



Liikkuvan tietojenkäsittelyn tutkimushaasteita*

Kimmo Raatikainen
Helsingin yliopisto
Tietojenkäsittelytieteen laitos
kimmo.raatikainen@cs.helsinki.fi

Tiivistelmä

Se liikkuu sittenkin kerrotaan Galileo Galilein mutisseen inkvisition edessä. Minusta se soveltuu tänään kuvaamaan hyvin tietojenkäsittelyn uudehkoa aluevaltausta, liikkuvaa tietojenkäsittelyä. Kolmen sanan lainauksessa keskeisin osa on *liikkuu*. Lainauksen *se* kaipa erittelyä. *Sittenkin* edustaa jotain lopullista — liikuminen on tunkeutunut tietojenkäsittelyn ytimeen. Toinen lainaus, jonka haluan nostaa esille, on peräisin tieteiskirjailija Gregory Benfordilta: *Any technology distinguishable from magic is insufficiently advanced*.¹ Tämän pitäisi olla mottona jokaiselle tietojenkäsittelytieteilijälle. Mitä ikinä teet, niin maallikosta lopputuloksen pitäisi olla maagista tai oikeammin niin jokapäiväistä, että siihen ei tarvitse kiinnittää huomiota.

Näistä lähtökohdista tarkastelen liikkuvan tietojenkäsittelyn tutkimushaasteita, jotka olen jakanut viiteen osa-alueeseen: *sopeutuvuus, tehokas langaton tiedonsiirto, toimintojen hajauttaminen, dynaamiset yhteysjärjestelmät ja tilannetietoisuus*.

1 Johdanto

Liikkuva tietojenkäsittely nousi käsitteenä esille 90-luvun puolivälissä. Sillä ruvettiin tarkoitamaan informaatiopalveluita, jotta ovat käytettävissä milloin tahansa ja missä tahansa. Liikkuvan tietojenkäsittelyn ja tietoliikenteen katsottiin olennaisesti muuttavan tapoja, joilla informaatiota käsitellään. Liikkuva tietojenkäsittely hyödyntää langatonta tiedonsiirtoa, Internetiä, paikannusjärjestelmiä sekä siir-

rettäviä ja hajautettuja ohjelmistoja. Alan keskeisiä haasteita ovat professori Leonard Kleinrockin [1, 2] mukaan riippumattomuus sijainnista ja laitteista, tietoturva, sopeutuminen uusiin teknologioihin, käyttäjäystävälliset käyttöliittymät ja sovelluksen osittaminen hajautettuun suoritussympäristöön.

Toinen hyvin samankaltainen käsite oli esitelty hieman aikaisemmin. Mark Weiser [9, 10] oli ruvennut puhumaan *näkymättömästä tietojenkäsittelystä*². Ta-

* Artikkelin on alunperin kirjoitettu vuonna 2001 Tietojenkäsittelytieteen seuran toteutumatta jäänyttä 20-vuotisjuhlakirjaa varten.

¹ Arthur C. Clarke esitti asian muodossa “*Any sufficiently advanced technology is indistinguishable from magic.*”

² Joissakin yhteyksissä on viljelty suomenkielistä väännöstä ubiikkietojenkäsittely.

voitteena oli upottaa tietojenkäsittely näkymättömäksi. Weiser käytti rinnastusta, että tietojenkäsittely pitäisi tehdä yhtä tavanomaiseksi kuin sähkö. Aluksi tällä tutkimussaralla mielenkiinto kohdistui pieniin erikoislaitteisiin, tietoliikenneprotokolliin, käyttäjän ja laitteiden vuorovaikutukseen, sekä uudentyylisiin sovellusohjelmiin. Ensimmäisten prototyyppien myötä tutkimuksen mielenkiinnon kohteiksi nousivat myös langaton tietoliikenne, osittaminen ja paikalliset operaatiot, sijainnin varmistaminen ja resurssien löytäminen, yksityisyyden suoja sekä virrankulutus [3].

Tietojenkäsittelytieteen kirjallisuudesta löytyy vielä kolmaskin käsite, jota käytetään kuvaamaan hyvin samankaltaista ongelma-aluetta. *Kaiken kattavan tietojenkäsittelyn* [6] tutkimuskohteena ovat liikkuvien käyttäjien tiedonsaanti, dynaamisesti muuttuvat käyttäjäryhmät, älykkäät tai aktiiviset huoneet, käyttötilanteen huomioonottaminen, ihmisen ja ympäristön vuorovaikutus sekä laitteiden uusien toiminnallisuuden hyödyntäminen.

Tässä kirjoituksessa käsittelen näitä kolmea käsitettä (*nomadic computing*, *ubiquitous computing* ja *pervasive computing*) synonyymeina, sillä niiden taustalla on yhteinen ongelma-alue.

2 Liikkuvuus

Alussa totesin, että Galilei-lainaukseni *se* kaipa erittelyä. Siis kuka liikkuu? Liikkuvassa tietojenkäsittelyssä on eroteltu ainakin neljä erilaista liikkuvuutta: käyttäjän liikkuvuus, laiteliikkuvuus, verkon liikkuvuus ja sovelluksen liikkuvuus.

Käyttäjän liikkuvuus tarkoittaa sitä, että käyttäjä kytkeytyy eri aikoina ja eri paikoissa eri laitteiden avulla samaan tietojenkäsittelyjärjestelmään. Työpaikan työhuoneessa ollessani käytän toista pöytä-

konettani sähköpostin käsittelyyn ja toista Word-dokumenttien käsittelyyn. Kun olen neuvotteluhuoneessa, käytän kannettavaa tietokonettani langattoman lähiverkon kautta sekä sähköpostin että dokumenttien käsittelyyn. Sen sijaan kalenteriani käytän vain toiselta pöytäkoneeltani ja kämmentietokoneeltani. Kotoani käytän näitä sovelluksia joko langattoman kotiverkkoni avulla kannettavalta tietokoneeltani tai kotikoneeltani. Matkoilla ollessani käytän joko kommunikaattoriani tai kannettavaa tietokonettani ja soittolinjaa. Joissakin paikoissa minun on mahdollista kytkeytyä vierailijaksi paikalliseen verkkoon ja sitä kautta päästä käsiksi osaan tietojärjestelmästä. Olennaista tässä on, että minulla on yksi identiteetti, useita eri yhteyslaitteita ja että samanaikaisesti voin käyttää eri sovelluksia eri laitteiden kautta.

Laiteliikkuvuudella tarkoitetaan laitteen kykyä automaattisesti siirtyä tietoliikenneverkon yhteyspisteestä toiseen. Kun käyttäjän liikkuvuudessa oli kyse käyttäjän ja yhteyslaitteen välisestä sidonnasta, niin laiteliikkuvuudessa on kyse yhteyslaitteen ja tietoliikenneverkon välisestä sidonnasta. GSM-matkapuhelinverkko (GPRS-dataverkko) ja langaton lähiverkko ovat esimerkkejä laiteliikkuvuudesta. GSM- tai GPRS-verkko huolehtii puhelun reitityksestä ja tarvittavista rekisteröinneistä, kun puhelin siirtyy tukiasemasta toiseen. Jos luon GSM-puhelimella datapuhelun työpaikkani modeemiin, niin työpaikkani järjestelmän kannalta kysymyksessä ei ole enää laiteliikkuvuus, sillä yhteyspisteeni siihen ei vaihdu, vaikka siirtyisin tukiasemasta toiseen.

Laiteliikkuvuudesta on yleistymässä edellä kuvattua vaativampi muoto. Nykyisin on saatavilla kämmentietokoneita ja älypuhelimia, jotka pystyvät käyttämään sekä GSM- ja GPRS-datapalvelua

että langatonta lähiverkkoa. Tällöin kehittyneen laiteliikkuvuuden tulisi tukea suлавaa siirtymistä yhteysteknologiasta toiseen.

Verkon, tai oikeammin *aliverkon*, *liikkuvuus* on laiteliikkuvuuden eräänlainen yleistys: Kokonainen aliverkko liikkuu siten, että tiedonsiirron reititys ulkomaailmasta kyseisen aliverkon yhdyskäytävään muuttuu. Junan, laivan, lentokoneen tai linja-auton sisäinen lähiverkko on hyvä esimerkki liikkuvasta aliverkosta.

Neljäs liikkuvuuden perusmuoto on *sovelluksen liikkuvuus*. Tällöin puhutaan usein liikkuvasta koodista. Sovelluksen suoritus siirtyy paikasta toiseen. Yksinkertaisimmat liikkuvan koodin muodot ovat jo käytössä. Palvelimelta ladataan ohjelma päätelaitteelle suoritettavaksi. Vaativampi muoto käsittää ohjelman tilan ja koodin siirtämisen uuteen käsittelypaikkaan.

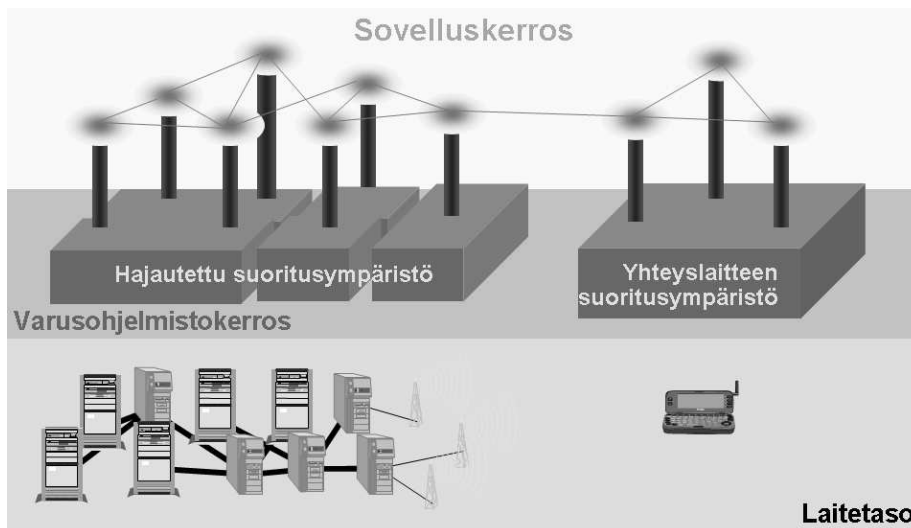
Samanaikaisesta sovelluksen ja käyttäjän liikkuvuudesta käytetään joskus termiä *istunnon liikkuvuus*. Tällöin on kyseessä aina hajautettu sovellus, jossa ainakin käyttöliittymä on yhteyslaitteella ja sovelluksen jotkut osat verkossa olevilla palvelinkoneilla. Usein olen käyttänyt esimerkkinä sähköpostin lukemista. Olen aloittanut sähköpostini lukemisen lähijunassa älypuhelimellani matkalla työpaikalle. Kun saavun työhuoneeseen, haluan jatkaa postini lukemista pöytäkoneellani siinä, mihin matkalla jäin. Tällöin sähköpostin käyttöliittymän on käynnistytävä pöytäkoneellani ja istuntoni tilan on siirryttävä kännykältäni pöytäkoneelleni. Lisäksi sähköpostin käsittelyohjelmiston jotkut osat saatetaan siirtää tehokkuussyistä verkon palvelinkoneilta pöytäkoneelleni. Vastaavasti poistuessani työhuoneestani määrittelemiäni sovellusten käyttöliittymät siirtyvät kännykkääni.

3 Viitekehys

Liikkuvan tietojenkäsittelyn tutkimusongelman jäsentäminen edellyttää viitekehystä, jonka avulla ongelma-alue on mahdollista osittaa hallittaviksi ja riittävän suppeiksi osa-alueiksi. On huomattava, että ei ole olemassa yhtä oikeaa viitekehystä. Valittu viitekehys vaikuttaa sekä tutkimusalueen rajaukseen että sen jaoteluun osa-alueisiin. Tätä taustaa vasten ei ole hämmästyttävää, että kirjallisuudessa esiintyy useita hyvinkin samankaltaisia mutta eri nimisiä tutkimusalueita sekä useita erilaisia tutkimusalueen osituksia. Käyttämäni viitekehysten olen johtanut oman tutkimukseni mielenkiinnon kohteista, joihin on voimakkaasti vaikuttanut tutkimusyhteistyöni suomalaisten telealan yritysten kanssa.

Kuvaan 1 olen luonnostellut viitekehysten, joka kuvaa tulevaisuuden hajautettua sovellusta. Sovelluserroksella sovellus koostuu useista osista, jotka keskustelevat keskenään. Varusohjelmistokerroksella on useita itsenäisiä suoritusympäristöjä. Sovelluksen osat, joita jotkut kutsuvat objekteiksi ja toiset komponenteiksi, on hajautettu suoritettavaksi eri suoritusympäristöihin. Laitetasolla on sovelluspalvelimia, verkkoelementtejä ja yhteyslaitteita tai -järjestelmiä. Yksi varusohjelmistokerroksen suoritusympäristö sitoo yhden tai usean laitteen sovelluksen osan kannalta loogisesti yhtenäiseksi suoritusympäristöksi. Nykyaikainen käyttöjärjestelmä tarjoaa jo yksinkertaisen suoritusympäristön, jossa näyttö, näppäimistö, hiiri, suoritin, keskusmuisti ja joukko oheislaitteita muodostavat tietojenkäsittelyjärjestelmän.

Selkeyden vuoksi kuvassa 1 on yksinkertainen yhden käyttäjän sovellus. Lisäksi yhteyslaitteen suoritusympäristö on oletettu yhdeksi hajauttamattomaksi kokonaisuudeksi. Nämä on tulkittava graafi-



Kuva 1: Sovelluksen viitekehys: Sovelluksen osittaminen ja hajautus

siksi yksinkertaistuksiksi. Jo aivan lähitulevaisuudessa monen käyttäjän sovellukset tulevat olemaan vähintään yhtä tärkeitä kuin yhden käyttäjän Web-surffailun kaltaiset sovellukset. Lisäksi tulevaisuudessa meidän on hallittava yhteyslaitejärjestelmät, jotka perustuvat kehoalueen — käyttäjän mukana kulkeva, esimerkiksi asusteisiin, avaimenperään tai kelloon upotettu, kokoelma ietojenkäsittelylaitteita — verkkoihin tai hajautettuihin kotiverkkoihin.

4 Tutkimushaasteet

4.1 Jako osa-alueisiin

Liikkuvan tietojenkäsittelyn tutkimuksen tavoitteena on tuottaa ratkaisuja, jotka mahdollistavat sovelluksen hajauttamisen heterogeeniseen suoritusympäristöön, jossa käsittelyn ja kommunikoinnin ominaisuudet voivat dynaamisesti muuttua. Pohjimmiltaan on kyse ratkaisuista, jotka mahdollistavat sovelluksen osan siir-

tymisen laitteelta toiselle ja suoritusympäristöstä toiseen. Lisäksi laitteiden välisen kommunikaation ominaisuudet voivat olennaisesti muuttua. Siirtyminen langattomasta lähiverkosta GPRS-verkkoon puottaa kapasiteetin sadasosaan. Käyttöliittymän siirtäminen pöytäkoneelta kännykälle muuttaa näytön ja syötön ominaisuuksia ratkaisevasti.

Olen jakanut liikkuvan tietojenkäsittelyn viiteen osa-alueeseen. Tämä jako ei ole ortogonaalinen, sillä sama tutkimusaihe voi esiintyä useassa osa-alueessa, ei kuitenkaan aivan samassa muodossa. Lisäksi on huomattava, että tutkimusaihe voi esiintyä jossakin muussa muodossa tietojenkäsittelytieteen muilla alueilla. Osa-alueeni ovat *sopeutuvuus* (adaptability), *tehokas langaton tiedonsiirto* (efficient always-on connectivity), *toimintojen hajauttaminen* (distribution of functionality), *dynaamiset yhteysjärjestelmät* (dynamic reconfigurability) ja *tilannetietoisuus* (context-awareness).

Vastoin yleistä suuntausta en ole nos-

tanut tietoturva ja siihen läheisesti liittyviä luottamusta ja yksityisyyden suoja omaksi osa-alueeksi. Tällä en vähättele niiden merkitystä — päinvastoin: tietoturva, luottamus ja yksityisyyden suoja ovat niin merkittäviä kaikessa tietojenkäsittelyssä, että niiden on oltava osa jokaista liikkuvan tietojenkäsittelyn tutkimus-alueita.

Tietoturvan eriyttäminen omaksi tutkimusalueeksi johtaa mielestäni ojasta alikkoon. Alueen tutkijat jätetään puuhailemaan omiin yhteisöihinsä, jolloin tietoturvaratkaisut jäävät irrallisiksi ja siksi usein huomioonottamatta. Tietoturvan, luottamuksen ja yksityisyyden suojan toteuttaminen käyttäjän eri tilanteissa haluamassa laajuudessa on liikkuvan tietojenkäsittelyn ehkä tämän hetken suurin haaste. Lähes yhtä merkittävä ja laaja haaste on virrankulutuksen hallinta.

4.2 Sopeutuvuus

Sopeutuvuuden perusajatus on yksinkertainen. Kun suoritusympäristö muuttuu, niin silloin sovelluksen käyttäytyminen muuttuu vastaamaan käyttäjän toiveita uudessa ympäristössä.

Käyttäjän toiveiden huomioonottaminen ei ehkä ole niin suoraviivaista kuin ensisilmäyksellä saattaa tuntua. Perusongelma on, että riittävän täsmällisten, kattavien ja ristiriidattomien toiveiden kokoelma on hyvinkin laaja. Sähköpostini osalta voisin haluta, että

1. Tärkeät viestit tai niiden lyhennelmät, jos viestin pituus on yli 200 merkkiä, toimitetaan perille aina kustannuksista välittämättä.
2. Ne ääniviestit, joiden kustannus on korkeintaan 0,1 euroa, toimitetaan minulle.
3. Tekstimuotoiset viestit toimitetaan minulle, jos kustannus on korkeintaan viisi senttiä. Mahdolliset liitteet voidaan poistaa mutta viestiä ei saa lyhentää. Tavallisten tekstiviestien vastaanottaminen ei saa häiritä tärkeiden viestien tai ääniviestien vastaanottamista.
4. Vain tärkeiden viestien vastaanottaminen voi viivästyttää ääniviestien vastaanottamista.
5. Vain hyvin tärkeiden viestien vastaanottaminen voi viivästyttää tärkeiden viestien lähettämistä.

Vaikka ylläolevassa esimerkissä on useita toiveita, se on kaukana riittävän täsmällisestä ja kattavasta määrittelystä. Esimerkkini edustaakin sopeutuvuuden yhtä keskeistä ongelmaa: miten käyttäjän toiveista, jotka lähes aina ovat enemmän tai vähemmän epätasällisia ja epätäydellisiä, joskus jopa keskenään ristiriitaisia, saadaan muodostettua säännöstö, joka on riittävän tarkka ja kattava, jotta sen ohjelmallinen käsittely ei aiheuta käyttäjälle odottamattomia yllätyksiä.

Käyttäjän toiveiden ja tavoitteiden oppiminen on sopeutuvuuden kannalta keskeinen tutkimusaihe. Täysin automatisoitu oppiminen tulee vielä pitkään olemaan utopiaa. Oppimisen aloittaminen tyhjästä tietämysjoukosta kestäisi liian kauan. Lisäksi oppimisjakson aikana tapahtuisi liian monia virheellisiä toimenpiteitä, jotka todennäköisesti ärsyttäisivät käyttäjää siinä määrin, että hän heittäisi hilavitkuttimellaan vesilintua.

Sen sijaan opettaminen saattaisi tulla kysymykseen. Samankaltaisesti kuin lapsi opetetaan vuosien saatossa perheen tavoille, sovellus opetetaan muutamassa päivässä käyttäjän tavoille. Opetusjakson aikana käyttöliittymä on yksi kriittisimmistä kohdista: miten käyttäjät haluavat antaa opet-

tamisen tarvitseman palautteen. Todennäköisesti palautejärjestelmänkin on oltava sopeutuva, jotta käyttäjä ei kyllästy palautteen antamiseen. Erityisesti positiivisen palautteen saaminen on usein ongelmallista. Joissakin tilanteissa palautteen puuttuminen voidaan tulkita, että järjestelmä valitsi käyttäjän haluaman toimenpidevaihtoehdon. Toisissa tilanteissa tällainen tulkinta on kyseenalainen tai jopa harhaanjohtava.

Toinen keskeinen ongelma-alue on tietämyskannan eli toiveita kuvaavan säännöstön koko ja päättelyn tarvitseman tietojenkäsittelyn määrä. Luultavimmin tietämys on ositettava siten, että osamalleista tulee riittävän pieniä. Pelkkä osittaminen ei kuitenkaan ole riittävää, vaan osamalleja on pystyttävä yhdistelemään eri tavoilla eri tarkoituksia varten.

Sopeutuminen ei voi olla ainoastaan reaktiivista, reagointia tapahtuneisiin muutoksiin. Kun patteri loppuu tai tietoliikenneyhteys katkeaa, niin useiden toimenpiteiden suoritus on mahdotonta. Monissa tilanteissa olisi kuitenkin jotain hyödyllistä voitu tehdä ennakolta. Ennaltakäsin sopeutuminen vaatii lähitulevaisuuden ennustamista. Ennustamisen keskeinen tutkimusongelma on ennusteen uskottavuus. Toisin sanoen järjestelmän tulisi pystyä erottamaan ennakoimattomat tilanteet ennustettavissa olevista tilanteista.

Jos tavoitteeksi otetaan täydellinen järjestelmä, niin ennustettavuus voidaan unohtaa. Tässä yhteydessä on syytä muistaa sanonta parhaasta hyvän pahimpana vihollisena. Realistisena tavoitteena on oltava järjestelmä, joka usein (tai ainakin joskus) toimii käyttäjän toiveiden mukaisesti muttei juuri koskaan käyttäjän toiveiden vastaisesti.

4.3 Langaton tietoliikenne

Kommunikaatioväylänä ilmatie on ongelmallinen. Viimeisimpien kymmenen vuoden aikana koodaustekniikan kehitys on lisännyt valtavasti kiinteiden yhteyksien kapasiteettia. Valitettavasti näitä kaikkia tuloksia ei voida hyödyntää ilmatielle, sillä koodaus on aina kompromissi informaatiotiheyden ja toisteisuuden välillä. Toisteisuutta tarvitaan suojautumisessa fyysisen tason häiriötä vastaan. Langattoman tiedonsiirron perusongelma on siirtotien epävakaas, häiriöt vaihtelevat sekä ajan, paikan että ympäristötekijöiden suhteen.

Usein kuulee sanottavan, että kolmannen sukupolven matkaviestinjärjestelmien tarjoama kaksi megabittia sekunnissa olisi riittävän nopea. Tällöin kuitenkin unohdetaan, että tuo 2 Mbps on yhteiskäyttöinen kapasiteetti. Lisäksi tietojenkäsittelyn historia ei tunne tapausta, jossa vapaa kapasiteetti olisi jäänyt pitkäksi aikaa käyttämättä. Kun vielä otetaan huomioon, että kiinteän verkon kapasiteetti lisääntyy huomattavasti nopeammin kuin langattomien verkkojen kapasiteetti, niin langattoman tiedonsiirron tehokkuus pysyy keskeisenä tutkimusaiheena.

Toisaalta on huomattava, että langattomat verkot eivät ole ominaisuuksiltaan samankaltaisia. Langattomissa lähiverkoissa, satelliittiyhteyksillä, soluverkoilla ja lyhyen kantaman radioverkoilla on kullakin omat erityisongelmansa. Tämän vuoksi liikkuva tietojenkäsittely tarvitsee ratkaisuja, jotka tukevat erilaisia langattomia linkkejä. Linkkitason loistavat ratkaisut eivät kuitenkaan ole riittäviä. Huonosti langatonta linkkiä hyödyntävä kuljetusprotokolla, tehoton esitystapa, sopimaton keskustelu- tai vuorovaikutustapa — jokainen näistä voi romahduttaa tiedonsiirron tehokkuuden. Lisäksi on huolehdittava, että rinnakkaiset tiedonsiirrot

jakavat käytettävissä olevan kapasiteetin tarkoituksenmukaisesti. Siksi tiedonsiirtoa langattomissa ympäristöissä on tutkittava kaikilla protokollatasoilla.

4.4 Toimintojen hajauttaminen

Tietojenkäsittely liikkuvalla laitteella käsin on haasteellista. Tulevaisuudessa liikkuva käyttäjä ei tule tyytymään yhden ainoan yhteyslaitteen käyttöön, ei siinäkään tapauksessa, että laite pystyisi käyttämään useita eri tiedonsiirtoteknologioita. Sen vuoksi istunnon siirtäminen, tai oikeammin istunnon yhden päätepisteen, siirtäminen laitteelta toiselle on oltava mahdollista.

Näissä tilanteissa sovelluksen osittaminen ja osien sijoittelu on mielenkiintoinen tutkimusaihe. Liikkuva tietojenkäsittely tarvitsee menetelmät, joilla sovelluksen osat sijoitellaan tarkoituksenmukaisella tavalla hajautettuun suoritusympäristöön. Lisäksi ympäristön olennaiset ja riittävän pitkäaikaiset muutokset voivat luoda tilanteen, missä osien uudelleensijoittelu on kannattavaa. Teknisenä toimenpiteenä uudelleensijoittelu on melko suoraviivaista, vaikkakaan ei täysin triviaalia. Sen sijaan päätteleysäännöt, joilla todetaan olosuhteiden olennaiset ja riittävän pitkäaikaiset muutokset, on erittäin haasteellinen tutkimusaihe.

Toinen hyvin keskeinen tutkimusaihe on vikasietoisuus. Perinteisesti monentamista (saman tiedon tai palvelun sijoittamista eri puolille verkkoa) on käytetty vikasietoisuuden ratkaisuna hajautetuissa järjestelmissä. Liikkuvassa tietojenkäsittelyssä tämä ei ole mahdollista, sillä kaikkea ei voida sijoittaa yhteyslaitteelle. Lisävaatimuksena on, että perussovellukset ovat ainakin jossakin määrin käyttökelpoisia, vaikka sovelluksen kaikki osat eivät olisikaan käytettävissä. Tämän vaatimuksen täyttäminen tulee olennaisesti

vaikuttamaan sovellusten toimintojen toteutukseen.

4.5 Dynaamiset yhteysjärjestelmät

Nykyiset yhteyslaitteet ovat pääasiassa integroituja kokonaisuuksia kuten kämmen-tietokone, sylitietokone ja matkapuhelin. Tilanne tulee kuitenkin muuttumaan melko pian. Seuraavan sukupolven matkapuhelimen seuraajat tulevat toteuttamaan hyvin erilaisia toimintoja kuin niiden nykyiset isovanhemmat.

Nämä tulevaisuuden viestintälaitteet tulevat olemaan henkilökohtaisen tietojenkäsittely- ja viestintäjärjestelmän ydin. Niiden keskeinen tehtävä tulee olemaan tarkoituksenmukaisen yhteysjärjestelmän muodostaminen ja ylläpitäminen. Ne nuuhkivat ympäristöään etsien sopivia oheislaitteita (näyttöjä, syöttimiä, prosessoreita, tukimuisteja) ja tietoliikennetyksiiä. Tulevaisuuden viestintälaitte muodostaa tarkoituksenmukaisimman järjestelmän, joka voidaan automaattisesti konfiguroida uudelleen. Teknisellä tasolla tällaisen toiminnallisuuden idut ovat jo olemassa. Sen sijaan päätteleysäännöt, joiden avulla tarkoituksenmukaisin järjestelmä valitaan, on lähes koskematon tutkimusaihe.

Lisäksi on huomattava, että tulevaisuuden viestintälaitteen on oltava — ainakin jossakin määrin — toimintakykyinen, vaikka ympäristöstä ei löytyisikään sopivia oheislaitteita. Toinen nykyisyydestä merkittävästi eroava vaatimus liittyy dynaamisiin virtuaaliverkkoihin. Tulevaisuuden viestintälaitteen on hallittava useita samanaikaisia virtuaaliverkkoja.

Dynaamisten yhteysjärjestelmien alueella on lukuisia haasteellisia tutkimusaiheita. Käsitteellisellä tasolla tutkimusaiheita ovat muun muassa profilit (ominai-

suuksien esittämismuodot), erilaiset kontekstit (olosuhteiden esittämismuodot), roolit (virtuaaliset identiteetit) ja luottamus. Teknisellä tasolla tutkimusaiheita ovat muun muassa todentaminen, tunnistaminen ja valtuuttaminen sekä löytäminen. Toimintojen hajauttamisen kanssa yhteinen merkittävä tutkimusaihe on päätely.

Tunnistamisessa ja todentamisessa hajautetun järjestelmän osat varmistavat muiden osapuolten identiteetin ja ominaisuudet. Onko Kalleksi itseään väittävä todella Kalle? Mitä hän osaa? Tekeekö hän sen minkä lupaa tehdä? Keskeinen teema on luottamus. Miten henkilökohtaisen tietojenkäsittely- ja viestintäjärjestelmän ydin muodostaa ja ylläpitää luottamusketjun, jonka avulla se toisaalta vakuuttuu omasta eheydestään ja toisaalta muiden osapuolten luotettavuudesta?

Valtuuttamisessa järjestelmä antaa jonkin tietojenkäsittelytehtävän suoritettavaksi jossakin muussa suoritussympäristössä. Tällöinkin luottamus on keskeisessä asemassa. Toimeksiantajan on oltava riittävässä määrin vakuuttunut, että tehtävä tulee suoritettua toimeksiannon mukaisesti. Toisaalta tehtävän suorittaja on äärimmäisen kiinnostunut tehtävän suorituksen mahdollisista sivuvaikutuksista. Miten toimeksisaaja varmistaa, että annetun tehtävän suorittaminen ei vaikuta suorittajan toimintaympäristön eheyteen?

Löytämisessä on kyse järjestelmän toimintaympäristön muutoksesta. Uusi hyödyllinen oheislaitte tai palvelu tulee käytettäväksi tai vanha katoaa. Järjestelmä voi aktiivisesti etsiä uusia komponentteja tai passiivisesti seurata ympäristön "palvelutarjontaa". Soveltuvia löytämismenetelmiä on melko runsaasti. Näistä pitäisi valita kuhunkin tilanteeseen tarkoituksenmukaisimmat. Löytäminen liittyy olennaisesti myös tilannetietoisuu-

teen, kuten myös profiilit, kontekstit ja roolit. Tässä kolmikossa keskeistä on erilaiset semanttiset kuvaukset: Mitä käyttäjä todella haluaa? Mitä eri oheislaitteet osaavat tehdä, että kustannukset? Millaisia suoritussympäristöjä erilaiset palvelut tarvitsevat?

4.6 Tilannetietoisuus

Sijaintia on alusta alkaen pidetty liikuvan tietojenkäsittelyn keskeisenä osa-alueena [2]. Sijaintiperustaiset sovellukset ovat jo jossakin määrin käytössä. Teknisesti on mahdollista lähettää mainoksia ja viestejä lähistöllä oleviin yhteyslaitteisiin. Näihin menetelmiin liittyen tutkimusaiheina tulevat kysymykseen muun muassa yksityisyys sekä säädösten vastaisen käytäytymisen estäminen. Tutkimushaasteena on myös riittävän joustavat ja monipuoliset sääntelymekanismit.

Toinen merkittävä haaste — ei ehkä niinkään tutkimusaihe mutta ainakin toteuttamishaaste — on eri sijaintitietoa koskevien standardien huomioonottaminen. Merkittävä tutkimusaihe sen sijaan on käyttäjän tulevan sijainnin ennustaminen. Tällaista ennustetta voidaan käyttää esimerkiksi tapaamisten automaattiseen sopimiseen ja viivästettyyn tiedon siirtoon. Sijainnin ennustamisessa merkittävässä asemassa on eri tietolähteiden yhdistäminen.

Sijaintitieto on yleisimmin (ja toistaiseksi lähes ainoa) käytetty tilannetieto. Kuitenkin mikä tahansa tieto, joka liittyy sen hetkiseen käyttötilanteeseen, voi olla tilannetietoa. Tilannetietoisuus tarkoittaa, että järjestelmä tai sovellus käyttää hyväkseen tilannetietoa: sijaintia, etene- misuuntaa tai -nopeutta, kellon- tai vuodenaikaa, säätilaa, käytettävissä olevia resursseja, lähistöllä olevien ihmisten tietoa, käyttäjän sen hetkistä tai tulevaa aktiiviteettia ja niin edelleen.

Järjestelmä on tilannetietoinen, jos se voi kerätä, tulkita ja käyttää tilannetietoa sekä muuttaa omaa toimintaansa vastaamaan käyttäjän toiveita kyseisessä käyttötilanteessa. Tilannetietoisuuden ehkä keskeisin haaste on käyttötilanteen joustava mallintaminen. Tavoitteena on tehokkaat tavat ja menetelmät esittää, ylläpitää, jakaa, suojata, päätellä ja kysellä tilannetietoa. On syytä huomata, että tilannetiedon joukko vaihtelee sovelluksesta toiseen. Siksi hajautettu tiedonhallinta on keskeinen osa tilannetietoisia sovelluksia.

Lopuksi vielä pieni esimerkki. Läheisyys (*proximity*) sisältää myös aikakomponentin. Haluaisin antaa henkilökohtaiselle apurilleni tehtävän: *vara minulle ja vaimolleni pöytä tänään klo 19 jostakin mukavasta lähellä olevasta ravintolasta*. Kun apurini aloittaa toimeksiantoni käsittelyn, sen on ratkaistava muun muassa seuraavat kysymykset:

- Missä ja milloin aion tavata vaimoni?
- Mitä me oikein tarkoitamme lähellä olevalla ravintolalla? Onko 500 metrin päässä oleva kohtuullisen mukava ravintola parempi kuin kahden kilometrin päässä oleva erittäin mukava ravintola?
- Kuinka samankaltaista minun ja vaimoni arvottaminen on? Miten mahdolliset erot arvottamisessa otetaan huomioon?
- Mitä mukava ravintola oikein tarkoittaa?

Yllä oleva esimerkki kuvaa joitakin niistä vaikeuksista, joita useamman kuin yhden ominaisuuden perusteella tapahtuva valinta luo. Siksi profiilit ja arvottamissäännöt ovat keskeisiä tutkimusaiheita. Pohjimmiltaan arvottamisessa tar-

vitaan semanttiset säännöt, joilla painotetaan eri ominaisuuksia ja eri ominaisuuksien arvoja.

5 Johtopäätelmät

Yli viisitoista vuotta sitten Mark Weiser [9] asetti näkymättömän tietojenkäsittelyn, tietokoneiden ja niiden välisen kommunikaation häivyttämisen käyttäjien ulottumattomiin, tietojenkäsittelytieteen tavoitteeksi. Weiserin alkuperäistä tutkimusohjelmaa [10] ovat myöhemmin täydentäneet muun muassa Alan Demers [3] ja Leonard Kleinrock [1, 2]. Wireless World Research Forum on julkaisut kaksi liikkuvan tietojenkäsittelyn ja langattoman tietoliikenteen tulevaisuutta käsittelevää kirjaa [7, 8]. Itse olen käsitellyt liikkuvan tietojenkäsittelyn tutkimushaasteita artikkelissa [4] ja käyttöjärjestelmien tutkimushaasteita artikkelissa [5].

Monet tutkimusohjelmissani käyttämäni termit ja käsitteet ovat lähtöisin Xerox PARC tutkimuslaitoksen [3, 9, 10] uraa uurtavasta työstä. Tämä ei kuitenkaan merkitse, että liikkuva tietojenkäsittely ei olisi edennyt viime vuosina. Kaikialle maailmaan laajentunut tutkimustoiminta on syventänyt tutkimusalueen ymmärtämistä ja erilaisten ratkaisujen hyviä ja huonoja puolia. Lukuisia tutkimusaiheita on kuitenkin vielä avoinna eli väitöskirjojen tekijöitä kaivataan.

Viitteet

- [1] R. Bagrodia, W. W. Chu ja L. Kleinrock: Vision, Issues, and Architecture for Nomadic Computing. *IEEE Personal Communications*, joulukuu 1995, ss. 14–27.
- [2] Cross-Industry Working Team: *Nomadcity in the NII*. <http://>

- www.lk.cs.ucla.edu/LK/lkxiwt/
[1.6.2007]
- [3] A. J. Demers: Research Issues in Ubiquitous Computing. *Proceedings of the 13th Annual ACM Symposium on Principles of Distributed Computing (PODC'94)*, elokuu 1994, ss. 2–8.
- [4] K. Raatikainen: A New Look at Mobile Computing. *Proceedings of the IEEE International Workshop on Convergent Technologies 2005*, kesäkuu 2005.
- [5] K. Raatikainen: Operating System Issues in Future End-User Systems. *Proceedings of the IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC 2005)*, syyskuu 2005.
- [6] M. Satyanarayanan: Pervasive Computing: Vision and Challenges. *IEEE Personal Communications*, elokuu 2001, ss. 10–17.
- [7] R. Tafazolli (toim.): *Technologies for the Wireless Future*. Wiley, 2005.
- [8] R. Tafazolli (toim.): *Technologies for the Wireless Future, Volume 2*. Wiley, 2006.
- [9] M. Weiser: The Computer for the Twenty-First Century. *Scientific American*, syyskuu 1991, ss. 94–104.
- [10] M. Weiser: Some Computer Science Issues in Ubiquitous Computing. *Communications of the ACM*, heinäkuu 1993, ss. 74–84.