.

[2] G. S. Dell, M. F. Schwartz, N. Martin, E. M. Saffran, and D. A. Gagnon. Lexical access in aphasic and nonaphasic speakers. *Psychological Review*, 104(4):801–838, 1997.

[3] M. Goldrick and B. Rapp. A restricted interaction account (RIA) of spoken word production: the best of both worlds. *Aphasiology*, 16(1/2):460–499, 2002.

[4] S. Haykin. *Neural Networks. A Comprehensive Foundation*. Prentice Hall, 1994.

[5] A. Järvelin, M. Juhola, and M. Laine. A neural network model of lexicalization for simulating the anomie naming errors of dementia patients. To appear in the proceedings of the 11th *World Congress on Medical Informatics 2004*. In press.

[6] A. Järvelin, M. Juhola, and M. Laine. Neural network modelling of word production in finnish: coding semantic and non-semantic features. Submitted to *Connection Science*.

[7] A. Järvelin, M. Juhola, and M. Laine. A novel neural network model for the simulation of word production errors of finnish nouns. Submitted to *Computer Speech and Language*.

[8] M. Laine, A. Tikkala, and M. Juhola. Modelling anomia by the discrete two-stage word production architecture. *Journal of Neurolinguistics*, 11(3):275–294, 1998.

[9] M. Laine, E. Vuorinen, and J. O. Rinne. Picture naming deficits in vascular dementia and alzheimer’s disease. *Journal of Clinical and Experimental Neuropsychology*, 19(1):126–140, 1997.

[10] W. J. M. Levelt, A. Roelofs, and A. S. Meyer. A theory of lexical access in speech production. *Behavioral and Brain Sciences*, 22(1):1–38, 1999.

A Comparison of Heterogeneous Proximity Functions and Euclidean Distance

Janne Lumijärvi

University of Tampere, Department of Computer Sciences

janne.lumijarvi@uta.fi

1. Introduction

Proximity functions [1-5] are an important component of many statistical and machine learning methods. These functions are typically applied to compute distances or similarities between pairs of objects (cases or examples). Methods that employ proximities include many clustering [1-3] and instance-based learning algorithms [4] for grouping and classification of objects such as patient cases. The choice of a proximity function is of utmost importance, because an unsuitable function may dramatically degrade the performance of a method that operates on proximities.

This work considers the use of different proximity functions in the classification of heterogeneous data, i.e. data described both with nominal and quantitative (ordinal, interval, and ratio) scaled variables. The scale [2,5] of a variable indicates the information available on relations between variable values. Since nominal values have no meaningful order, nominal scale carries less information than quantitative scales, whose values can be arranged. Of a pair of unordered values one can only observe whether values are equal: A physician may compare two cases, for example, by examining whether the location of tinnitus is the left ear in both cases.

Heterogeneous data is problematic for proximity functions assuming the same scale for all the variables. Application of the well-known Euclidean distance function to nominal values is questionable, because arithmetical operations for the unordered values are meaningless. As an example, consider values “no tinnitus”, “left”, “right”, and “bilateral” with scores 0, 1, 2, and 3. The Euclidean distance would rate “right” twice as distant from “no tinnitus” than “left”, which is not reasonable from the medical point of view.

When a proximity measure is to be applied to the heterogeneous data, there are basically three approaches. Firstly, the nature of the data may simply be ignored. Soundness of results largely depends on the method and the task, but problems described above are likely to emerge.

Secondly, variables can be transformed to meet scale assumptions. If the nominal scale is the simplest in the data, the categorization of interval and ratio scaled variables is needed. Deciding the number of categories is difficult, and when categories are treated as nominal the order information is lost. Transformations from a lower to a higher scale are impossible, but a k -valued nominal variable can be coded as k binary or $k-1$ dummy variables [6] for which arithmetical operations are legal. The obvious drawback of dummy coding is the increased data dimensionality.

Thirdly, one can analyze the heterogeneous data as it is by applying a heterogeneous proximity function that is able to handle different scales. This approach requires no transformations thus avoiding the loss of information, as well as the increase of dimensionality.

In the following, six proximity functions are compared with two aims in mind. Five heterogeneous functions were weighed against the Euclidean distance to assess whether heterogeneous functions are better than a function assuming the same scale. In addition, the performance of typical value-matching-based heterogeneous functions with respect to the Heterogeneous Value Difference Metric (HVDM) function [5] of Wilson and Martinez is studied. HVDM utilizes class information, which has shown to be a competitive approach [5]. This function has been found useful in classification tasks involving medical data [7,8]. However, HVDM can only be applied when class information is available. Perform-

ance of the functions was evaluated using a nearest neighbor classifier and a collection of heterogeneous medical data sets.

2. Methods

2.1 Nearest neighbor classification

Nearest neighbor technique is a classic instance-based machine learning method [4] which classifies the unseen examples in a testing set T into C mutually exclusive classes on the basis of the k nearest examples in a learning set L . Each test example is labelled as belonging to the most frequent class of its neighborhood. The proximity functions were evaluated with a three-nearest neighbor classifier (3-NN), which is less sensitive to noise than the common 1-NN classifier [4].

2.2 Proximity functions

Function $f: E \times E \rightarrow R$ is a proximity function in a space E , if for every example $x, y \in E$, there exists a lower or upper bound f_0 for f so that $f(x,x) = f_0$ and $f(x,y) = f(y,x)$ [3]. Next, the proximity, i.e. distance and similarity functions evaluated in this study are shortly described.

Heterogeneous functions were compared to probably the most well known proximity measure – the Euclidean distance [1-5], a variant of the general Minkowski function. The Euclidean distance (EUCL) is defined as:

$$D_2(x, y) = \sqrt{\sum_{i=1}^m (x_i - y_i)^2}, \quad (1)$$

where m is the number of variables and the value of the i th variable for an example x is denoted x_i ($1 \leq i \leq m$). In these experiments, each quantitative variable was normalized by dividing it by four standard deviations ($4\sigma_i$) as in [5]. Also, every nominal variable was similarly normalized to make the Euclidean distance wholly insensitive to scales.

Heterogeneous proximity functions

Aha's Heterogeneous Euclidean-Overlap Metric [5] (HEOM)

$$HEOM(x, y) = \sqrt{\sum_{i=1}^m h_i(x_i, y_i)^2} \quad (2)$$

is a distance function that treats nominal and quantitative variables differently:

$$h_i(a, b) = \begin{cases} 1, & \text{if } a \text{ or } b \text{ is missing,} \\ lb(a, b), & \text{if } i\text{th variable is nominal,} \\ |a - b|/rng_i, & \text{if } i\text{th variable is quantitative,} \end{cases} \quad (3)$$

where rng_i is the range of the i th variable and lb is an overlap function

$$lb(a, b) = \begin{cases} 0, & \text{if } a = b, \\ 1, & \text{otherwise.} \end{cases} \quad (4)$$

Gower's similarity function [1] (GOW) is defined as:

$$GOWER(x, y) = \frac{\sum_{i=1}^m g_i(x_i, y_i)}{\sum_{i=1}^m lb(x_i, y_i)}, \quad (5)$$

where g is

$$g_i(a, b) = \begin{cases} 0, & \text{if } a \text{ or } b \text{ missing,} \\ 1 - lb(a, b), & \text{if } i\text{th variable nominal,} \\ 1 - |a - b|/rng_i, & \text{if } i\text{th variable quantitative,} \end{cases} \quad (6)$$

and lb is an indicator function for observed values

$$lb(a, b) = \begin{cases} 1, & \text{if } a \text{ and } b \text{ observed,} \\ 0, & \text{otherwise.} \end{cases} \quad (7)$$

Gower's function differs from HEOM in that the Manhattan distance is computed instead of the Euclidean distance, and values are normalized by definition into range [0,1] by dividing the sum of similarities by the number of observed value pairs. It is assumed that the sum is greater than zero.

Another similarity measure of this study is Estabrook-Rogers similarity function [9] (ER):

$$ER(x, y) = \frac{\sum_{i=1}^m er_i(x_i, y_i)}{m}, \quad (8)$$

where missing values and nominal values of the i th variable are treated as in Gower's function:

$$er_i(a, b) = \begin{cases} 0, & \text{if } a \text{ or } b \text{ is missing,} \\ 1 - lb(a, b), & \text{if variable nominal,} \\ \frac{2u_i + 1 - |a - b|}{2u_i + 2 + |a - b| u_i}, & \text{if variable quantitative and } |a - b| \leq u_i, \\ 1, & \text{otherwise.} \end{cases} \quad (9)$$

The Manhattan distance for quantitative variables has an upper bound u_i , values over which are considered maximally dissimilar. In this study $u_i = rng_i - 2$, which is the largest reasonable value [9].

Ichino-Yaguchi generalized Minkowski metric [10] (GEM) is based on the Cartesian space model and can compute distances for sets and intervals of attribute values by utilizing the Cartesian join and meet operators. Below, GEM is very briefly described in its Euclidean data point form with parameter value $\gamma = 0.5$:

$$GEM(x, y) = \sqrt{\sum_{i=1}^m gem_i(x_i, y_i)^2}, \quad (10)$$

and

$$gem(a, b) = \begin{cases} 0, & \text{if } a = b, \\ 1/dm_i, & \text{if } i\text{th variable nominal,} \\ |a - b|/dm_i, & \text{if } i\text{th variable discrete,} \\ |a - b|/rng_i, & \text{if } i\text{th variable continuous,} \end{cases} \quad (11)$$

where dm_i is the size of value domain of the i th variable. In contrast to the other heterogeneous functions, GEM normalizes the difference in a nominal variable with the size of value domain.

Also, Heterogeneous Value Difference Metric [5] (HVDM) has a main function that resembles the Euclidean distance:

$$HVDM(x, y) = \sqrt{\sum_{i=1}^m hvdm_i(x_i, y_i)^2}. \quad (12)$$

Treatment of nominal variables differs greatly from that of the heterogeneous functions described above. Instead of simple value matching, HVDM makes use of the class information to compute conditional probabilities. Function $hvdm_i$ is evaluated for the nominal values with the normalized and simplified Value Difference Metric [5] defined as:

$$vdm_i(x_i, y_i) = \sqrt{\sum_{c=1}^C |N_{i, x, c}/N_{i, x} - N_{i, y, c}/N_{i, y}|^2}. \quad (13)$$

In equation (13) $N_{i, x, c}$ is the number of examples in the learning set L with value x_i and class c , $N_{i, x}$ is the number of examples in L that have value x_i , and C denotes the number of classes. Distance $hvdm_i$ for other than nominal attributes is $|x_i - y_i| / 4\sigma_i$. The missing values are processed as in HEOM.

3. Materials

Total of 12 heterogeneous medical data sets were used as test material (see Table 1). These data passed the following criteria:

1. At least one multi-valued nominal variable with statistically significant ($p < 0.05$) dependency on the class variable as measured with Cramér's V [11].
2. Large enough for 10-fold cross-validation.
3. No missing values or few enough of them to allow the completion of the data.

The aids2 [12], depression [13], low birth weight [14], postoperative nausea and vomiting (PONV) [15], and VA lung cancer [12] data sets were complete. Total of 564 and 3 examples were excluded from the aids2 and VA lung cancer data sets, because these cases had been censored before one year and 90 days, respectively. To avoid treatment of missing data in functions, variables having missing values were excluded from the PONV

data. For the same reason, the missing values of acute appendicitis [16], benign breast disease [14], muscular dystrophy carriers [17], prostate cancer [14], vertigo [8], and Cleveland heart disease [18] were imputed with the modes and medians.

4. Results

The six proximity functions were evaluated experimentally using a 3-NN classifier and 10-fold cross-validation. Cross-validation is a technique where the data is divided into k disjoint sets of equal size and each set is used once as the test set and the union of the other $k-1$ sets as the learning set. Prediction accuracy, the ratio of correctly classified test examples to all the test examples, and true-positive rate (TPR), the ratio of correctly classified positive test examples to all the positive test examples, were used to evaluate the nearest neighbor classification.

Data set	N	C	V	B	M	O	Q
Acute appendicitis	13 33	1 3	1 6	9	4	1	2
Aids2	22 97	2	5	2	1	0	2
Benign breast disease	20 0	2	1 1	1	1	2	7
Depression	29 4	2	1 4	7	3	2	2
Heart disease	30 3	2	1 3	3	3	1	6
Low birth weight	18 9	2	7	3	1	0	4
Muscular dystrophy	12 5	2	6	0	1	0	5
PONV (placebo)	14 1	2	1 4	8	4	0	2
PONV (prophylaxis)	16 6	2	1 4	8	4	0	2
Prostate cancer	38 0	2	7	2	1	1	3
Vertigo	91 4	1 0	3 8	1 1	1	1 0	1 6
VA lung cancer	13 7	2	6	2	1	0	3

Table 1 – The data sets used in the experiments. N, C, and V are the numbers of cases, classes, and variables, respectively. B, M, and O refer to the numbers of binary, multi-valued nominal, and ordinal variables. Q is the number of interval and ratio scaled variables.

The results were dependent, because the comparison required the same 10-fold partition to be used with all the proximity functions. Due to the small sample size the two-tailed Wilcoxon signed ranks test [11] was used instead of the paired t test to examine whether differ-

ences between the pairs of proximity functions were significant ($p < 0.05$). Since the Wilcoxon signed ranks test was repeated 15 times, a Bonferroni correction [11] was also made to the probability associated with each test by multiplying it with the number of tests.

Table 2 shows the prediction accuracies of the 3-NN classifier using different proximity functions on the 12 data sets. Medians did not differ much, and neither the original nor the Bonferroni corrected p values of the paired tests were statistically significant.

Data set	GEM	ER	EUC L	GOW	HEOM	HVDM
Acute appendicitis	59	62	60	60	60	62
Aids2	59	56	58	57	58	58
Benign breast disease	74	71	75	75	73	72
Depression	79	83	80	83	82	81
Heart disease	83	78	83	79	79	81
Low birth weight	62	68	65	66	66	67
Muscular dystrophy	89	78	86	86	83	89
PONV (placebo)	92	90	92	89	89	92
PONV (prophylaxis)	90	90	90	91	90	89
Prostate cancer	71	70	73	71	71	74
Vertigo	73	80	75	78	78	79
VA lung cancer	66	69	73	69	68	71
Median	73	74	75	77	76	76

Table 2 – Accuracies (%) of proximity functions in 10-fold cross-validated 3-NN classification.

Table 3 presents the median TPRs of the 3-NN classifier using different proximity functions on the 12 data sets. Medians are quite similar (64-67%), except that of HVDM (73%). The paired tests deemed most differences statistically insignificant. Significant differences were in favor of the HVDM function (HVDM > ER and HVDM > HEOM) at the adjusted α level. An additional significant difference at original α level was HVDM > GOW.

Data set	GEM	ER	EUC L	GOW	HEOM	HVDM
Acute appendicitis	11	12	14	8	8	14
Aids2	57	54	57	56	57	57
Benign breast disease	62	55	62	61	59	60
Depression	52	57	54	57	55	56

Heart disease	83	78	83	78	79	80
Low birth weight	52	57	57	57	56	59
Muscular dystrophy	85	68	76	79	75	84
PONV (placebo)	92	90	92	89	89	92
PONV (prophylaxis)	87	85	85	87	85	86
Prostate cancer	70	68	72	69	69	73
Vertigo	47	67	55	73	77	81
VA lung cancer	66	69	73	69	68	71
Median	64	67	67	67	66	73

Table 3 – Median TPRs (%) of proximity functions in 10-fold cross-validated 3-NN classification.

5. Discussion

Five heterogeneous functions and the Euclidean distance were compared by means of the prediction accuracies and TPRs of the 3-NN classifier. Four of the functions (HEOM, Gower, ER, and GEM) treat nominal values simply by comparing their equality. The fifth heterogeneous function (HVDM) is more sophisticated than the four value-matching-based functions. HVDM function evaluates distances between nominal values using conditional probabilities based on the class information.

The first objective of the comparison was to verify the intuitive hypothesis that heterogeneous functions would be more appropriate for computing proximities in the data sets described with a mixture of nominal and quantitative attributes than functions insensitive to scales.

There were no statistically significant differences in the prediction accuracies and the medians of TPRs of the Euclidean distance and the five heterogeneous functions. The results were clearly not in the expected direction and differed from the results of the previous studies. The Euclidean distance outperformed HEOM in [5] and [19], where it also was better than Gower's function. Earlier, HVDM has been found better than the Euclidean distance [19].

The second objective was to compare the value-matching-based heterogeneous functions with HVDM. The use of the class information is also a weakness, because the HVDM function can be applied only when the classes are known. Therefore, it was of interest to know how the previously unevaluated ER and GEM functions, would perform compared to HVDM.

Results showed, as expected [19], that the HEOM produced lower TPRs than HVDM. Furthermore, at the original less stringent α

level, HVDM outperformed Gower's function as in the previous study [19]. ER was significantly worse than HVDM at the adjusted α level. Although GEM and HVDM did not differ significantly, it is obvious, knowing the similar nature of the value-matching-based functions, that HVDM was better than the other heterogeneous functions of the present study. The ability to make use of the external information is clearly an important factor behind the good relative performance of HVDM.

Explaining why the relative performance of the Euclidean function was better than expected in the earlier [5,19] and present studies is an interesting subject for future study. This requires additional, more controlled experiments. For example, normalization should be as similar as possible in the different functions. Future experiments should include data sets with larger numbers of multi-valued nominal attributes. For this, synthetic data sets must probably be generated. Future work should also include additional classification methods utilizing proximity functions.

The limitation of this study was the small number of data sets. The data was selected more carefully than in the previous studies [5,19], where missing values were allowed to affect the results and the prediction capability of the nominal variables was not considered. Since it was difficult to meet all the requirements, the collection of data sets was not as large as was wished. Accuracies have earlier [5,19] shown significant differences between functions, and it is likely that a larger collection would have produced more significant results.

To summarize, the Euclidean distance was as good as, and the HVDM function better than, the value-matching-based HEOM, Gower, ER, and GEM functions.

Acknowledgments

The author wishes to thank Erkki Pesonen, PhD, for providing the acute appendicitis data. The author acknowledges the financial support by the Academy of Finland.

References

- [1] B. S. Everitt, S. Landau, and M. Leese, *Cluster Analysis*. 4th ed. London: Arnold, 2001.
- [2] S. Sharma, *Applied Multivariate Techniques*. New York: Wiley, 1996.
- [3] J. Boberg, *Cluster Analysis: A Mathematical Approach with Applications to Protein Structures*. Academic dissertation. Turku: Turku Centre for Computer Science, University of Turku, Finland, 1999.
- [4] T. M. Mitchell, *Machine Learning*. New York: McGraw-Hill, 1997.
- [5] D. R. Wilson and T. R. Martinez, Improved heterogeneous distance functions. *J Artif Intell Res* 1997: 6: 1-34.
- [6] A. Agresti, *An Introduction to Categorical Data Analysis*. New York: Wiley, 1996.
- [7] J. Laurikkala, *Knowledge Discovery for Female Urinary Incontinence Expert System*. Academic dissertation. Tampere: Department of Computer Sciences, University of Tampere, Finland, 2001.
- [8] K. Viikki, *Machine Learning on Otonerological Data: Decision Trees for Vertigo Diseases*. Academic dissertation. Tampere: Department of Computer Sciences, University of Tampere, Finland, 2002.
- [9] G. F. Estabrook and D. J. Rogers, A general method of taxonomic description for a computed similarity measure. *BioScience* 1966: 16, Nov: 789-793.
- [10] M. Ichino and H. Yaguchi, Generalized Minkowski metrics for mixed feature-type data analysis. *IEEE Trans Syst Man Cybern* 1994: 24, 4 (Apr): 698-708.
- [11] M. A. Pett, *Nonparametric Statistics for Health Care Research: Statistics for Small Samples and Unusual Distributions*. Thousand Oaks: SAGE Publications, 1997.
- [12] W. N. Venables and B. D. Ripley, *Modern Applied Statistics with S-PLUS*. New York: Springer, 1999.
- [13] A. A. Afifi and V. Clark, *Computer-Aided Multivariate Analysis*. London: Chapman & Hall, 1996.
- [14] D. W. Hosmer and S. Lemeshow, *Applied Logistic Regression*. New York: Wiley, 2000.
- [15] K. Viikki, M. Juhola, I. Pyykkö, and P. Honkavaara, Evaluating training data suitability for decision tree induction. *J Med Syst* 2001: 25: 133-144.
- [16] E. Pesonen, J. Ikonen, M. Juhola, and M. Eskelinen, Parameters for a knowledge base for acute appendicitis. *Methods Inf Med* 1994: 33: 220-226.
- [17] M. Percy, Procedures for the detection of muscular dystrophy carriers. In: D. J. Hand, F. Daly, A. D. Lunn, K. J. McConway, and E. Ostrowski, eds. *A Handbook*

- of Small Data Sets*. London: Chapman & Hall, 1996: 223-228.
- [18] C. L. Blake and C. J. Merz, UCI Repository of machine learning databases [<http://www.ics.uci.edu/~mlearn/MLRepository.html>]. Irvine, CA: University of California, Department of Information and Computer Science, 1998.
- [19] J. Laurikkala and M. Juhola, Nearest neighbour classification with heterogeneous proximity functions. In: A. Hasman, B. Blobel, J. Dudeck, R. Engelbrecht, G. Gell, and H-U. Prokosch, eds. *Medical Infobahn for Europe: Proceedings of MIE2000 and GMDS2000*. Studies in Health Technology and Informatics, vol. 77. Amsterdam: IOS Press, 2000: 753-757.

Kamada-Kawai algoritmin muunnelma dynaamisten verkkojen tiiviiseen asetteluun

Otto Seppälä
Tietojenkäsittelyopin laboratorio
Teknillinen korkeakoulu
oseppala@cs.hut.fi

1 Johdanto

Verkon visualisointi on prosessi, jossa verkolle rakennetaan graafinen esitys verkon tietorakenne-esitykseen pohjautuen. Verkon solmujen sijainnin tässä graafisessa esityksessä sanotaan muodostavan **verkon asettelun** (graph layout). Tyypillisesti tämä asettelu rakennetaan ihmisen luettavaksi tai jonkin teknisen asetteluongelman ratkaisuksi. Käsittelemme tässä ihmisen luettavaksi tarkoitettuja verkkoja, joiden ongelmia kirjoittaja on tutkinut myös diplomityössään [4].

Verkon asettelun perustavoitteena on pyrkiä tuottamaan verkon alkioiden asettelu, joka on visuaalisesti miellyttävä. Jotta tällainen asettelu voitaisiin tuottaa automaattisesti, käytetään asettelun laadun mittana joukkoa kriteerejä, joita asettelu-algoritmi pyrkii maksimoimaan joko suoraan tai välillisesti. Esitämme seuraavassa joukon kriteerejä, jotka mittaavat jollakin näyttölaitteella esitettävän, rakenteeltaan muuttuvan verkon asettelun hyvyttä.

1. Vierussolmujen läheisyys: Jos kahden verkon solmun välillä on kaari, on loogista että solmut sijaitsevat lähekkäin myös verkon visualisaatiossa. Verkon solmujen etäisyyden asettelussa tulee siis riippua niiden välisen lyhimmän polun pituudesta tietorakenteesta.

2. Ei päällekkäisiä solmuja: Jos kaksi solmua asetellaan kokonaan tai osittain päällekkäin, solmuissa olevan tiedon lukeminen tulee vaikeaksi tai mahdottomaksi ja solmuihin liittyvät kaaret voidaan tulkita väärin. Asettelu ei saa koskaan sisältää päällekkäisiä solmuja.

3. Minimimäärä risteäviä kaaria: Kun verkon solmut asetellaan, on mahdollista, että jotkin solmuja yhdistävät kaaret risteävät toisensa. Jokaisen risteävän kaariparin kohdalla on riski, että kaaret tulkitaan väärin. Purchase et al. toteavat [3], että verkon luettavuus vähenee, kun asettelun sisältämien kaarten risteymien määrä kasvaa. Risteävät kaaret myös hidastavat olennaisesti verkon tulkitsemista.

Verkon asettelua, joka ei sisällä lainkaan risteäviä kaaria kutsutaan *planaariseksi*. Suurimmalle osalle verkkoja ei ole mahdollista löytää planaarisista asettelua verkon topologian vuoksi.

4. Tehokas tilankäyttö: Kun verkon visualisointi on tarkoitettu esitettäväksi näyttöruudulla, näytön resoluutio ja näyttöpinta-ala rajoittavat kerrallaan esitettävien verkkoalkioiden määrää. Mikäli visualisaation koko ylittää näytön rajat, käyttäjä joutuu selaamaan visualisaatiota, mikä taas häiritsee tehokasta työskentelyä. Jos koko verkkoa skaalataan

pienemmäksi, menetetään verkon luettavuus. Verkon sijoittelualgoritmin pitäisi siksi käyttää tarjolla oleva pinta-ala mahdollisimman tehokkaasti.

5. Mahdollisimman nopea asettelu: Sijoittelualgoritmin täytyy voida sijoitella solmut nopeasti, mikäli verkon visualisointia käytetään interaktiivisessa sovelluksessa.
6. Ajallisesti toisiaan seuraavien verkon tilojen asettelujen tulee erota toisistaan mahdollisimman vähän muuttumattomien verkon osien osalta. Jos lokaali muutos verkon tietorakenteeseen aiheuttaa laajemman muutoksen asettelussa, käyttäjä joutuu tulkitsemaan muuttuneen alueen uudelleen. Tällöin on myös suuri riski siitä, että tapahtunut muutos verkossa jää havaitsematta.

Käytännössä viimeisin ehdoista voidaan purkaa kahdeksi ehdoksi:

- Minimimäärä liikkuneita solmuja: On erittäin vaikeaa seurata joukkoa yhtäaikaaisesti liikkuvia keskenään samankaltaisia alkioita. Tämän vuoksi asettelualgoritmin tulisi minimoida liikkuneiden solmujen määrä.
- Jokaisen liikkuneen solmun liike tulee minimoida. Mitä kauempana solmu on alkuperäisestä paikastaan, sitä hankalampaa liikkeen seuraaminen on. Jos solmu siirretään kauas alkuperäisestä paikastaan, voidaan se virheellisesti tulkita poistetuksi ja vastaavasti uudessa paikassaan lisätyksi.

Iteratiiviset metodit kuten Kamada-Kawai [2] algoritmi täyttävät implisiittisesti suuren osan näistä kriteereistä. Kun verkko läpikäy muutoksia, voidaan algoritmia jatkaa käyttämällä vanhaa asettelua iteraation alkutilana. Kun tarkoituksena on esittää pienessä tilassa dynaamisesti muuttuvia verkkoja, täytyy perusalgoritmin toimintaan tehdä joitakin muutoksia.

Kappaleessa 2 on lyhyesti kuvattu perusalgoritmin toiminta ja kappaleessa 3 on esitetty muutokset joilla pyritään täyttämään kuvaamme kriteerit.

2 Kamada-Kawai verkkojen asettelualgoritmi

Kamada-Kawai (KK) asettelualgoritmissa verkkoa mallinnetaan jousimallilla. Jousimallissa verkon solmut on kiinnitetty toisiinsa virtuaalijousilla, jotka kohdistavat solmuihin Hooken lain mukaisia voimia.

Kuten kaikki fyysiset mallit, pyrkii tämäkin malli kohden tasapainotilaa, mallin energiaminimiä. KK-algoritmissa pyritään etsimään paikallinen minimi koko järjestelmän kaikkien joustojen jännitysenergialle. Koska joustojen potentiaalienergia on suhteessa niiden poikkeamaan lepotilasta, pyrkii minimointi saattamaan visualisoinnin kaarten pituudet mahdollisimman lähelle niiden haluttua pituutta.

Jatkossa termillä *etäisyys* tarkoitetaan kahden solmun v_i, v_j välisen lyhimmän polun pituutta. Tätä etäisyyttä merkitään $d_{path}(v_i, v_j)$. Kun tarkoitetaan solmujen etäisyyttä toisistaan visualisaatiossa käytetään termiä *visuaalinen etäisyys*, jota merkitään vastaavasti $d_{vis}(v_i, v_j)$.

KK-mallissa kaikkien solmujen v_i, v_j välillä on em. jousi, jonka pituus l_{ij} on suoraan verrannollinen solmujen väliseen etäisyyteen $d_{path}(v_i, v_j)$ verkossa G . Näin lähekkäiset solmut saadaan sijoittumaan lähekkäin myös visualisaatiossa ja päinvastoin. Joustojen jäykkyys k_{ij} on verrannollinen solmujen etäisyyden $d_{path}(v_i, v_j)$ neliön käänteislukuun, mikä vähentää kaukana toisistaan olevien solmujen riippuvuutta toisistaan. Solmujen etäisyyttä toisistaan asettelun kulloisessakin tilassa on kaavassa 1 merkitty $d_{vis}(v_i, v_j)$.

$$\sum_{i=1}^N \sum_{j=i+1}^N \frac{1}{2} k_{ij} (d_{vis}(v_i, v_j) - l_{ij})^2 \quad (1)$$

KK käyttää minimin löytämiseen Newton-Raphson-iteraatiota. Iteraatiolla ei pyritä ratkaisemaan koko yhtälön minimiä kerralla, vaan minimiä kohti liikutaan siirtämällä solmuja yksi kerrallaan. Laskenta tapahtuu kahdessa askeleessa. Aluksi algoritmi valitsee siirrettävän solmun laskemalla approksimaation jännitysenergioille. Tämän jälkeen jännittyneintä solmua siirretään kohti minimiä Newton-Raphson-iteraatiolla kunnes iteraatioaskelen pituus tippuu alle halutun rajan.

Solmun valinta ja iteraatio toistetaan kunnes valinnassa laskettu energia-approksimaatio putoaa alle halutun rajan.

3 Muutokset algoritmiin

Jos tarkastellaan algoritmin toimintaa suhteessa määrittelemiimme kriteereihin, toimii Kamada-Kawai algoritmi monessa suhteessa hyvin. Vierussolmut sijoittuvat lähekkäin ja solmujen väliset jouset estävät päällekkäisten solmujen esiintymisen asettelussa tehokkaasti. Risteävien kaarten määrään algoritmi vaikuttaa välillisesti, sillä risteävien kaarten päätepisteet ovat jännityksessä toisiinsa nähden. Risteämiä kuitenkin esiintyy myös paikallisissa minimeissä, joten niitä on käytännössä mahdotonta täysin välttää.

Kriteerien 4, 5, ja 6 suhteen sen sijaan voidaan tehdä parannuksia pienillä muutoksilla.

3.1 Tehokas tilankäyttö

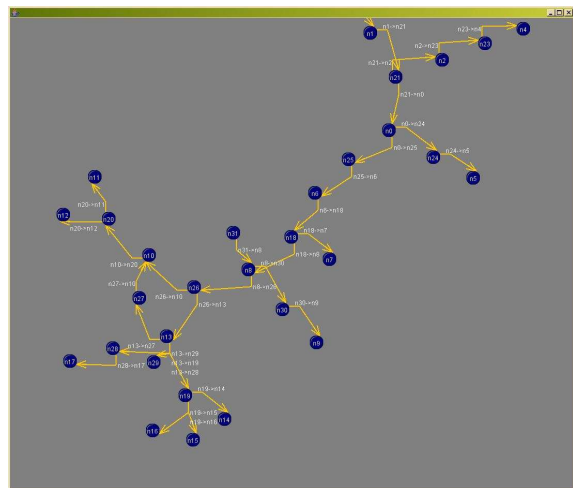
Verkon asettelun tilankäyttöä voidaan parantaa muuttamalla funktiota, joka laskee polkujen pituuksista vastaavat jousimallin jousien pituudet. Alkuperäinen funktio laskee jousien pituudet kertomalla polkujen pituudet jollakin vakiokertoimella k . Yksinkertaisin pahin tapaus tällaisilla jousilla on linkitetty lista, jossa

jokainen solmu on kiinnitetty kahteen naapurisolmuun. Tällainen rakenne kuvautuu suoraksi jonoksi. Lyhytkin ketju ylittää nopeasti rajallisen kuva-alueen rajat. Tämän vuoksi esitämme, että funktion joka laskee jousien pituudet vastaavien polkujen pituuksista tulisi kasvaa alkuperäistä funktiota hitaammin. Käytännössä voidaan käyttää painotettua keskiarvoa jostakin alilinearisesta funktiosta ja alkuperäisestä etäisyysfunktioista.

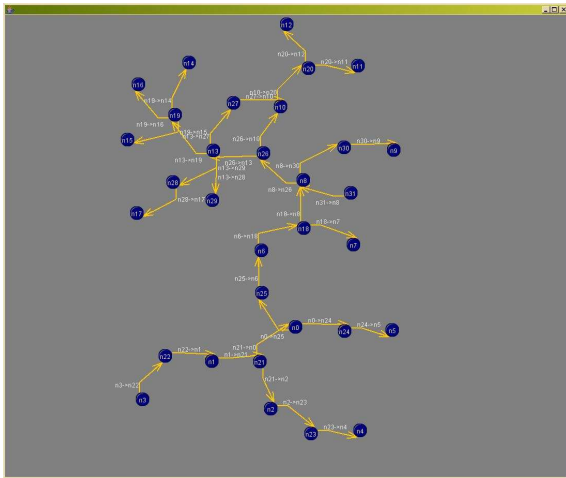
Koetulokset tämänkaltaisilla funktioilla tuottivat tiheämmin pakattuja asetteluja, mutta pitivät solmut silti erillään toisistaan. Yksi mahdollinen funktio on kuvattu alla. Tässä muuttuja a on painotuskerroin väliltä $[0, 1]$.

$$l_{ij} = k \cdot (a + a \cdot \log(d_{path}(i, j)) + (1 - a) \cdot d_{path}(i, j)) \quad (2)$$

Kuvat 1 ja 2 esittävät, kuinka etäisyysfunktio vaikuttaa verkon asetteluun. Molemmissa kuvissa on aseteltu sama verkko käyttäen eri etäisyysfunktioita. Kuva 1 esittää asettelun alkuperäisellä funktiolla ja kuva 2 funktiolla 2 kun $a = 0.8$. On helppo havaita, että jälkimmäisen kuvan esitysmuoto käyttää vähemmän tilaa.



Kuva 1: Alkuperäinen etäisyysfunktio



Kuva 2: Tiivis etäisyysfunktio

3.2 Epäyhtenäisten verkkojen asettelu

Alkuperäinen KK toimii vain yhtenäisillä verkoilla, sillä asettelualgoritmi johtaa visualisaation kaarten pituudet verkon polkujen pituuksista. Koska epäyhtenäisessä verkossa polku kahden solmun välillä, jotka kuuluvat eri yhtenäisiin komponentteihin on äärettömän pitkä, algoritmi joko sijoittaa nämä solmut äärettömän kauas toisistaan tai (jousten jäykkyystekijästä johtuen) mahdollisesti antaa eri yhtenäisten komponenttien peittää toisensa asettelussa.

Esitämme ratkaisuna ongelmaan ns. *virtuaalikaaret*, jotka yhdistävät yhtenäiset komponentit toisiinsa. Virtuaalikaari eroaa tavallisesta kaaresta siinä, että sitä ei koskaan piirretä näkyviin, ja sen pituus on olennaisesti suurempi kuin normaalin kaaren. Virtuaalikaari voi syntyä uuden solmun tai yhtenäisen komponentin lisäyksen yhteydessä, tai kun poistetaan kaari, joka muuten jakaisi verkon useampaan yhtenäiseen komponenttiin. Käytännön kokeissa epäyhtenäisten verkkojen asettelu saatiin toimimaan helposti virtuaalikaarten avulla.

3.3 Algoritmin nopeus

KK käyttää asettelun laskemiseen jonkin all pairs shortest paths (APSP) algoritmin laskemaa etäisyysmatriisia. APSP:n ylläpitoon verkon painojen päivityksen yhteydessä on käytetty algoritmitutkimuksessa runsaasti aikaa ja vanhimmat tulokset on esitetty jo 50-luvulla. Esitämme seuraavassa eräät muutokset, joilla Floydin algoritmin[1] käsittelemien solmuparien määrää voidaan supistaa. Nämä muutokset on valittu vain yksinkertaisuutensa vuoksi, algoritmillisesti tehokkaampia menetelmiä on lukuisia.

Kaaren lisäys Kaaren $v_i, v_j \in E$ lisäys ei koskaan voi pidentää yhtään polkua kahden solmun välillä, joten vanhan matriisin arvoja voidaan käyttää uuden matriisin alkuarvoina. Kaikkia matriisin arvoja ei tarvitse päivittää. Verkon solmut voidaan jakaa kolmeen ryhmään sen mukaan, eroaako etäisyys solmusta v_k uuden kaaren päätepisteisiin v_i, v_j vähintään kahdella.

Ryhmä 1 $d_{path}(v_i, v_k) + 1 < d_{path}(v_j, v_k)$

Ryhmä 2 $d_{path}(v_i, v_k) > d_{path}(v_j, v_k) + 1$

Ryhmä 3 $|d_{path}(v_i, v_k) - d_{path}(v_j, v_k)| < 2$

Ainoastaan kaaret ryhmien 1 ja 2 välillä täytyy päivittää, sillä muissa tapauksissa ryhmien välillä olevat kaaret eivät voi tuottaa lyhyempiä polkuja. Koska kaikki syntyvät lyhyemmät polut kulkevat lisätyn kaaren kautta, voidaan päivitys laskea ajassa $O(V^2)$.

Kaaren poisto Kaaren poistossa päivitettävät solmut löydetään tutkimalla poistettavan kaaren päätepisteiden sarakkeita etäisyysmatriisissa.

Ryhmä 1 $d_{path}(v_i, v_k) + 1 = d_{path}(v_j, v_k)$

Ryhmä 2 $d_{path}(v_i, v_k) = d_{path}(v_j, v_k) + 1$

Ryhmä 3 $d_{path}(v_i, v_k) = d_{path}(v_j, v_k)$

Jos solmu v_k kuuluu ryhmään 1, on mahdollista, että lyhin polku solmusta v_i kulkee solmun v_j kautta. Käänteinen pätee ryhmälle 2. Ryhmään 3 kuuluvien solmujen reitit eivät taasen voi kulkea poistetun kaaren kautta, sillä etäisyys kaaren päätepisteisiin on sama.

Tässäkin tapauksessa ainoastaan ryhmien 1 ja 2 väliset kaaret täytyy päivittää, sillä muut polut eivät kulkeneet poistetun kaaren kautta. Poiston tapauksessa päivitys rakennetaan merkitsemällä päivitettävien polkujen pituus arvoon V , ja suorittamalla valitua APSP-algoritmia kunnes ratkaisu saavutetaan. Tällaisen ratkaisun päivitys on hidasta, $O(V^3)$ vaikka päivitettävien polkujen määrää voidaankin em. tavalla rajoittaa.

Solmujen lisäykset ja poistot Kun asetelusta poistetaan tai siihen lisätään solmuja, jotka eivät alunperin kuuluneet siihen, ei Kamada-Kawai algoritmin kaikkia askeleita tarvitse suorittaa. Käytännössä vain päivitetään etäisyysmatriisi lisäämällä tai poistamalla vastaava sarake-rivi pari, sekä lisäämällä virtuaalikaari, joka tekee verkosta yhtenäisen. Tämän jälkeen solmun paikka verkossa voidaan etsiä suorittamalla KK:n iteraatio-osa. Vastaavasti poistettava solmu on aina yhteydessä vain virtuaalikaarella muuhun verkkoon, joten poisto tapahtuu vain poistamalla haluttu rivi ja sarake etäisyysmatriiseista.

3.4 Minimimuutokset toisiaan seuraavien tilojen välillä

Koska yksittäisten solmujen lisäykset ja poistot sekä kaarten poistot eivät vaadi muun verkon asettelun uudelleenlaskentaa, saavutetaan minimimuutuskriteeri näissä tapauksissa aina. Ainoa potentiaalisesti suurempia muutoksia aiheuttava operaatio onkin kaaren lisäys.

Koska Kamada-Kawai on luonteeltaan iteratiivinen metodi, voidaan asettelun aiempaa

tilaa käyttää iteraation alkukohtana. On todennäköistä, että kaaren lisääminen ei vaikuta suureen osaan solmuja, mutta tätä on käytännössä mahdoton taata. Esimerkkinä huonosta tapauksesta on mm. tilanne, jossa kahden suuren yhtenäisen verkon jotkin keskeiset solmut yhdistetään kaarella toisiinsa.

4 Yhteenveto

Kun dynaamisesti muuttuvan verkon visualisaatio halutaan esittää rajoitetulla kuva-alalla täytyy verkon asettelualgorimeihin tehdä joitakin keskeisiä muutoksia. Tässä paperissa on esitetty joukko muutoksia Kamada-Kawai asettelualgoritmiin, jotka parantavat algoritmin toimintaa edellämainitussa tilanteessa. Esitimme myös joukon kriteerejä, joiden mukaan asettelun laatua voidaan arvioida. Lisäksi kuvattiin, kuinka esitetyt muutokset parantavat algoritmin toimintaa näiden kriteerien suhteen.

Viitteet

- [1] R. W. Floyd. Algorithm 97: shortest path. *Communications of the ACM*, 5(6):345, 1962.
- [2] T. Kamada and S. Kawai. An algorithm for drawing general undirected graphs. *Information Processing Letters*, 31(1), 1989.
- [3] H. C. Purchase, R. F. Cohen, and M. James. Validating graph drawing aesthetics. In F. Brandenburg, editor, *Symposium on Graph Drawing 95, Passau Germany, 1995, Lecture Notes in Computer Science volume 1027*, pages 435–446. Springer Verlag, 1996.
- [4] O. Seppälä. Visual Java Debugger for Computer Science Education. Master's thesis, Helsinki University of Technology, 2003.

Konferenssi-protokollan automaattinen testaus

Antti Kervinen ja Pablo Virolainen

Tampereen teknillinen yliopisto, Ohjelmistotekniikan laitos

Antti.Kervinen@tut.fi

Pablo.Virolainen@tut.fi

1 Johdanto

Yleisesti käytössä olevissa automaattisissa testausmenetelmissä joko ihminen tai kone (mahdollisesti ihmisen avustuksella) laatii joukon testitapauksia. Kukin testitapaus kuvaa tapahtumaketjun, joka päättyy testikohteelta saaduista vasteista riippuen tulokseen *hyväksytty* (testikohde käyttäytyi niin kuin odotettiin), *hylätty* (testikohde teki jotain laitonta) tai *ratkaise-maton* (testikohde käyttäytyi sallitulla tavalla, mutta testitapauksella testattavaksi tarkoitettu asia jäi testaamatta).

Samoilutestaus [4] on automaattista testausta, missä testiä ohjaavalle tietokoneohjelmalle, *testikoneelle*, annetaan valmiiden testitapausten sijaan yksi *vaatimuskone*.

Tässä paperissa esitetään testaustulokset, jotka saimme testatessamme samoilutestausmenetelmällä erästä konferenssi-protokollaa palvelutasolta. Konferenssi-protokolla tarjoaa keskustelupalvelun, missä samaan konferenssiin osallistuvat keskustelijat saavat toistensa lähettämät viestit. Protokolla on toteutettu Twenten yliopistossa [2]. Lähdekoodia voidaan käänösaikana parametrisoida niin, että tuloksena on erilaisia virheitä sisältäviä mutanteja. Tätä toteutusta ja sen mutanteja on testattu aiemminkin [1, 6]. Aiemmissa testeissä testikohteina on ollut yksittäinen protokollan proses-

si, ei useiden prosessien yhdessä tuottama palvelu. Valitsimme palvelunäkökulman saadaksemme testikohteeksi todellista rinnakkaisuutta sisältävän järjestelmän.

2 Samoilutestaus

Vaatimuskone on järjestelmältä odotettua käyttäytymistä kuvaava suunnattu graafi, jonka kaaret on nimetty. Testiajon aikana testikone pitää kirjaa siitä, missä vaatimuskone-graafin solmussa suoritus on menossa. Aloittamista varten yksi solmuista on erityinen "alkutila". Jokainen kaaren nimi vastaa testikohteelle lähetettävää syötettä tai sen palauttamaa vastetta.

Solmusta lähtevät kaaret määräävät, mitä testiajon ollessa tässä solmussa voidaan tehdä. Vaihtoehtoja ovat jonkin kaaren nimeä vastaavan syötteen lähettäminen testikohteelle; testikohteen vasteen kuunteleminen, jos jokin lähtevien kaarien nimistä on vaste; testikohteen uudelleenkäynnistäminen ja testin aloitus alkutilasta; ja testauksen lopettaminen. Vaatimuksista poikkeava käyttäytyminen voidaan havaita sen kieltäytymisenä annettusta syötteestä tai sellaisen vasteen antamisena, joka solmusta lähtevien kaarten mukaan ei ole sallittu. Jos mitään näkyvää vastetta ei saada, sitä pidetään erityisenä *hiljaisuus*-vasteena.

Samoilutestauksen etuina tavanomaiseen automaattiseen testaukseen verrattuna ovat muunmuassa korkeampi automaation taso, koska ihmistä ei tarvita keksimään testitapauksia, ja parempi hyötysuhde, koska siinä ei synny turhia, ”ratkaisemattomia” testiajoja. Lisäksi tiettyjen virheiden löytäminen muuttuu todennäköisemmäksi, kun ennen jokaista testitapausta testikohdetta ei järjestelmällisesti pakoteta alkutilaan. Kun testi muodostaa pitkän katkottoman suorituksen, testattavan järjestelmän sisäisten tietorakenteiden ja laskureiden käsittelyvirheet saattavat vaikuttaa ulkoiseen käyttäytymiseen — vaikkei niin voisiakaan käydä testauksessa, missä lyhyiden testitapausten välillä testikohteen tila pyritään palauttamaan testitapausta edeltäväksi.

Nämä edut ovat suurimmat testattaessa jatkuvasti toiminnassa olevia rinnakkaisuutta sisältäviä järjestelmiä. Tällaisille järjestelmille on ominaista suuri määrä tiloja, joissa järjestelmä voi kokonaisuutena olla, monet samanaikaisesti sallitut vasteet ja toiminnan epädeterministisyys ulkoisen tarkkailijan näkökulmasta.

Olkoon tarkastelun kohteena keskustelupalvelun tuottava tietoliikenneprotokolla käyttäjien näkökulmasta testattuna. Sallittuja tapahtumia voi esimerkiksi olla kahden eri sanoman vastaanottaminen (epädeterminismi: sanomat voidaan vastaanottaa missä järjestyksessä tahansa, vaikka järjestelmään olisi annettu syötteet aina samassa järjestyksessä) ja samanaikaisesti kolmannen osallistujan viestin lähettäminen keskusteluun. Sen sijaan että kirjoitettaisiin kaikkia näitä vaihtoehtoja varten testitapauksia, samoilutestauksessa testikone voi vaatimuskonetta seuraten yrittää testata mahdollisimman suuren osan näistä vaihtoehtoisista käyttäytymisistä sen ajan puitteissa, mikä sillä on käytettävissään.

Eräs samoilutestaukseen liittyvä tutkimusongelma on, miten valitaan vaatimuskonesta seuraava testiaskel — kuun-

nellako vastetta vai lähettääkö jokin syöte, mikä? Nyt raportoitavissa testiajoissa käytetyt valinta-algoritmit on julkaistu hiljattain [5].

3 Konferenssiprotokolla

Konferenssiprotokollassa käyttäjä voi liittyä konferenssiin, lähettää niihin viestejä, jotka muut samaan konferenssiin osallistuvat vastaanottavat, ja poistua konferensseista.

3.1 Toteutus ja mutaatiot

Testaamamme protokollan toteutus sallii käyttäjän kuuluvan korkeintaan yhteen konferenssiin kerrallaan. Protokollan päät palvelevat yhtä käyttäjää kerrallaan ja pitävät kirjaa *potentiaalisista osallistujista* ja *osallistujista*. Potentiaaliset osallistujat on joukko IP- ja porttiosoitepareja, jotka luetaan käynnistyksen yhteydessä asetustiedostosta. Tämän jälkeen joukko on muuttumaton. Osallistujat, joka on osajoukko potentiaalisista osallistujista, on tyhjä kun käyttäjä ei kuulu mihinkään konferenssiin. Muulloin tässä joukossa pidetään kirjaa käyttäjän konferenssin muista osallistujista.

Protokollan päät lähettävät toisilleen *join*, *answer*, *data* ja *leave*-sanomia seuraavien sääntöjen mukaisesti:

1. Kun käyttäjä A liittyy konferenssiin k , lähetetään potentiaalisille osallistujille $join(A, k)$.
2. Jos käyttäjä A kuuluu konferenssiin k ja vastaanotetaan $join(B, k)$, A :n osallistujiin lisätään B ja lähettäjälle vastataan $answer(A, k)$. Muutoin $join$ unohdetaan.
3. Jos käyttäjä A kuuluu konferenssiin k ja vastaanotetaan $answer(B, k)$, A :n osallistujiin lisätään B . Muutoin $answer$ unohdetaan.
4. Kun käyttäjä A lähettää viestin m , lähetetään osallistujille $data(m)$.

5. Jos käyttäjä A kuuluu konferenssiin k , vastaanotetaan $data(m)$ ja lähettäjä B löytyy osallistujien joukosta (lähettäjän osoitteen perusteella), kerrotaan A :lle viestin m saapuneen lähettäjältä B . Jos B :tä ei löydy osallistujista, vastataan lähettäjälle $join(A, k)$.
6. Kun käyttäjä A poistuu konferenssista k , lähetetään $leave(A, k)$ osallistujille ja tyhjennetään osallistujajoukko.
7. Jos vastaanotetaan $leave(B, k)$ poistetaan B osallistujista.

Testattujen mutaatioiden virheet esitetään taulukossa 1.

Testatut järjestelmät sisältävät kolme asiakasta. Jokaista mutaatiota kohti tehtiin kaksi testattavaa järjestelmää. Ensimmäisen sisälsi yhden mutatoitua ja kaksi ehjää toteutusta, toinen sisälsi kaksi mutatoitua ja yhden ehjän toteutuksen.

3.2 Vaatimukset

Laadimme vaatimuskoneen TVT-verifiointityökaluilla [3]. Se sisältää kaksi konferenssia, joihin käyttäjät voivat liittyä, yhden lähetettävän viestin ja kolme ”järkevästi” käyttäytyvää käyttäjää. Käyttäjät yrittävät liittyä konferenssiin vain silloin kun eivät kuulu mihinkään konferenssiin, yrittävät lähettää viestejä vain konferenssissa ollessaan ja yrittävät poistua konferenssista vain kuuluessaan johonkin konferenssiin. Vaatimuskone¹ sisältää noin 192 000 tilaa ja 853 000 tilasiirtymää.

Vaadimme käyttäjien vastaanottavan kaikki konferenssiinsa lähetetyt viestit, vaikkei sitä luvatakaan protokollan toteutuksessa eikä spesifikaatiossa. Koska kaikkia protokollaprosesseja ajettiin samassa tietokoneessa testikoneen kanssa, sanomien katoaminen oli erittäin epätodennäköistä (ellei testiä ajettu hyvin nopeasti, jolloin käyttäjät lähettelivät vieste-

jä lähes yhtäaikaa). Tämä vaatimus auttoi havaitsemaan mutatoinnista johtuvia viestien hukkumisia, mikä oli tärkeää, sillä suurin osa mutatoinnin vaikutuksista voitiin havaita juuri näin.

4 Testiajot

Testiajojen tulokset esitetään taulukossa 2. Taulukon riveillä näytetään kutakin mutanttia koskevien testitapausten tulokset. Kolmella pääsarakeella on käytetty valinta-algoritmi: Pelaaja, Ahne satunnainen ja Satunnainen algoritmi. Kukin pääsarake sisältää sarakkeen yhden mutantin (1) ja kaksi mutanttia (2) sisältävälle järjestelmälle. Taulukon soluissa esitetään kahden testiajon keskiarvo suoritettujen testiaskelien määrästä ennen kuin vaatimuksista poikkeava käyttäytyminen havaittiin. Jos virhettä ei havaittu 10 000 testiaskelen kuluessa, taulukossa on viiva.

Pelaaja-algoritmi (esitely [5] nimellä *Adaptive player, complex step evaluation function*) laskee jokaisessa vaatimuskoneen solmussa viisi testiaskelta eteenpäin ja valitsee seuraavan tapahtuman siten, että se saa todennäköisimmin viiden askeleen kuluessa ensisijaisesti lähetettyä ja kuunneltua mahdollisimman monta aiemmin testaamatonta syötettä ja vastetta. Toissijaisesti se pyrkii mahdollisimman moneen aiemmin käymättömään vaatimuskoneen solmuun ja seuraavaksi se pyrkii suorittamaan mahdollisimman monta sellaista vaatimuskoneen kaarta, joita ei ole aiemmin suoritettua. Viimeimpänä se välttää suorittamasta sellaisia kaaria, jotka on suoritettu jo hyvin monta kertaa. Algoritmi esitellään tarkemmin valinta-algoritmien vertailuun keskittyvässä julkaisussamme.

Ahne satunnainen (esitely [5] nimellä *Greedy random*) algoritmi pyrkii valitsemaan testiaskelen siten, että solmussa

¹http://www.cs.tut.fi/~ask/MBT04/CL3_CF2_MSG1.lsts.gz

katetaan aina aiemmin kattamaton kaari (vasteet ovat etusijalla syötteisiin nähden). Jos se ei ole mahdollista, valinta on satunnainen. Satunnainen algoritmi valitsee satunnaisesti yhden solmun kaarista. Jos se oli vaste, jäädään kuuntelemaan testikohteen vastetta, muutoin lähetetään kaarta vastaava syöte.

Testiajoja ajettiin rinnakkain 29 tietokoneella (Solaris 8 käyttöjärjestelmät, koneissa oli 1–2 kpl UltraSparc v9 - prosessoreja, kellotaajuuksilla 296–648 MHz), mutta kunkin testiajon kaikki prosessit sijaitsivat samalla tietokoneella.

5 Tulosten analysointia

Testitulokset osoittivat, että hyvin käyttäytyvien käyttäjien tapauksessa testatuista mutaatioista 18 eivät todennäköisesti vaikuta protokollan tarjoamaan palveluun. Kuitenkin voi olla, että kaksi näistä valittua eritavalla mutatoitua asiakasta samassa järjestelmässä rikkovat palvelun.

Lisäksi uskomme, että osa näistä mutanteista olisi osoittautunut virheellisiksi, jos vaatimuskone olisi sisältänyt yhdenkin ”hankalan” käyttäjän, joka olisi sanonut esimerkiksi ”join” kuullessaan jo johonkin konferenssiin, ”leave” vaikei kuulukaan konferenssiin ja lähettänyt sanomia vaikei kuulukaan konferenssiin.

Ahneeseen ja satunnaiseen valinta-algoritmiin jouduimme lisäämään ylimääräisen viiveen kahdesta syystä. Kun syötteitä annettiin testikohteelle nopeassa tahdissa, alkoi protokollan toteutuksen käyttämät UDP-kanavat hukata sanomia. Tästä seurasi toisinaan käyttäjien toisilleen lähettämien viestien katoaminen ja siten poikkeaminen vaatimuskoneen käyttäytymisestä.

Toisena syynä oli seuraava ilmiö. Käyttäjä A lähettää viestin konferenssiin k , johon ei kuulu käyttäjä B . Tällöin vaadimme, ettei käyttäjä B ei voi vastaanottaa tätä viestiä. Kuitenkin käyttäjän B liittyessä nopeasti konferenssiin k toteutus mahdol-

listaa viestin vastaanoton. Näin käy esimerkiksi jos A :n protokollaprosessi ei saa ajoaikaa ennen B :n liittymistä konferenssiin, jolloin todellinen viestien lähetysjärjestys onkin eri kuin vaatimuskoneen oletama.

Kiitokset Työ tehtiin TEKESin, Conformiq Software Oy Ltd:n ja Nokian tutkimuskeskuksen rahoittamassa SASOKE-projektissa.

Viitteet

- [1] Belinfante, A., Feenstra, J., de Vries, R. G., Tretmans, J., Goga, N., Feijs, L., Mauw, S. & Heerink, L.: “Formal Test Automation: A Simple Experiment”. *International Workshop on Testing of Communication Systems*, Kluwer Academic, 1999, sivut 179–196.
- [2] Conference Protocol Case Study, <http://fmt.cs.utwente.nl/ConfCase/>. Viimeksi päivittänyt Jan Feenstra 19.11.1999.
- [3] Hansen, H., Virtanen, H. & Valmari, A.: “Merging State-Based and Action-Based Verification”. *Proc. ACSD 2003, 3rd International Conference on Application of Concurrency to System Design*, IEEE Computer Society 2003, sivut 150–156.
- [4] Helovuo, J. & Leppänen, S.: “Exploration Testing”. *Proc. ICACSD 2001, 2nd IEEE International Conference on Application of Concurrency to System Design*, IEEE Computer Society 2001, sivut 201–210.
- [5] Kervinen, A. & Virolainen, P.: “Heuristics for Faster Error Detection with Automated Black Box Testing”. *Proc. Model-Based Testing 2004* 27.–28.3. 2004 Barcelona, Espanja. Julkaistaan ENTCS-sarjassa.
- [6] Pyhälä, T. & Heljanko, K.: “Specification Coverage Aided Test Selection”. *Proc. ACSD’2003, Third International Conference on Application of Concurrency to System Design*, IEEE 2003, sivut 187–195.

Taulukko 1: Mutaatiot

No	Poikkeava käyttäytyminen
0	– (toimivaksi tarkoitettu versio)
10	Ei käsittele käyttäjän liittymispyyntöä
11	Ei käsittele käyttäjän lähetykspyyntöä
12	Ei käsittele käyttäjän poistumispyyntöä
13	Ei käsittele vastaanotettua sanomaa <i>join</i>
14	Ei käsittele vastaanotettua sanomaa <i>data</i>
15	Ei käsittele vastaanotettua sanomaa <i>answer</i>
16	Ei käsittele vastaanotettua sanomaa <i>leave</i>
17	Ei lähetä <i>answer</i> saatuaan <i>join</i>
18	Ei lähetä <i>join</i> saatuaan <i>data</i> ¹
19	Ei päivitä osallistujia saatuaan <i>answer</i>
20	Ei päivitä osallistujia saatuaan <i>leave</i>
21	Ei lähetä <i>join</i> liityttäessä
22	Ei lähetä <i>data</i> lähetykspyynnössä
23	Ei lähetä <i>leave</i> poistuttaessa ¹
24	Ei jätä konferenssia poistuttaessa
25	Ei tyhjennä osallistujia poistuttaessa
26	Ei liity konferenssiin liittymispyynnössä
27	Ei päivitä osallistujia saatuaan <i>join</i>
28	Ei välitä viestiä saatuaan <i>data</i>
29	Poistaa aina ensimmäisen osallistujan saatuaan konferenssiinsa <i>leave</i> ¹
30	Käsittelee <i>leave</i> riippumatta konferenssista
31	Ei tarkista osallistujia saatuaan <i>answer</i>
32	Ei tarkista potentiaaliaalisia osallistujia saatuaan <i>answer</i>
33	Ei tarkista konferenssia saatuaan <i>answer</i>
34	Ei tarkista osallistujia saatuaan <i>data</i> , ei välitä viestiä asiakkaalle ¹
35	Ei tarkista osallistujia saatuaan <i>join</i>
36	Ei tarkista potentiaaliaalisia osallistujia saatuaan <i>join</i>
37	Ei tarkista konferenssia saatuaan <i>join</i>
38	Poistuu konferenssista heti liityttyään
39	Lähetää kahdesti <i>join</i> liityttäessä
40	Lähetää kahdesti <i>data</i> viestittäessä
41	Lähetää kahdesti <i>leave</i> poistuttaessa
42	Lähetää aina <i>join</i> saatuaan <i>data</i>
43	Käsittelee käyttäjän nimen väärin
44	Asettaa saapuneen viestin pituudeksi 1500
45	Lähetää <i>data</i> takaisin lähettäjälleen
50	Lähetää kahdesti <i>answer</i> saatuaan <i>join</i>
51	Lähetää kahdesti <i>join</i> saatuaan <i>data</i>
52	Käsittelee sanomia ilman konferenssiakin
53	Ei ymmärrä vastaanottamia UDP-paketteja
54	Ei ymmärrä käyttäjältä saamia kommentoja
55	Saa käyttäjältä vain <i>answer</i> kommentoja
56	Ei voi kuunnella UDP-porttia (bind error)
57	Saa heti käynnistyttyään QUIT-signaalin
58	Puhdistaa aina lähetyksvälimuistin
59	Alustuksessa suurin määrä mutanteja ylitetty
60	Käyttäjää ja konferenssia ei nimetä

Taulukko 2: Tulokset

No	Pelaaja		Ahne sat.		Sat.	
	1	2	1	2	1	2
0	–	–	–	–	–	–
10	26	20	20	22	72	16
11	32	26	54	37	28	18
12	62	62	55	31	45	48
13	174	26	38	30	126	31
14	32	26	27	18	258	90
15	26	26	197	21	29	52
16	–	–	–	–	–	–
17	146	26	118	37	157	90
18	–	–	–	–	–	–
19	26	26	50	24	99	38
20	–	–	–	–	–	–
21	26	26	48	29	241	50
22	32	26	60	37	51	25
23	–	–	–	–	–	–
24	62	56	74	62	320	129
25	–	–	–	–	–	–
26	26	20	22	16	55	53
27	261	26	17	19	38	47
28	32	26	43	40	180	25
29	1784	1769	381	558	351	213
30	–	–	–	–	–	–
31	–	–	–	–	–	–
32	–	–	–	–	–	–
33	–	–	–	–	–	–
34	32	26	23	43	431	29
35	283	26	34	16	47	114
36	–	–	–	–	–	–
37	112	110	30	33	53	32
38	26	20	40	64	99	17
39	–	–	–	–	–	–
40	7095	7708	–	–	8657 ²	47 ²
41	–	–	–	–	–	–
42	–	–	–	–	–	–
43	164	29	306	118	85	76
44	–	–	–	–	–	–
45	31	21	16	34	178	35
50	–	–	–	–	–	–
51	–	–	–	–	–	–
52	26	20	33	19	27	37
53	26	20	26	21	35	19
54	26	20	29	15	43	18
55	26	20	33	24	97	32
56	26	20	22	24	41	42
57	13	9	11	9	17	10
58	–	–	–	–	–	–
59	–	–	–	–	–	–
60	26	20	41	15	39	21

¹ Selitys on toteutuksen lähdekoodin mukainen ja poikkeaa toteutuksen mukana tulleet dokumentaatiosta (*mutants.txt*).

² Virhe löytyi vain toisella testijalla.

Todistuskompleksisuudesta Boolean piirien toteutuvuustarkastuksessa

Matti Järvisalo
Teknillinen korkeakoulu
Tietojenkäsittelyteorian laboratorio
matti.jarvisalo@hut.fi

1 Johdanto

Lauselogiikan toteutuvuusongelmassa [17] kysytään, onko annetulle lauselogiikan lauseelle olemassa sen toteuttavaa totuusjakelua. Toteutuvuusongelma on tyypillinen **NP-täydellinen** ongelma [5], jolle ei tunneta yleistä polynomiaikaista ratkaisualgoritmia. Muun muassa suunnittelu-, mallintarkastus-, testaus- ja verifiointiongelmia [12, 4, 13, 3] voidaan ratkaista toteutuvuusongelmatapauksina. Tehokkaille toteutuvuustarkastimille on näin ollen suuri kysyntä. Toteutuvuustarkastimia onkin kehitetty ja käytetty onnistuneesti monien erilaisten ongelmien ratkaisemiseen.¹ Sekä täydelliset, systemaattiset että satunnaiset paikalliseen hakuun perustuvat toteutuvuustarkastusmenetelmät (katso esimerkiksi [18, 15, 14, 1]) ovatkin kehittyneet huomattavasti viimeisten noin 10 vuoden aikana. Yleisesti tehokkaita menetelmiä ei silti ole kyetty kehittämään.

Tehokkaat täydelliset toteutuvuustarkastimet perustuvat tyypillisesti *Davis-Putnam-Logemann-Loveland-menetelmään* (DPLL) [8], joka edellyttää syötteen olevan *konjunkttiivisessa normaalimuodossa* (KNM) [16], eli ongelmatauksen tulee koostua joukosta disjunktioita. KNM:n käyttö ongelman mallinuskienä on hyvin vaivalloista. Tämän

vuoksi tyypillisesti käytetään ensin yleisempää esitysmuotoa, joka tämän jälkeen käännetään KNM:oon. Polynomiaikainen käänös KNM:oon edellyttää apumuuttujien käyttöönottoa. Ongelmana on, että toteutuvuustarkastimien laskenta-aika kasvaa pahimmassa tapauksessa eksponentiaalisesti muuttujien määrän suhteen. Lisäksi käänös saattaa hävittää instanssin sisäisen rakenteen, jota muutoin olisi ollut mahdollista käyttää hyväksi hakuprosessissa.

Tässä työssä esitellään yleisemmälle, *Boolean piireiksi* [17] kutsutulle esitysmuodolle kehitetty taulumenetelmä. Menetelmää voidaan pitää DPLL-menetelmän yleistyksenä Boolean piireille. Boolean piirit ovat hyvin luonnollinen ja ongelman sisäisen rakenteen säilyttävä esitysmuoto monille käytännön ongelmille. Työ käsittelee laskennalliselta kannalta oleellisen *haarauttavan säännön*² käyttöä. Haarauttavan säännön käyttö johtaa pahimmassa tapauksessa eksponentiaaliseen määrään turhaa työtä toteutuvuushaun aikana. Työssä tarkastellaan, miten haarauttavan säännön rajoittaminen vaikuttaa toteutuvuustarkastuksen tehokkuuteen. Eriytyisesti tarkastellaan, kuinka lyhyitä todistuksia on mahdollista tuottaa haarauttavan säännön rajoitukset huomioonottaen.

Työssä osoitetaan, että haarauttavan

¹ Katso [10, 21] katsauksina erilaisiin toteutuvuustarkastustekniikoihin.

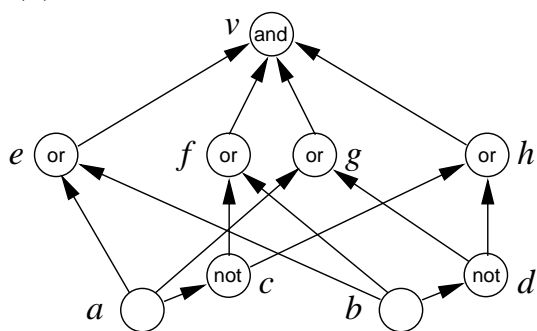
² Englanniksi *splitting rule* tai *cut rule*.

säännön käytön rajoittaminen millä tahansa tarkastelluista tavoista kasvattaa pienimmän mahdollisen todistuksen kokoa pahimmassa tapauksessa eksponentiaalisesti. Tulokset pätevät DPLL-menetelmään yleisesti käytetyllä käännöksellä Boolean piireistä tuotettujen konjuktiivisessa normaalimuodossa olevien lauseiden osalta.

2 Boolean piireistä

Boolean piirit ovat suunnattuja, asyklisiä verkkoja, joiden solmuja kutsutaan *porteiksi*. Portit voidaan jakaa kolmeen joukkoon: *tulosporteihin*, *väliporteihin* ja *syöteporteihin*. Jokaiseen tulos- ja väliporttiin liittyy Boolean funktio. Esimerkiksi Boolean piiristä on kuvassa 1. Kuvan piirissä v on tulosportti, c, d, e, f, g, h väliportteja ja a, b syöteportteja.

Piirin kaaret kuvaavat porttien funktionaalista riippuvuutta toisistaan. Esimerkiksi kuvan 1 piirissä porttien b, c, f välinen riippuvuus on muotoa $f = \text{or}(b, c)$, ja edelleen porttien a, c välillä on riippuvuus $c = \text{not}(a)$.



Kuva 1: Esimerkki Boolean piiristä.

Boolean piirien semantiikka on ilmeinen: kiinnitettäessä totuusarvot syöteporteihin määrittyvät muiden porttien totuusarvot yksikäsitteisesti. *Rajoitetussa Boolean piirissä* sallitaan rajoitteet piirin porttien totuusarvoiksi. *Boolean piirien toteutuvuusongelmassa* kysytään, onko annetun rajoitetun Boolean piirin syöteporteille olemassa sellaista totuusarvojakelua, joka ei aiheu-

ta ristiriitaa piirin rajoitteiden kanssa. Jos tällainen totuusjakelu on olemassa, sanotaan kyseistä piiriä *toteutuvaksi*, ja muulloin *toteutumattomaksi*. Kuten lauselogiikan tapauksessa, myös Boolean piirien toteutuvuusongelma on **NP**-täydellinen. Esimerkiksi kuvan 1 piirille ei ole sen toteuttavaa totuusjakelua rajoitteella “portti v on *tos*”. Toisaalta rajoitteilla “portti e on *epätos*” ja “portti g on *tos*” totuusjakelu $a \rightarrow \text{epätosi}, b \rightarrow \text{epätosi}$ toteuttaa piirin.

3 Taulumenetelmä

Eräs systemaattinen tapa ratkaista Boolean piirien toteutuvuusongelman instansseja on *taulumenetelmän* käyttäminen [7]. Taulumenetelmässä rakennetaan *tauluksi* kutsuttava binääripuu. Taulun juureen kuvataan tarkasteltavan Boolean piirin funktionaaliset riippuvuudet ja piirissä olevat rajoitteet. Jokaista *todeksi* (*epätodeksi*) rajoitettua porttia v kohden lisätään merkintä T_v (E_v). Esimerkkinä kuvan 1 piiri rajoitteella “portti v on *tos*” on kuvattuna kuvan 2 taulun merkinnöiksi 1–8.

Taulumenetelmä koostuu *säännöistä*. Taulun juuren luomisen jälkeen taulua rakennetaan käyttäen taulumenetelmän sääntöjä. Taulun jokainen haara edustaa totuusjakelua, joka mahdollisesti toteuttaa tarkasteltavan piirin. Jos taulun haarassa on merkinnät T_v ja E_v jollekin portille v , sanotaan haaraa *ristiriitaiseksi*. Jos taulun kaikki haarat ovat ristiriitaiset, sanotaan taulua *hylkäykseksi* (refutaatioksi). Jos rajoitetulle Boolean piirille on mahdollista rakentaa hylkäys, on piiri *toteutumaton*. Hylkäys on siis todistus piirin toteutumattomuudesta.

Tässä työssä tarkasteltava *BC-menetelmä* koostuu luonnollisista *päätelysäännöistä* koskien not-, or- ja and-portteja, sekä *taulun haarauttavasta säännöstä*.

Päätelysäännöt ovat seuraavat. Jos $v = \text{not}(v')$ ja haarassa on $\mathbf{T}v$ ($\mathbf{E}v$), voidaan päätellä $\mathbf{E}v'$ ($\mathbf{T}v'$), ja päinvastoin. Jos $v = \text{or}(v_1, \dots, v_k)$, niin voidaan päätellä seuraavaa.

- Jos haarassa on $\mathbf{E}v$, niin voidaan päätellä $\mathbf{E}v_i$ kaikilla $1 \leq i \leq k$, ja päinvastoin.
- Jos haarassa on $\mathbf{T}v_i$ jollakin $1 \leq i \leq k$, niin voidaan päätellä $\mathbf{T}v$.
- Jos haarassa on $\mathbf{T}v$ ja

$$\mathbf{E}v_1, \dots, \mathbf{E}v_{i-1}, \mathbf{E}v_{i+1}, \dots, \mathbf{E}v_k,$$

niin voidaan päätellä $\mathbf{T}v_i$.

Jos $v = \text{and}(v_1, \dots, v_k)$, niin voidaan päätellä seuraavaa.

- Jos haarassa on $\mathbf{T}v$, niin voidaan päätellä $\mathbf{T}v_i$ kaikilla $1 \leq i \leq k$, ja päinvastoin.
- Jos haarassa on $\mathbf{E}v_i$ jollakin $1 \leq i \leq k$, niin voidaan päätellä $\mathbf{E}v$.
- Jos haarassa on $\mathbf{E}v$ ja

$$\mathbf{T}v_1, \dots, \mathbf{T}v_{i-1}, \mathbf{T}v_{i+1}, \dots, \mathbf{T}v_k,$$

niin voidaan päätellä $\mathbf{E}v_i$.

1.	$v = \text{and}(e, f, g, h)$	
2.	$e = \text{or}(a, b)$	
3.	$f = \text{or}(b, c)$	
4.	$g = \text{or}(a, d)$	
5.	$h = \text{or}(c, d)$	
6.	$c = \text{not}(a)$	
7.	$d = \text{not}(b)$	
8.	$\mathbf{T}v$	
9.	$\mathbf{T}e$	(1, 8)
10.	$\mathbf{T}f$	(1, 8)
11.	$\mathbf{T}g$	(1, 8)
12.	$\mathbf{T}h$	(1, 8)
13.	$\mathbf{T}a$	(haar.)
14.	$\mathbf{E}a$	(haar.)
15.	$\mathbf{E}c$	(6, 13)
20.	$\mathbf{T}b$	(2, 9, 14)
16.	$\mathbf{T}b$	(3, 10, 15)
21.	$\mathbf{T}d$	(4, 11, 14)
17.	$\mathbf{E}d$	(7, 16)
22.	$\mathbf{E}d$	(7, 21)
18.	$\mathbf{E}h$	(5, 15, 17)
23.	\times	(21, 22)
19.	\times	(12, 18)

Kuva 2: BC-hylkäys kuvan 1 Boolean piirille.

Haarauttavan säännön ideana on salia *tapausanalyysi*, jossa tutkitaan, miten

päätelyä voidaan jatkaa, kun portille, jolle ei ole vielä merkintää haarassa, valitaan totuusarvo. Käytettäessä haarauttavaa sääntöä piirin porttiin v jakautuu siis käsiteltävä haara kahteen haaraan, joista toiseen merkitään $\mathbf{T}v$ ja toiseen $\mathbf{E}v$.

Esimerkkinä taulun rakentamisesta on kuvassa 2 hylkäys kuvan 1 Boolean piirille.

4 Ongelmanasettelu

Haarauttavan säännön käytössä ongelmalista on, että pahimmassa tapauksessa rakennettavaan tauluun tulee eksponentiaalinen määrä merkintöjä tarkasteltavan piirin koon suhteen, kun taas päätelysäännöillä luotavissa olevien merkintöjen määrä on lineaarinen. Tästä syystä olisikin mielekästä rajoittaa haarauttavan säännön käyttö johonkin luonnolliseen piirin porttien *pieneen* osajoukkoon. Työn ongelmanasettelu on seuraava.

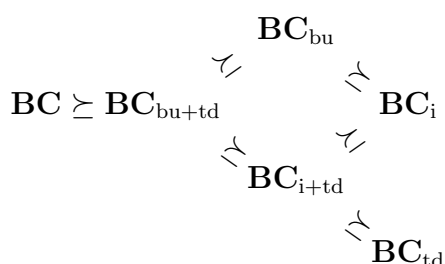
Miten rajoitukset haarauttavan säännön käytölle vaikuttavat hylkäysten kokoon?

Hylkäyksen koko määritellään merkintöjen määräksi hylkäyksessä, missä hylkäyksen juuri lasketaan yhdeksi merkinnäksi. Mittana hylkäyksen koolle käytetään *todistuskompleksisuutta* [2], joka määritellään pienimmän mahdollisen hylkäyksen kooksi tietylle piirille. Todistuskompleksisuuteen pohjautuen voidaan sanoa, että taulumenetelmä T *polynomisesti simuloi* [6] taulumenetelmää T' (merkitään $T \succeq T'$), jos jokaiselle toteutumattomalle Boolean piirille \mathcal{C} pätee, että jos \mathcal{C} :lle on mahdollista rakentaa menetelmässä T' hylkäys kokoa n , niin \mathcal{C} :lle on mahdollista rakentaa menetelmässä T hylkäys, jonka koko on enintään $p(n)$, missä p on polynomi. Jos $T \succeq T'$ pätee, mutta $T' \not\geq T$ ei päde, merkitään $T \succ T'$. Jos sekä $T \succeq T'$ että $T' \succeq T$ eivät päde, merkitään $T \not\approx T'$.

Tässä työssä tarkastellaan seuraavia BC-menetelmän muunnelmia, joissa haarauttavan säännön käyttöä on rajoitettu.

- BC_i : Haarauttavan säännön käyttö on rajoitettu syöteportteihin.
- BC_{td} : Haarauttavan säännön käyttö on rajoitettu tulosportteihin ja sellaisiin portteihin, joiden jollekin isälle on merkintä haarassa.
- BC_{i+td} : Haarauttavan säännön käyttö on rajoitettu syöte- ja tulosportteihin sekä sellaisiin portteihin, joiden jollekin isälle on haarassa merkintä.
- BC_{bu} : Haarauttavan säännön käyttö on rajoitettu syöteportteihin ja sellaisiin portteihin, joiden jollekin lapselle on merkintä haarassa.
- BC_{bu+td} : Haarauttavan säännön käyttö on rajoitettu syöte- ja tulosportteihin sekä sellaisiin portteihin, joiden jollekin lapselle tai isälle on merkintä haarassa.

Suoraviivaisesti rajoitustavan sekä reaktion \succeq transitiivisuuden perusteella saatava polynomiseen simuloitavuuteen perustuva BC-menetelmän muunnelmien välinen järjestys on esitetty kuvassa 3.

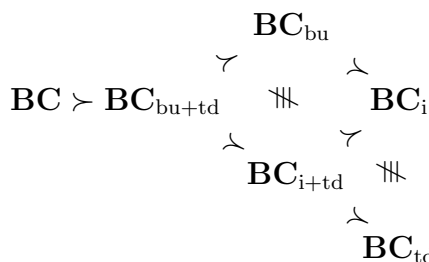


Kuva 3: Suoraviivaisesti nähtävä polynomiseen simuloitavuuteen perustuva BC-menetelmän muunnelmien välinen järjestys.

5 Tulokset

Tulokset osoittavat, että jokainen tarkasteltu tapa rajoittaa haarauttavan säännön

käyttöä johtaa eksponentiaaliseen todistuskompleksisuuden kasvuun. Lisäksi kaikkien haarauttavan säännön suhteen rajoitettujen menetelmien välillä on eksponentiaalisia eroja todistuskompleksisuuden suhteen. Yhteenveto tuloksista on esitetty kuvassa 4.



Kuva 4: Yhteenveto työn tuloksista. Selkeyden vuoksi kuvasta puuttuu merkintä $BC_{bu} \neq BC_{td}$.

On huomattavaa, että tulokset pätevät DPLL-menetelmään yleisesti käytetyllä *Tseitinin käännöksellä* [20] Boolean piireistä tuotettujen konjuktiivisessa normaalimuodossa olevien lauseiden osalta. Näin ollen työn tulokset osoittavat, että harkitut luonnolliset, paikalliset haarauttavan säännön rajoitukset DPLL-menetelmään perustuvissa toteutuvuustarkastimissa kasvattavat pahimmassa tapauksessa todistuksen kokoa eksponentiaalisesti, toisin kuin usein on oletettu kokeellisiin tuloksiin vedoten [19, 9]. Tulokset perustuvat tiettyihin piirikonstrukcioihin, kuten kyyhkyslakkaperiaatteen kuvaukseen Boolean piirinä, sekä BC-menetelmän *resoluutio rajoittuneisuuteen*. Todistukset tuloksille löytyvät kirjoittajan diplomityöstä [11].

6 Yhteenveto

Työssä tutkitaan, miten haarauttavan säännön käytön rajoittaminen vaikuttaa todistusten pituuksiin Boolean piirin toteutuvuustarkastuksessa. Tulokset osoittavat, että kaikki tarkastellut rajoitustavat aiheuttavat pahimmassa tapauksessa pienimmän mahdollisen todistuksen kokoon eksponentiaalisesti kasvun. Myös rajoi-

tustapojen välillä osoitetaan olevan eksponentiaalisia eroja. Tulokset osoittavat, että paikalliset haarauttavan säännön rajoitukset DPLL-menetelmään perustuvissa toteutuvuustarkastimissa kasvattavat pahimmassa tapauksessa todistuksen kokoa eksponentiaalisesti.

Viitteet

- [1] R.J. Bayardo, R.C. Schrag. Using CSP look-back techniques to solve real-world SAT instances. *Proc. 14th National Conference on Artificial Intelligence*, s. 203–208, 1997.
- [2] P. Beame, T. Pitassi. Propositional proof complexity: past, present, and future. *Bulletin of the EATCS*, 65:66–89, 1998.
- [3] A. Biere and W. Kunz. SAT and ATPG: Boolean engines for formal hardware verification. *Proc. 20th IEEE/ACM International Conference on Computer Aided Design*. IEEE Press, 2002.
- [4] E. Clarke, A. Biere, R. Raimi, Y. Zhu. Bounded model checking using satisfiability solving. *Formal Methods in System Design*, 19(1):7–34, 2001.
- [5] S.A. Cook. The complexity of theorem-proving procedures. *Conference record of 3rd annual ACM Symposium on Theory of Computing*, s. 151–158. ACM Press, 1971.
- [6] S.A. Cook, R.A. Reckhow. The relative efficiency of propositional proof systems. *Journal of Symbolic Logic*, 44(1):36–50, 1979.
- [7] M. D’Agostino, D.M. Gabbay, R. Hähnle, J. Posegga. *Handbook of Tableau Methods*. Kluwer Academic Publishers, 1999.
- [8] M. Davis, G. Logemann, D. Loveland. A machine program for theorem proving. *Communications of the ACM*, 5(7):394–397, 1962.
- [9] E. Giunchiglia, A. Massarotto, and R. Sebastiani. Act, and the rest will follow: Exploiting determinism in planning as satisfiability. *Proc. 15th National Conference on Artificial Intelligence and of the 10th Conference on Innovative Applications of Artificial Intelligence*, s. 948–953. AAAI Press, 1998.
- [10] J. Gu, P.W. Purdom, J. Franco, B.W. Wah. Algorithms for the satisfiability (SAT) problem: a survey. *Satisfiability Problem: Theory and Applications, DIMACS: Series in Discrete Mathematics and Theoretical Computer Science*, 35, s. 19–152. AMS, 1997.
- [11] M. Järvisalo. Proof Complexity of Cut-Based Tableaux for Boolean Circuit Satisfiability Checking. Research Report A90, Laboratory for Theoretical Computer Science, Helsinki University of Technology, 2004.
- [12] H. Kautz and B. Selman. Planning as satisfiability. *Proc. 10th European Conference on Artificial Intelligence*, s. 359–363, 1992.
- [13] T. Larrabee. Test pattern generation using Boolean satisfiability. *IEEE Transactions on Computer-Aided Design*, 11(1):6–22, 1992.
- [14] C.M. Li, Anbulagan. Heuristics based on unit propagation for satisfiability problems. *Proc. 15th International Joint Conference on Artificial Intelligence*, 1997.
- [15] M.W. Moskewicz, C.F. Madigan, Y. Zhao, L. Zhang, S. Malik. Chaff: Engineering an efficient SAT solver. *Proc. 38th Design Automation Conference*, s. 530–535. ACM, 2001.
- [16] A. Nerode and R.A. Shore. *Logic for Applications*. Springer, USA, 1997.
- [17] C.H. Papadimitriou. *Computational Complexity*. Addison-Wesley, USA, 1994.
- [18] Bart Selman, Henry Kautz, Bram Cohen. Local search strategies for satisfiability testing. *Second DIMACS implementation challenge: cliques, coloring and satisfiability, DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, 26, s. 521–532. AMS, 1996.
- [19] O. Shtrichman. Tuning SAT checkers for Bounded Model Checking. *Computer Aided Verification; 12th International Conference, LNCS 1855*, s. 480–494, Springer, 2002.
- [20] G.S. Tseitin. On the complexity of derivation in propositional calculus. *Automation of Reasoning 2: Classical Papers on Computational Logic 1967-1970*, s. 466–483. Springer, 1983.
- [21] L. Zhang, S. Malik. The quest for efficient Boolean satisfiability solvers. *CADE-18, LNCS 2392*, s. 295–313. Springer, 2002.

Paneeli

Uudet tuulet - minne menee tietojenkäsittelytiede?

Erkki Sutinen, pj.
Joensuun yliopisto

Workshopit

Tutkijakoulujen yhteistoiminnan avaaminen

Tietojenkäsittelytieteen tutkijakoulut 2001-2003

Pasi Fränti
Joensuun yliopisto

Elämänkaari-workshop

Tohtorius Naisille Lasikatto?

Tarja Tiainen
Tampereen yliopisto

Miksi tietojenkäsittelytieteessä on niin vähän naisia?

Wilhelmiina Hämäläinen
Joensuun yliopisto

Tietojenkäsittelytieteen seuran toimintamuotojen kehittäminen

Paneeli: "Uudet tuulet - minne menee tietojenkäsittelytiede?"

Puheenjohtaja: Prof. Erkki Sutinen

Joensuun yliopisto

Tietojenkäsittelytieteen laitos

Erkki.Sutinen@joensuu.fi

Päivien avauspaneelissa kysytään, minne tietojenkäsittelytiede on menossa.

Ongelmanasettelu heijastelee monissa laitoksissa käytyä keskustelua tietojenkäsittelytieteen ydinalueesta suhteessa sen moniin sovelluksiin ja niihin liittyvään tutkimukseen.

Samalla teema liittyy uudistuviin tutkimintoihin ja muuttuvien opetussuunnitelmien painotuksiin.

Tarkoitus on, että panelistit tuovat -provosoivastikin - esille omia näkemyksiään oppiaineemme sisällöstä ja sen kehitystrendeistä.

Paneelissa pyritään ylittämään perinteinen asiantuntija-yleisö -vastakkainasettelu ja saamaan koko salintäysi yhteiseen työskentelyyn.

Tällä tavalla interaktiivisen paneelin työskentelytapa edustaa toivon mukaan tietojenkäsittelytieteen uusia tuulia itsekin.

Tietojenkäsittelytieteen tutkijakoulut 2001-2003

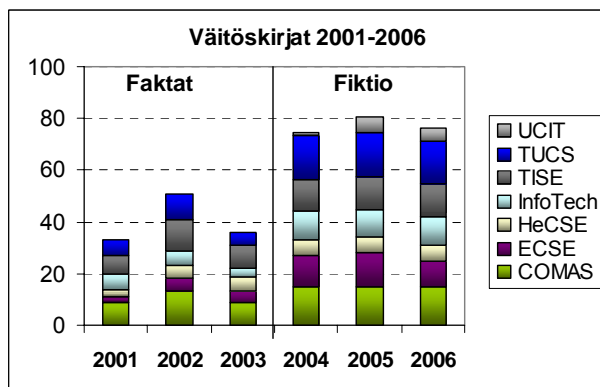
Pasi Fränti

Tietojenkäsittelytieteen laitos, Joensuun yliopisto

etunimi.sukunimi@cs.joensuu.fi

Päivien ohjelmaan on merkitty tutkijakoulujen yhteistoiminnan avaamiseen tarkoitettu *workshop*. Tämän tiivistelmän tarkoitus on koostaa lyhyesti muutamia faktoja tämän alustukseksi. Rajasin tutkijakoulut seuraaviin:

COMAS	Graduate school in Computing and Mathematical Sciences http://www.it.jyu.fi/comas/index.html
ECSE	Itä-Suomen korkeakoulujen tietotekniikan tutkijakoulu http://www.it.lut.fi/opiskelu/ecse/
HeCSE	Helsingin tietojenkäsittelytieteen ja -tekniikan tutkijakoulu http://www.cs.helsinki.fi/hecse/
InfoTech	Infotech Oulu tutkijakoulu http://www.infotech.oulu.fi/
TISE	Tietojenkäsittelytieteen ja -tekniikan tutkijakoulu http://www.cs.tut.fi/tise/
TUCS	Turku Centre for Computer Science http://www.tucs.fi/
UCIT	Käyttäjakeskeisen tietotekniikan tutkijakoulu http://www.cs.uta.fi/ucit/



Kuva 1. Tutkijakouluissa suoritettujen väitöskirjain (aiheenrajauksen mukaisesti).

Kuvassa 1 on kooste tietojenkäsittelytieteen tutkijakouluissa 2001-2003 suoritetuista väitöskirjoista (faktat), sekä koulujen tavoitteet vuosille 2004-2006 (fiktio). Kuten tuloksista näkee, tämän tarkastelun perusteella vuonna 2004 olisi tulossa voimakas kasvupiikki väitöskirjojen lukumäärässä.

Taulukkoon 1 on koostettu väitöskirjojen määrä vuosina 2001-2003 verrattuna tutkijakoulupaikkoihin vuonna 2003. Vertailu näillä luvuilla ei ehkä ole paras mahdollinen, mutta käyköön tähän tarkoitukseen. Toisaalta tutkijakoulupaikkojen määrät eivät UCIT:a lukuun ottamatta ole merkittävästi muuttuneet tarkasteluajanjaksona.

Liitteessä 1 on listattu kyselyn tuloksena tai muusta lähteestä ongitut tutkijakoulujen vahvat painopistealueet. Liitteessä 2 on kaikki tutkijakouluissa vuosina 2001-2003 suoritettujen laskuissa mukana olevat väitöskirjat.

Koska tähän tiivistelmään on oletettavasti jäänyt lukuisia virheitä, päivitetty versio on [wepissä](http://www.cs.joensuu.fi):

<http://www.cs.joensuu.fi/pages/franti/tutkijakoulut2004.htm>

Taulukko 1. Tutkijakoulupaikat vuonna 2003 ja väitöskirjat 2001-2003.

	Tutk.koul. paikat	Väitöskirjat
COMAS	14	31
TISE	36	28
TUCS	28	21
InfoTech	30	15
HeCSE	23	14
ECSE	10	11
UCIT	12	0*

* UCIT aloitti toimintansa vasta vuoden 2002 alussa.

Liite 1: Tutkijakoulujen vahvuudet

COMAS

- Tietojärjestelmätiede
- Ohjelmisto- ja informaatiotekniikka
- Tilastollinen analyysi ja laskennallinen tilastotiede
- Tieteellinen laskenta ja optimointi
- Kognitiotiede ja ihmislähtöinen informaatioteknologia.

ECSE

- Oppivat ja älykkäät järjestelmät
- Opetusteknologia
- Ohjelmistotekniikka
- Tietoliikennetekniikka

HeCSE

- Oppivat ja älykkäät järjestelmät
- Tietojenkäsittelyteoria
- Tulevaisuuden tietoverkkojen ohjelmistotekniikka
- Core Computer Science

InfoTech

- Tietojenkäsittelytiede ja -tekniikka
- Ohjelmistotekniikka ja tietojärjestelmät

TISE

- Tietojenkäsittelyoppi
- Signaalinkäsittely
- Digitaal- ja tietokonetekniikka
- Ohjelmistotekniikka
- Tietoliikennetekniikka

TUCS

- Reliable Software
- Advanced Management Systems
- Algorithmics
- Discrete Mathematics for Inf. Tech.
- Information Systems Laboratory
- Network Economics Institute

Liite 2: Valmistuneet väitöskirjat

2003

COMAS:

Honkaranta A: From genres to content analysis in organizations. JY.

Hämäläinen S: WCDMA radio network performance, JY.

Kärkkäinen S: Orientation analysis of stochastic fibre systems with an application to paper research, JY.

Markkula J: Geographic personal data, its privacy protection and prospects in a location-based service environment, JY.

Niemelä M: Visual search in graphical interfaces, JY.

Pekkola S: Multiple media in group work: emphasizing individual users in distributed and real-time CSCW systems, JY.

Saalasti S: Neural networks for heart rate time series analysis, JY.

Raitamäki J: An approach to linguistic pattern recognition using fuzzy systems, JY.

Taskinen S: On nonparametric tests of independence and Robust Canonical Correlation Analysis, JY.

ECSE:

Mielikäinen Jarno: Lossless Compression of Hyperspectral Images, Computer Science, LTY.

Kämäräinen Joni-Kristian: Feature Extraction using Gabor Filters, Computer Science, LTY.

Kolesnikov Alexander: Efficient Algorithms for Vectorization and Polygonal Approximation, Computer Science, JoY.

Smolander Kari: On the Role of Architecture in Systems Development, Computer Science, LTY.

HeCSE:

Campadello Stefano: Middleware Infrastructure for Distributed Mobile Applications, Computer Science, HY.

Helin Heikki: Supporting Nomadic Agent-based Applications in the FIPA Agent Architecture, Computer Science, HY.

Kurhila Jaakko: Considering Individual Differences in Computer-Supported Special and Elementary Education, Computer Science, HY.

Lindström Jan: Optimistic Concurrency Control Methods for Real-Time Databases, Computer Science, HY.

Mäkinen Veli: Parameterized Approximate String Matching and Local Similarity Based Point-Pattern Matching, Computer Science, HY.

Taina Juha: Design and Analysis of a Distributed Database Architecture for IN/GSM Data, Computer Science, HY.

InfoTech:

Hooli Kari: Equalization in WCDMA Terminals, OY.

Mäenpää Topi: The local binary pattern approach to texture analysis - extensions and applications, OY.

Mäntytjärvi Jani: Sensor-based context recognition for mobile applications, OY.

Niskanen Matti: A visual training based approach to surface inspection, OY.

Tujkovic Djordje: Space-time turbo coded modulation for wireless communication systems, OY.

Ulkuniemi Pauliina: Purchasing software components at the dawn of market, OY.

TISE:

Kuusilinna Kimmo: Studies and Implementations of Bus-based Interconnections, Digitaali- ja tietokonetekniikka, TTKK.

Hännikäinen Marko, Design of Quality of Service Support for Wireless Local Area Networks, Digitaali- ja tietokonetekniikka, TTKK.

Tammi Mikko, Techniques for Low Bit Rate Speech Coding – Speech Modeling in WI and RCELP Coders, Digitaali- ja tietokonetekniikka, TTY.

Vasilache Adriana, On Vector Quantization with Regular Geometric Structure, Signaalinkäsittely, TTY.

Bregovic Robert, Optimal Design of Perfect-Reconstruction and Nearly Perfect-Reconstruction Multirate Filter Banks, Signaalinkäsittely, TTY.

Gotchev Atanas, Spline and Wavelet Based Techniques for Signal and Image Processing, Signaalinkäsittely, TTY.

Koppinen Konsta, Design, Analysis and Generalization of Polynomial Predictive Filters, Signaalinkäsittely, TTY.

Tohka Jussi, Global Optimization-Based Deformable Meshes for Surface Extraction from Medical Images, Signaalinkäsittely, TTY.

Karsisto Konsta, A New Parallel Composition Operator for Verification Tools, Ohjelmistotekniikka, TTY.

Mykkänen Jouni, Delineation of Brain Structures from Functional Positron Emission Tomography Images, Tietojenkäsittelyoppi, TaY.

Hyyrö Heikki, Practical Methods for Approximate String Matching, Tietojenkäsittelyoppi, TaY.

TUCS:

Timo Lainema: Enhancing Organizational Business Process Perception – Experiences from Constructing and Applying a Dynamic Business Simulation Game, Business Gaming, TKKK.

Mika Hirvensalo: Studies on Boolean Functions Related to Quantum Computing. Quantum Computing, TY.

Pentti Virtanen: Measuring and Improving Component-Based Software Development, Software Analysis, TY.

Adekunle Okunoye: Knowledge Management and Global Diversity - A Framework to Support Organisations in Developing Countries. Knowledge Management, TY.

Antonina Kloptchenko. Text Mining Based on the Prototype Matching Method. Data Mining, AA.

2002

COMAS:

Akimov V.V.. Domain decomposition methods for the problems with boundary layers. JY.

Forsell M: Improving component reuse in software development. JY.

Hämäläinen T: Broadband network quality of service and pricing. JY.

Kovalainen M: Computer mediated organizational memory for process control. moving CSCW research from an idea to a product. JY.

Lensu A: Computationally intelligent methods for qualitative data analysis JY.

Martikainen J: Efficient solvers for discretized elliptic vector-valued problems. JY.

Mursu A: Information systems development in developing countries: risk management and sustainability analysis in nigerian software companies. JY.

Ollila E: Sign and rank covariance matrices with applications in multivariate analysis, Mathematics and Statistics, JY.

Ryabov V: Handling imperfect temporal relations. JY.

Seiukova-Rivkind L: Mathematical and numerical analysis of free boundary problems of the flow problems in unbounded domains. JY.

Seleznyov A: An anomaly intrusion detection system based on intelligent user recognition. JY.

Tsymbal A: Dynamic intergation of data mining methods in knowledge discovery systems. JY.

Virtanen P: Neuro-fuzzy systems in financial and control engineering. JY.

ECSE:

Kauranne Tuomo: Introducing Parallel Computers into Operational Weather Forecasting, Applied Mathematics, LTTK.

Lensu Lasse: Photoelectric Properties of Bacteriorhodopsin Films for Photosensing and Information Processing, Computer Science, LTTK.

Kyrki Ville: Local and Global Feature Extraction for Invariant Object Recognition, Computer Science, LTTK.

TKKtunen Pentti: Data-Parallel Computation in Parallel and Distributed Environments, Data Communications, LTTK.

Eriksson-Bique Stephen: An Algebraic Theory of Multidimensional Arrays, Computer Science, JoY.

HeCSE:

Heljanko Keijo: Combining Symbolic and Partial Order Methods for Model Checking 1-Safe Petri Nets ,Computer Science & Engineering, TKK.

Hoyer Patrick: Probabilistic Models of early Vision, Computer Science, HY.

Lokki Tapio: Physically-based Auralization - Design, Implementation, and Evaluation, Computer Science & Engineering, TKK.

Ollikainen Vesa: Simulation Techniques for Disease Gene Localization in Isolated Populations, Computer Science, HY.

Tuovinen Antti-Pekka: Object-Oriented Engineering of Visual Languages, Computer Science, HY.

InfoTech:

Katz Marcos: Code acquisition in advanced CDMA networks, OY.

Martinkauppi Birgitta: Face colour under varying illumination - analysis and applications, OY.

Repo Tapio: Modeling of structured 3-D environments from monocular image sequences, OY.

Abrahamsson Pekka: The role of commitment in software process improvement, OY.

Tuikka Tuomo: Towards computational instruments for collaborating product concept designers, OY.

Siponen Mikko: Designing secure information systems and software: Critical evaluation of the existing approaches and a new paradigm, OY.

TISE:

Hamila Ridha: Synchronization and Multipath Delay Estimation Algorithms for Digital Receivers, Tietoliikennetekniikka, TTKK.

Burian Adrian: Signal and Image Restoration Using the Median Cost Function or Phase Retrieval, Signaalinkäsittely, TTKK.

Yli-Kaakinen Juha: Optimization of Digital Filters for Practical Implementations, Signaalinkäsittely, TTKK.

Huupponen Eero: Advances in the Detection of Sleep EEG Signal Waveforms, Signaalinkäsittely, TTKK.

Rakkolainen Ismo: Novel Applications and Methods for Virtual Reality, Signaalinkäsittely, TTKK.

Tiainen Tarja: Information System Specialist Predispositions, Tietojenkäsittelyoppi, TaY.

Siermala Markku: Local Prediction of Secondary Structures of Proteins from Viewpoints of Rare Structure, Tietojenkäsittelyoppi, TaY.

Isomäki Hannakaisa: The Prevailing Conceptions of the Human Being in Information Systems Development: Systems Designer's Reflections, Tietojenkäsittelyoppi, TaY.

Viikki Kati: Machine Learning on Otoneurological Data: Decision Trees for Vertigo Diseases, Tietojenkäsittelyoppi, TaY.

Helenius Marko: A System to Support the Analysis of Antivirus Products' Virus Detection Capabilities, Tietojenkäsittelyoppi, TaY.

Aho Isto: Interactive Knapsacks: Theory and Applications, Tietojenkäsittelyoppi, TaY.

TUCS:

Lauri Heikkilä: Electroluminescence in Si-SiO₂ layer structures. Microelectronics, TY.

Nelly Noykova: Modelling and identification of microbial population dynamics in wastewater treatment. Biomathematics, TY.

Jouni Smed: Production Planning in Printed Circuit Board Assembly. Industrial Optimization, TY.

Vesa Halava: The Post Correspondence Problem for Marked Morphisms. Theoretical Computer Science, TY.

Ion Petre: Commutation Problems on Sets of Words and Formal Power Series. Theoretical Computer Science, TY.

Vladimir Kvassov: Information Technology and the Productivity of Managerial Work, IT Management, ÅA.

Franck Tétard: Managers, Fragmentation of Working Time, and Information Systems, IT Management, ÅA.

Jan Manuch: Defect Theorems and Infinite Words. Theoretical Computer Science, TY.

Kalle Ranto: Z₄-Goethals Codes, Decoding and Designs. Coding Theory, TY.

Arto Lepistö: On Relations between Local and Global Periodicity, Theoretical Computer Science, TY.

2001

COMAS:

Fomin Vladislav V: The process of standard making. The case of cellular mobile technology, JY.

Hirvonen Kullervo: Towards better employment using adaptive control of labour costs of an enterprise, JY.

Häkkinen Erkki: Design, implementation and evaluation of neural data analysis environment.

Majava Kirsi: Optimization-based techniques for image restoration, JY.

Nahar Nazmun: Information technology supported technology transfer process. A multi-site case study of high-tech enterprises, JY.

Päivärinta Tero: A genre-based approach to developing electronic document management in the organization, JY.

Saarinen Kari: Near infra-red measurement based control system for thermo-mechanical refiners, JY.

Smolianski Anton: Numerical modeling of two-fluid interfacial flows, JY.

Taskinen Ilkka: Cluster priors in the Bayesian modelling of fMRI data. Mathematics and Statistics, JY.

ECSE:

Ikonen Jouni: Improving Distributed Simulation in a Workstation Environment, Data Communications, LTTK.

Tukiainen Markku: Developing a New Model of Spreadsheet Calculation: a Goals and Plans Approach, Computer Science, JoY.

HeCSE:

Rousu Juho: Efficient Range Partitioning in Classification Learning, Computer Science, HY.

Peura Markus: Attribute Trees as Adaptive Object models in Image Analysis, Neuroverkkojen tutkimusyksikkö, TKK.

Salmenkivi Marko: Computational Methods for Intensity Models, Computer Science, HY.

InfoTech:

Frantti Tapio: Timing of fuzzy membership functions from data, OY.

Jokela Timo: Assessment of user-centred design processes as a basis for improvement action. An experimental study in industrial settings, OY.

Saarinen Ilkka: Reverse link feedback power control in pilot symbol assisted systems, OY.

TISE:

Giurcaneanu Ciprian Doru: On Lossless Signal Compression, Signaalinkäsittely, TTKK.

Linne Marja-Leena: Computational Model for Granule Neuron Excitability, Signaalinkäsittely, TTKK.

Tico Marius: On Design and Implementation of Fingerprint-Based Biometric Systems, Signaalinkäsittely, TTKK.

Katara Mika: Aspects of Continuous Behaviour – Design of Real-Time Reactive Systems, Ohjelmistotekniikka, TTKK.

Laurikkala Jorma: Knowledge Discovery for Female Urinary Incontinence Expert Systems, tietojenkäsittelyoppi, TaY.

Niemi Tapio: Methods for Logical OLAP Design, Tietojenkäsittelyoppi, TaY.

TUCS:

Kalle Parvinen: Adaptive metapopulation dynamics. University of Turku, Applied Mathematics, Biomathematics, TY.

Jussi Puhakainen: Electronic Business in Interactive Digital Networks - from Transactional toward Interactive Focus, Electronic Business, TKKK.

Timo Leino: Itsenäiskäytön johtaminen tietohallinnon osa-alueena. IT Management, TKKK.

Tero Aittokallio: Characterization and Modelling of the Cardiorespiratory System in Sleep-disordered Breathing. University of Turku, Mathematical Modelling, TY.

Ivan Porres: Modeling and Analyzing Software Behavior in UML. Software Analysis, ÅA.

Mauno Rönkkö. Stepwise Development of Hybrid Systems. System Development, ÅA.

Tohtorius Naisille Lasikatto?

Tarja Tiainen

Tampereen yliopisto, Tietojenkäsittelytieteiden laitos, Epanet,

tarja@cs.tampere.fi

1. Tiivistelmä

Vaikka nykyisin suomalaisista yliopisto-opiskelijoita on enemmistö naisia, akateeminen ura ajatellaan miehisenä alueena. Akateemisen maailman lisäksi myös tekniikkaan liitetään miehisyyteen. Nämä kaksi aluetta yhdistyvät teknologian tutkimuksessa, joten esimerkiksi tietojenkäsittelytieteiden laitoksella miehisyyttä otetaan itsestään selvyytenä eikä naisten vähäisyyttä henkilökunnassa ihmetellä.

Pätevien naisten miehistä poikkeavan kohtelun kiistaton todistaminen on tuskin mahdollista, koska yliopistolla virkoihin sijoittuminen on monimutkainen sosiaalisesti muotoutuva prosessi. Kuitenkin kun katsomme, miten naisia ja miestohtorit sijoittuvat laitoksien virkoihin, on selvä erilaisuus havaittavissa. Esimerkiksi sopii Tampereen yliopiston tietojenkäsittelytieteiden laitoksella (tkt-laitoksella) väitelleiden sijoittumista kyseisen laitoksen virkoihin. Tkt-laitoksella on tarkastettu 30 väitöskirjaa; väittelijöissä on 22 miestä ja 8 naista. Tkt-laitoksella on töissä 22 tohtoria; heistä miehiä on 19 ja naisia 3. Miehistä 6 on väitellyt muualle kuin tkt-laitokselle; yhtään muualla väitellyttä naista ei ole tkt-laitoksella töissä.

2. Tausta

Suomalaiset yliopistot ovat muodollisesti sukupuolineutraaleja, mutta käytännössä yhteiskunnan sukupuolirakenne on näkyvissä myös yliopistoissa ja niissäkin sitä uusinnetaan jatkuvasti [Husu 2001]. Vaikka suomalaisista yliopisto-opiskelijoita on nykyisin enemmistö naisia, akateeminen ura ajatellaan miehisenä alueena. Naiset eivät suorita pelkästään perustutkintoja (maisterintutkintoa), vaan tohtorintutkinnon suorittajista 45 % on naisia. Kuitenkin yliopistossa naiset sijoittuvat alimpiin virkoihin: tuntiopettajista 82 % on naisia ja assistenteista 52 %, mutta professoreista vain 20 % [Taulukoista Kota-tietokannasta 2002, s. 45].

Akateemisen maailman lisäksi miehisyyttä liitetään myös tekniikkaan [katso esim. Lie 1995; Grint & Gill 1995; Vehviläinen 1999]. Nämä kaksi miehisyyteen liitettyä aluetta yhdistyvät teknologian tutkimuksessa, josta on seurauksena se, että esimerkiksi tietojenkäsittelytieteiden laitoksella miehisyyttä otetaan itsestään selvyytenä.

Kuvaan tässä paperissa, miten teknologian akateemisen maailman maskuliinista sukupuolta uusinnetaan. Tarkastelen tilannetta käyttämällä esimerkkinä Tampereen yliopiston tietojenkäsittelytieteiden laitoksella (tkt-laitoksella) väitelleiden sijoittumisista

töihin samalle laitokselle. Jokaisessa yksittäisessä viranhoidotapauksessa voitaneen perustella oikeutus valitun henkilön valintaan; kuitenkin lopputulos on voimakkaasti sukupuolisesti vinoutunut henkilöstörakenne. Kuvaan tätä vinoumaa esittämällä tilaston tkt-laitoksella väitelleiden sijoittumisesta laitoksen virkoihin ja avaamalla tilastoon liittyviä keskusteluja.

3. Tapaus tkt-laitos

Vaikka tkt-laitoksella ei ole kovin paljoa naisopiskelijoita (osuus on useina vuosina ollut alle 10%), naiset ovat innokkaita ja tuloksellisia jatko-opiskelijoita: laitoksen 30 väitöskirjasta 8 on naisen tekemää (naisten osuus tohtoreista on 27 %).

Taulukko 1: Nais- ja miestohtorien valmistuminen ja työtilanne kevätlukukaudella 2004 tkt:ssa

	Naiset	Miehet
Tohtoroituminen tkt:ssä	8	22
Osuus tkt-tohtoreista	27%	73%
Tkt-tohtoreista töissä tkt:ssä	3	14
Muualla tohtoroituneista töissä tkt:ssä	0	6
Tohtoreita töissä tkt:ssä	3	20
Osuus tkt:ssa työskentelevistä tohtoreista	13%	87%

Naisten ahkera tohtorintutkintojen suorittaminen ei kuitenkaan näy tkt-laitoksen virantäytöissä ja -hoidoissa. Naisten osuus tkt-laitoksella työskentelevistä tohtoreista on vain 13%. Osin suuri ero naisten tohtorin tutkinnon suorittamiseen (27% tutkinnonsuorittaneista) ja työskentelystä tkt-laitoksella (naisia 13% tohtoreista) selittyy siitä, että tkt-laitokselle on palkatta 6 muualla tohtorin tutkinnon suorittanutta

miestä, mutta ei yhtään muualla tohtorin tutkinnon suorittanutta naista.

4. Selityksiä vinoumaan

Kun katsomme tilastoa väitelleistä, ei naisten vähäisen osuuden selittäjäksi kelpaa se, että ei olisi päteviä naisia. Tämän väitteen virheellisyys tulee korostuneesti esille, kun katsomme, millaisiin paikkoihin tkt-laitokselle väitelleet naiset ovat sijoittuneet töihin: kolme heistä on professoreina muualla kuin tkt-laitoksella. Pätevien naisten puutteen lisäksi on esillä kaksi muuta selitystä: naisten oma valinta ja tkt-laitoksen politiikka. Käsittelen näitä seuraavassa.

4.1 Naisten omat valinnat

Kun tkt-laitoksella nostan keskusteluun naistohtorien vähäisyyden henkilökunnassa, niin selitykseksi annetaan nimenomaan naisten tekemät yksilölliset valinnat. Mainittu naistohtorien tekemä valinta on hakeutuminen muualle töihin. Tkt-laitoksella näyttää vallitsevan ajattelutapa, jossa sukupuolirakenne on sivuutettu joko tietoisesti tai ymmärryksen puutteen takia. Samanlainen sukupuolirakenteen olemassaolon sivuuttava ajattelutapa näyttäisi vallitsevan laajemminkin tietojenkäsittelyalalla [katso esim. Vehviläinen 1994; Kuosa 2000].

Yksittäisen naistohtorin kannalta sijoittumattomuus tkt-laitokselle riippuu paitsi tkt-laitoksesta ja sen houkuttelevuudesta niin myös tarjolla olevista muista työpaikoista. Perinteistä akateemisesta tilanteesta poiketen tietojenkäsittelyopin tohtoreille on tarjolla työpaikkoja myös akateemisen maailman ulkopuolella. Osa tietojenkäsittelyopin tohtoreista työskentelee teollisuudessa koko väitöskirjaprosessin ajan ja jatkaa tohtoroitumisen jälkeen entistä työuraansa. Tkt-laitoksen naistohtoreista kukaan ei ole valinnut teol-

lisuusuraa. Osalle naistohtoreista muut yliopistot ovat kuitenkin näyttäneet houkuttelevina vaihtoehtoina; mutta onko tämä naisten yksilöllinen valinta vai tkt-laitoksella vallitseva vinouma?

4.2 Tkt-laitos sukupuolisen viinon uusinajana

Yliopistolla virkoihin sijoittuminen on monimutkainen sosiaalisesti muotoutuva prosessi. Yleensä yliopistovirat, etenkin professuurit, ajatellaan pitkäaikaisina ja hoitajalleen pysyviksi. Tkt-laitos poikkeaa kuitenkin tästä mallista. Laitos on kasvanut voimakkaasti 1990-luvulla ja sinne on perustettu uusia vakinaisia ja määräaikaisia (1- ja 2-vuotisia) virkoja. Uusien virkojen alojen määrittely on sosiaalinen prosessi, joka määrittää sitä, keitä voidaan pitää pätevänä kyseiseen virkaan ja ketkä voivat määrittellä hakijoiden pätevyyden. Tätä akateemisessa maailmassa noudatettavaa käytäntöä perustellaan pyrkimyksellä objektiivisuuteen, kuitenkin se antaa myös mahdollisuuden ”hyvä veli” -toimintaan, kuten Suomen tietotekniikan alkuaikoina 1960-luvulla oli tapana [Vehviläinen 1996].

Tkt-laitoksen viroista monet ovat väliaikaisessa hoidossa viran ollessa avoina tai virkaan valitun hoitaessa jotain muuta virkaa. Määräaikaisiin virkoihin ja viranhoitoihin laitoksen johtaja voi tehdä nimityksen omalla päätöksellään, kuitenkin käytännössä laitoksen johtaja usein neuvottelee tilanteesta laitoksen vakinaisten professorien kanssa (jotka kaikki ovat miehiä).

Tkt-laitoksen kannalta laitoksen sukupuolen muotoutuminen ei ole tietoista aktiivista toimintaa. Sukupuoli muotoutuu arkisten päätösten yhteydessä, jolloin neuvottelun kohteena on sukupuolen asemasta mm tkt-laitoksella opetettävien eri suuntautumisalojen

asema. Sukupuolen lisäksi selitystä naistohtorien sijoittumattomuuteen tkt-laitokselle voi hakea väitöskirjojen aloista (ja eri alojen asemasta laitoksella).

Lähteet

- Grint, Keith and Gill, Rosalind (Eds.) (1995), *The Gender-Technology Relation, Contemporary Theory and Research*, Taylor & Francis, p. 192-209.
- Husu, Liisa (2001), Sexism, Support and Survival in Academia. Academic Women and Hidden Discrimination in Finland. Department of Social Psychology, University of Helsinki.
- Kuosa, Tarja (2000), Masculine World Disguised as Gender Neutral. In Balka, Ellen and Smith, Richard (Eds.), *Women, Work and Computerization*. Kluwer Academic Publishers, Boston, USA. Proceedings of IFIP TC9 WG9.1 17th Conference on WWC, 8 - 11.6.2000, Vancouver, British Columbia, Canada, pp. 119-126.
- Lie, Merete (1995), Technology and Masculinity: The Case of Computer. *The European Journal of Women's Studies*, 2:3, August, pp. 379-394.
- Taulukoita KOTA-tietokannasta (2002), Opetusministeriö:
<http://www.minedu.fi/julkaisut/pdf/87kota.pdf> (haettu 20.5.2002)
- Vehviläinen, Marja (1994), Reading Computing Professionals' Codes of Ethics - A Standpoint of Finnish Office Workers. In Gunnarsson, Ewa and Trojer, Lena (Eds.), *Feminist Voices on Gender, Technology and Ethics*. Centre for Women's Studies, Lulea University of Technology, Lulea, Sweden, p. 145-161.
- Vehviläinen, Marja (1996), "Maailmoista ilman naisia" Tietotekniikan sukupuolieroihin, kirjassa Kinnunen, Merja ja Korvajärvi, Päivi (toim.), *Työelämän sukupuolista-vat käytännöt*, Vastapaino, Tampere, s. 143-170.
- Vehviläinen, Marja (1999), Gender and Computing in Retrospect: The Case of Finland. *IEEE Annals of History of Computing*, Vol. 21, No. 2, 1999. pp. 44-51.

Miksi tietojenkäsittelytieteessä on niin vähän naisia?

Wilhelmiina Hämäläinen
whamalai@cs.joensuu.fi

1 Johdanto

Miksi tietojenkäsittelytieteessä on niin vähän naisia? Täyää kysymystä on pohdittu vakavasti maailmalla. ACM:n alaisuuteen kuuluva ”Committee on the status of Women in computer science” on kartoittanut syitä, miksi tietojenkäsittelytiede alkujaankin houkuttelee vähemmän naisia ja miksi naiset karsiutuvat akateemisesta uraputkesta miehiä helpommin. Tärkeimmiksi syiksi määriteltiin 1) naisten itsetunto-ongelmat, 2) mentorien ja roolimallien puute, 3) sukupuolinen syrjintä sekä 4) vaikeudet tasapainottaa ura- ja perhevelvollisuuksia. [PPR⁺90] Myös Suomessa naisten osuus tietojenkäsittelytieteen opiskelijoista on vähäinen ja professoritasolla löytyy enää ani harvoja naisia. Kaiken lisäksi uusi tutkimus on osoittanut, että Suomessa koulutusalat ovat paljon tiukemmin sukupuolittuneita kuin EU-maissa keskimäärin [Kur03]. Olisikin syytä kartoittaa, missä määrin ACM:n raportoimat ongelmat pätevät Suomessa ja mitä niiden korjaamiseksi voitaisiin tehdä.

2 Naisten heikompi itsetuottamus

ACM:n raportin mukaan naisten heikompi itsetuottamus on tärkein syy, miksi naiset jättävät Tietojenkäsittelytieteen opiskelun. Tyttöjen heikompi itsetunto kehittyy jo varhain, mutta samat koululuokissa esiinty-

vät ongelmat voivat pahentua miesvaltaisella alalla opiskellessa. Mikäli luokassa on vain yksi tai kaksi naisopiskelijaa, nämä joutuvat helposti muiden huomion kohteiksi. He kokevat, että heidän pitää osoittaa olevansa yhtä hyviä kuin miesopiskelijat ja ansaita kunnioitus. Opettajan ja miesopiskelijoiden vähättelevä asenne voi musertaa epävarman naisopiskelijan itsetunnon lopullisesti. Naisopiskelijoiden itsetuntoa lannistaa myös (niin nais- kuin miesprofessorien) tapaa huomioida enemmän miesopiskelijoita: he muistavat paremmin miesopiskelijoiden nimet, kysyvät näiltä useammin ja haastavampia kysymyksiä. Toisaalta naisopiskelijoita keskeytetään useammin, heidän panoksensa pistetään miesten ansioksi tai jätetään kokonaan huomioimatta. Itsetunto-ongelma voi muuttua vielä merkittävämmäksi jatko-opintovaiheessa, kun opiskelijat saavat palautetta pääasiallisesti vain ohjaajiltaan ja muilta jatko-opiskelijoilta. Perustutkintovaiheessa opiskelijat saavat kuitenkin objektiivisempaa palautetta arvosanojen muodossa. [PPR⁺90, Lev89, GC01]

Tämä on ongelma, johon kuka tahansa Tietojenkäsittelytieteen opettaja ja ohjaaja voi vaikuttaa. Opettajan tulisi kohdistaa samat odotukset kaikkiin opiskelijoihin, antaa yhtäläisesti huomiota ja palautetta heidän työnsä laadusta, ja oppia tiedostamaan erilaiset kommunikointityylit. Lisäksi olisi tärkeää tarjota naisille mahdollisuus ammatilliseen kokemukseen ja houkutella heitä laitoksen tutkimusprojekteihin – ideaalisesti jo

perustutkinnon aikana. [PPR⁺90]

3 Mentorit ja roolimallit

Mentoreilla ja roolimalleilla on yllättävän suuri merkitys naisten värväyksessä perus- ja jatko-opiskelijaksi. Nuoret tytöt tarvitsevat rohkaisevia roolimalleja jo varhain uskaltaakseen hakeutua Tietojenkäsittelytieteen alalle, mutta roolimalleja tarvitaan kaikissa uran vaiheissa. Esimerkiksi laitoksen naistyöntekijät ja vanhemmat opiskelijat voivat olla tärkeitä roolimalleja nuoremmille opiskelijoille. Tyttöjen tulisi kuitenkin saada roolimalleja jo ennen yliopistoa – joko omista opettajista, vierailevia yliopiston naistutkijoista tai luonnontieteissä vaikuttaneiden naisten elämänerkeroista. [PPR⁺90]

Jatko-opintovaiheeseen siirryttäessä mentorilla on hyvin ratkaiseva merkitys jatko-opiskelijan elämässä. Mentorilta saa tietoa ja neuvoja, kuinka hankkia rahoitusta, saada julkaisuja, löytää yhteistyökumppaneita ja päästä sisälle tutkijayhteisöön. Spertus [Spe91] on havainnut, että monen naisopiskelijan ura katkeaa mentorin puutteeseen. Naisopiskelijalla on kyllä virallinen (yleensä miespuolinen) jatko-opintojen ohjaaja, mutta tältä ei välttämättä riitä aikaa ja tukea naisopiskelijalleen. Tärkeimpinä syinä Spertus pitää sitä, että joidenkin miesten vaikea kohdata naisia, he saattavat pitää naisia vähemmän älykkäinä, heikommin sitoutuvina, tai kokonaan sopimattomina akateemiselle uralle, tai he yksinkertaisesti pelkäävät, että läheinen työskentely naisen kanssa voisi herättää epäilyksiä suhteesta.

Erityisesti naisopiskelijoiden mentoroinnista on saatu erittäin hyviä kokemuksia. Mentorin ei välttämättä tarvitse olla oma ohjaaja, vaan mentorointia voi tarjota myös joku muu vanhempi tutkija. Nykyisin mentorointi voi tapahtua myös Internetin tai sähköpostien kautta (ns. "telementoring", [pat01]).

Esimerkiksi Stanfordin yliopistossa toimiva "Big Sister Mentoring Program" [sta] välittää (naispuolisia) mentoreita naisopiskelijoille. Ideaalisinta tietysti olisi, että myös miesprofessorit olisivat valmiita mentoroimaan naisopiskelijoitaan.

4 Sukupuolinen syrjintä

ACM:n raportissa sukupuolinen syrjintä määritellään "holhoavaksi käytökseksi ja oletukseksi, että naiset vähemmän päteviä ja/tai sitoutuneita kuin miehet, riippumatta siitä ovatko oletukset tietoisia vai eivät". Sukupuolista syrjintää Tietojenkäsittelytieteen laitoksella on käsitelty yksityiskohtaisesti MIT:n raportissa "Barriers to Equality in Academia" [bar83]. Raportti vahvistaa muidenkin tutkimusten havainnon, että lievä mutta jatkuva syrjintä voi olla vahingollisempaa kuin satunnainen avoin syrjintä. Ongelmana on se, että lievää syrjintää ei yleensä tunnusteta syrjinnäksi, koska tekoja ei ole tarkoitettu syrjiviksi ja kukin yksittäinen tapaus vaikuttaa vähäpätöiseltä yksinään arvioituna.

Gürerin ja Campin [GC01] raportti mainitsee erityisesti kolme sukupuolisen syrjinnän ilmentymää: Ensinnäkin naistutkijat kärsivät ns. näkymättömyyssyndroomasta: naiset ovat aliedustettuina monissa tärkeissä tehtävissä, kuten alan lehtien toimituskunnissa, konferenssien komiteoissa ja kutsuvieraina, ja alan järjestöjen (kuten ACM:n ja IEEE:n) toimikunnissa. Toiseksi naisten nimitys- ja palkkauskäytännöt, joita on kinnitetty huomiota myös Suomessa. Erityisesti lainsäädännön ulkopuolelle jäävät tutkijoiden apurahat ja määräaikaiset virkasuhteet mahdollistavat naisten syrjinnän yliopistolla [Kur03]. Kolmanneksi suoranainen naisiin kohdistuva häirintä, johon voivat syyllistyä niin naiskuin mieskollegat. Häirintä voi ilmetä vain yleisenä vähättelevänä asenteena naisia koh-

taan, tai avoimemmin sopimattomina vitseinä, epäkorrektina puhutteluna, työympäristön sisustamisella alastonkuvien tai pahimassa tapauksessa jopa seksuaalisena häirintänä. Erityisesti Gürer ja Camp muistuttavat siitä, kuinka tärkeää on mikrotukihenkilöiden kunnioittava käyttäytyminen nuoria ja epävarmoja naisopiskelijoita kohtaan.

5 Uran ja perheen tasapainotus

Monen naisen ura katkeaa – ainakin tilapäisesti – perheen perustamiseen. Usein lastenhankinta ja väitöskirjan laadinta sattuvat yksiin, eikä monellakaan naisella riitä voimia kahteen niin vaativaan projektiin. Suomessa äitiyslomalle on varattu reilusti aikaa, mutta perhevelvollisuudet eivät kuitenkaan pääty lasten syntymään ja nykyisin yhä useammat isät haluaisivat myös osallistua lastenkasvatukseen. Toisaalta mahdollinen äitiysloma tai vain ”siinä iässä oleminen” voi vaikuttaa naistutkijan virkanimityksiin, vaikei kyseinen nainen olisi perhettä suunnitellutkaan. Isien mahdollisuus osallistua lastenhoitoon voi siis vähentää myös lapsettomiin naisiin mahdollisesti kohdistuvaa syrjintää, samalla kun se parantaa perheellisten tietojenkäsittelijöiden elämänlaatua. Joustavat työajat, kotitietokone ja Internet-yhteys sekä mahdollisuus tehdä töitä kotona helpottavat myös perheen ja uran yhteensovittamista. [GC01, Spe91]

6 Muita vaikeuksia

Gürerin ja Campin raportissa luetellaan myös joukko muita ongelmia, jotka vaikeuttavat naisten etenemistä tietojenkäsittelytieteilijän uralla. Edellä mainittujen lisäksi he mainitsevat mm. seuraavat tekijät:

- Tyttöjen/naisten asenteet tietokoneita kohtaan.
- Tyttöjen aiemmat vähäisemmät kokemukset tietokoneiden käytöstä.
- Riittävän varhainen rohkaisu – tyttöjä olisi rohkaistava jo esikoulussa tutustumaan tietokoneisiin.
- Pääasiassa pojille suunnitellut tietokonepelit, jotka ovat usein ensimmäinen kokemus tietokoneista.
- Maskuliiniset työympäristöt opetuksessa ja teollisuudessa.
- Mahdollisuus opiskella ja tutkia vain naisten yhteisössä. Yhdysvalloissa naiskoulujen on havaittu kohentavan tyttöjen ja nuorten naisten itsetuntoa matematiikassa, luonnontieteissä ja insinöörityeissä, mikä puolestaan tukee urakehitystä.
- Tasavertaiset mahdollisuudet käyttää tietokoneita: tyypillisesti pojat dominoivat koulujen tietokoneiluokissa.
- Tutkijakoulut: naisten osuus romahtaa siirryttäessä maisteriopinnoista tohtoriopintoihin. Tutkijakoulujen tulisi pannaostaa erityisesti naisopiskelijoiden värväykseen ja jäämiseen alalle.
- Yhteiskunnassa vallitsevat arvot, joiden mukaan tietotekniikka sopii vain miehille.

Lisäksi Spertus [Spe91] on käsitellyt maskuliinista kielenkäyttöä, joka on ongelma varsinkin englannin kielessä (she/he). Myös suomenkielen terminologiassa puhutaan esimerkiksi ”isä- ja poikasolmuista”, vaikka kekseliäs opiskelija muuttaakin ne puheessaan vaikka ”emoiksi” ja ”lapsiksi”.

7 Ratkaisuja?

Gürer ja Camp [GC01] sekä Spertus [Spe91] tarjoavat myös monia ratkaisuja naisten Tietojenkäsittelytieteessä kohtaamiin ongelmiin. Verkosta löytyy myös mainio opettajan käsikirja ”Achieving gender equity in science classrooms” [han96], jonka käytännön ohjeet käyvät paljolti yksiin sen kanssa, mitä yleensä pidetään hyvänä opetuksena. On myös syytä huomata, että useimmat havaituista ongelmista koskevat yhtäläillä muitakin vähemmistöihin kuuluvia opiskelijoita. Vaikkei laitoksella tietävästi esiintyisikään mitään naisiin kohdistuvaa syrjintää, kannattaa opettajien ja väitöskirjaohjaajien silti tutustua alan kirjallisuuteen,

Viitteet

- [bar83] Barriers to equality in academia: Women in computer science at mit. Technical report, Laboratory for Computer Science and the Artificial Intelligence Laboratory at M.I.T., 1983.
- [GC01] D. Gürer and T. Camp. Investigating the incredible shrinking pipeline for women in computer science. Technical report, National Science Foundation, 2001.
- [han96] Achieving gender equality in science classrooms: A guide for faculty, 1996. http://www.brown.edu/Administration/Dean_of_the_College/homepginfo/equity/Equity_handbook.html.
- [Kur03] Hannele Kurki. Sukupuoli tutkimusjärjestelmässä. Number 9. Suomen Akatemia, 2003.
- [Lev89] N. Leveson. Women in computer science. Technical report, NSF CISE Cross Directorate Activities Advisory Committee, 1989.
- [pat01] Pathways Project, 2001. <http://www.pathways-project.org.uk/>.
- [PPR⁺90] A. Pearl, M. Pollack, E. Riskin, B. Thomas, E. Wolf, and A. Wu. Becoming a computer scientist: a report by the acm committee on the status of women in computing science. *Communications of the ACM*, (11):47–58, Nov 1990.
- [Spe91] E. Spertus. Why are there so few female computer scientists? Technical Report 1315, MIT Artificial Intelligence Laboratory, 1991.
- [sta] Big sister mentoring program. <http://www.stanford.edu/group/wics/mentoring.shtml>.

Book Orders

Joensuu University Library
P.O.Box 107, FIN-80101 Joensuu, FINLAND, EU
Tel. +358 13 251 2652, fax +358 13 251 2691
e-mail: joepub@joensuu.fi