

Opiskelijoiden hyödyntäminen tietojenkäsittelytieteen opettamisessa

Hanna Arpiainen

Kandidaatin tutkielma
HELSINGIN YLIOPISTO
Tietojenkäsittelytieteen laitos

Helsinki, 30. huhtikuuta 2013

Tiedekunta — Fakultet — Faculty		Laitos — Institution — Department	
Matemaattis-luonnontieteellinen		Tietojenkäsittelytieteen laitos	
Tekijä — Författare — Author			
Hanna Arpiainen			
Työn nimi — Arbetets titel — Title			
Opiskelijoiden hyödyntäminen tietojenkäsittelytieteen opettamisessa			
Oppiaine — Läroämne — Subject			
Tietojenkäsittelytiede			
Työn laji — Arbetets art — Level	Aika — Datum — Month and year	Sivumäärä — Sidoantal — Number of pages	
Kandidaatin tutkielma	30. huhtikuuta 2013	16	
Tiivistelmä — Referat — Abstract			
<div>Kandidaatin tutkielma</div> <div>ACM Computing Classification System (CCS):</div>			
Avainsanat — Nyckelord — Keywords			
avainsana 1, avainsana 2, avainsana 3			
Säilytyspaikka — Förvaringsställe — Where deposited			
Muita tietoja — Övriga uppgifter — Additional information			

Sisältö

1	Johdanto	1
2	Muutokset tietojenkäsittelytieteen opetuksessa	1
2.1	Opiskelijamäärien kasvu	1
2.2	Muutokset sisällössä	2
2.3	Pula opettajista	2
2.4	Suurten kurssien ongelmat	2
3	Oppilaat opettajina	4
3.1	Ohjaajien työtehtäviä	4
3.2	Millainen on hyvä ohjaaja?	5
3.3	Ohjaajien koulutus	7
3.4	Käytännön organisaatio	8
4	Seurauksia	9
4.1	Hyötyjä	9
4.2	Haasteita	11
5	Yhteenveto	12
6	Lähteet	13

1 Johdanto

Tietojenkäsittelytieteen opetuksen tarve on kasvanut [29]. Usein taloudellisten resurssien puute kuitenkin johtaa opiskelupaikkojen lukumäärän rajoittamiseen. Lisäksi etenkin 1980-luvulla osaavista opettajista oli pulaa, joten henkilökuntaan jouduttiin palkkaamaan riittämättömästi koulutettuja ulkopuolisia. Teollisuuden menestys houkuttelee opiskelijoita siirtymään nopeasti työelämään opintojen jatkamisen sijaan, jolloin on yhä vaikeampi saada uusia opettajia uusille tietojenkäsittelytieteen opiskelijoille [27].

Resurssi- ja opettajapulasta seuraavaa ongelmaa voi helpottaa kehittämällä yhteistyötä teollisuuden kanssa, koska sillä on taloudellisia resursseja tukea opetusta [29]. Toinen ratkaisu on palkata alkeiskursseille mahdollisimman pätevää henkilökuntaa.

Kolmas ratkaisu ongelmaan on hyödyntää opiskelijoita opetuksessa. Monet yliopistot ovat päättäneet hyödyntää opiskelijoita kurssien opetukseen, sillä he ovat halvempia kuin pitkälle koulutettu työvoima. Taloudellisen edun lisäksi opiskelijoiden on havaittu olevan hyödyllisiä monella muullakin tavalla, sillä opiskelijoiden käytön ohjaajina on havaittu muun muassa nostavan opetuksen laatua ja parantavan kurssin ilmapiiriä.

Tässä artikkelissa käsittelemme opiskelijoiden hyödyntämistä tietojenkäsittelytieteen opetuksessa. Esittelemme, millä tavoin opiskelijoita voi käyttää alkeiskurssien ohjaajina, mitä työtehtäviä heillä on ja millaista organisaatiota ohjaajien käyttäminen vaatii. Käsittelemme myös, mitä hyötyjä ja haasteita ohjaajien käyttämisestä on.

2 Muutokset tietojenkäsittelytieteen opetuksessa

Opiskelijamäärien vaihtelu, opetustyylien uudistuminen ja pula opettajista ovat aiheuttaneet muutoksia tietojenkäsittelytieteen opetukseen. Kasvaneista opiskelijamääristä seuraaviin suuriin kursseihin voi liittyä ongelmia, jotka heikentävät opetuksen laatua.

Tietojenkäsittelytieteen opetuksen arvostus on kasvanut, eikä sitä enää pidetä toisarvoisena tutkimukselle [5]. Monet yliopistot ovat todeneet laadukkaan opetuksen olevan keskeinen osa laitoksen toimintaa ja pyrkineet parantamaan opetusympäristöään.

2.1 Opiskelijamäärien kasvu

Tietojenkäsittelytieteen opiskelijoiden lukumäärän vaihtelu on syklistä [29]. Esimerkiksi kotitietokoneiden ja World Wide Webin yleistymisen kasvattivat alan suosiota lisäten opiskelijoiden määrää, kun taas IT-kuplan puhkeaminen 2000-luvun alussa vähensi opiskelijamäärää. Esimerkiksi Stanfordin yliopistossa tietojenkäsittelytieteen opiskelijoiden määrä nousi 3907:stä 11140:een vuosina 1974-1985 [25].

Ajoittaisista taantumisista huolimatta opiskelijoiden määrä on pitkällä ajanjaksolla kasvanut, ja teollisuus tarvitsee yhä enemmän osaaajia [29]. IT-kuplan puhkeamisen aiheuttamasta opiskelijakadosta seurasi pula tietojenkäsittelytieteen koulutetuista ammattilaisista. Yhdysvaltojen yliopistojen pitäisi pystyä kaksin- tai kolminkertaistamaan tietojenkäsittelytieteen valmistuneiden opiskelijoiden määrää, jotta alan uusiin työpaikkoihin ja eläköityvien ammattilaisten korvaamiseen riittäisi osaaajia.

Muiden aineiden opiskelijat haluavat opiskella tietojenkäsittelytieteen perusteita, koska he uskovat, että ohjelmointitaidot auttavat heitä työmarkkinoilla [29]. Tämä lisää tietojenkäsittelytieteen opetuksen tarvetta. Etenkin alkeiskursseille pitäisi pystyä järjestämään riittävästi opetustarjontaa.

2.2 Muutokset sisällössä

Ketterän ohjelmistokehityksen yleistymisen takia tarvitaan yhä enenevässä määrin ohjelmistoalan ammattilaisia, jotka osaavat toimia ohjaajina ohjelmoijille ja ohjelmointitiimeille [36]. Tämän takia oppilaitosten on annettava tietojenkäsittelytieteen opiskelijoille mahdollisuus opetella ohjelmoinnin lisäksi ohjaajana toimimiseen liittyviä taitoja.

Nykymaailmassa tarvitaan ongelmanratkaisukykyä, jota alkeiskurssien on pystyttävä opettamaan [37]. Opetuksen tyyli on muuttunut opetuskeskeisestä ajattelutavasta oppimiskeskeiseen. Opiskelijalle ei enää vain kerrota tietoa, vaan he rakentavat itse osaamisensa vertaamalla uutta tietoa aiemmin opittuun. Oppimiskeskeisessä ajattelutavassa vastuu oppimisesta on opiskelijalla itsellään.

Teknologian kehitys on muuttanut oppimisympäristöä, ja opetusta voi tarjota virtuaalimaailmassa [37].

2.3 Pula opettajista

Tietojenkäsittelytieteen alaa on vaivannut puute pätevistä opettajista [27]. Teollisuuden menestys houkuttelee opiskelijat jättämään opintonsa ja siirtymään töihin, jolloin uusien opettajien kouluttaminen on yhä vaikeampaa. Työvoimapula aiheuttaa ylityötä ja stressiä opettajille, mikä lisää heidän ovat halukkuuttaan siirtyä muihin töihin. Opettajien puute vaikeuttaa kasvaviin opiskelijamääriin vastaamista.

Yliopiston täyspäiväiset opettajat voivat olla kiinnostuneempia opettamaan syventäviä kursseja tai osallistumaan tutkimukseen alkeiskurssien opettamisen sijaan [32].

2.4 Suurten kurssien ongelmat

Kurssikoon kasvaessa kasvaa myös kurssin tehtävien korjaukseen ja opiskelijoiden henkilökohtaiseen ohjaukseen tarvittava työmäärä [14]. Suurissa

opiskelijamäärissä parhaiden ja heikoimpien opiskelijoiden ero voi olla hyvin suuri, joten ääripäät tarvitsisivat erityisen paljon yksilöllistä huomiota. Monilla opiskelijoilla ei ole tietojenkäsittelytieteestä aiempaa kokemusta, ja suuret luennot ja yksin tehtävät harjoitukset voivat tuottaa heille hankaluuksia [19]. Suurilla luennoilla opiskelijoilla voi olla vaikeuksia keskittyä ja seurata opetusta [6]. Uusien opiskelijoiden voi olla hankala lähestyä kokenutta luennoijaa [15].

Se, että opiskelija pääsee opiskeluun mukaan heti opintojen alussa, ehkäisee syrjäytymistä [31]. Aktiivinen osallistuminen ehkäisee opintojen kesken jättämistä [6], mutta pelkkä luentojen passiivinen kuuntelu ei rohkaise opiskelijoita osallistumaan [15]. Yliopisto-opintojen itsenäisyys on uutta monille opiskelijoille, ja heidän on helppo oppia huonoja opiskelutapoja ja olla välittämättä läsnäolon tärkeydestä kursseilla [15]. Luentoihin painottuvilla kursseilla voi olla vaikeuksia tarjota opiskelijoille roolimalleja, jotka näyttäisivät esimerkillään, kuinka opinnoissa pysyy mukana.

Suurten ryhmäkokojen seurauksena opiskelijat uhkaavat hukkua massaan ja voivat sen seurauksena menettää mielenkiintonsa alaa kohtaan [14]. Suurten luentokurssien opiskelijoilla voi olla heikompi itseluottamus omien kykyjensä, opintomenestyksensä tai tietojenkäsittelytieteen hyödyllisyyden tulevaisuudessa suhteen kuin pienryhmissä opiskelevilla [6].

Kokeiden ja tehtävien tekeminen vaikeutuu oppilasmäärän kasvaessa, jos koetehtävistä halutaan saada mahdollisimman nopeasti tarkistettavia [14]. Tehtävät tulisi olla määritelty mahdollisimman selkeästi, sillä epäselvien kysymysten aiheuttamia väärinymmärryksiä tulee suurten opiskelijamäärien kanssa enemmän.

Kurssikoon kasvaessa sen hallinnointi vaikeutuu, mukaan lukien kurssinateriaalin valmistelu ja jakaminen opiskelijoille, neuvonnan tarjoaminen ja henkilökunnan hallinnointi [7]. Suurella kurssilla voi olla haasteellista taata, että kaikki opiskelijat arvostellaan tasavertaisesti.

Suurten luentojen passiivisella kuuntelemisella ja asioiden ulkoa opettelulla opiskelija voi yltää pintapuoliseen oppimiseen, mutta syvempään oppimiseen vaaditaan aktiivista opiskelua [6]. Pintapuolisella oppimisella tarkoitetaan opiskelutapaa, jossa opiskelija keskittyy yksityiskohtiin kokonaisuuden sijasta suoriutuakseen yksittäisistä tehtävistä [1]. Opiskelu voi olla ylimalkaista eikä välttämättä johda opeteltavan aiheen ymmärtämiseen. Syvässä oppimisessa opiskelija yhdistelee itse aktiivisesti aiemmin oppimaansa ja pyrkii ymmärtämään käsiteltävän asian. Syvässä oppimisessa opiskelija pyrkii kokonaisvaltaiseen ymmärtämiseen, ja se auttaa kehittämään kriittistä ajattelua ja käsitteiden pitkäaikaista muistamista.

Passiivinen luentojen kuuntelu ei vastaa tietojenkäsittelytieteen alan työelämää, jossa tarvitaan ongelmanratkaisutaitoja ja kykyä toimia ryhmässä [35].

3 Oppilaat opettajina

Kasvaviin opiskelijamääriin, suuriin luentokursseihin ja työvoimapulaan liittyviä ongelmia voi ratkaista käyttämällä opiskelijoita tietojenkäsittelytieteen opetuksessa. Opiskelijat voivat toimia ohjaajana ryhmälle alkeiskurssin opiskelijoita [25]. Ryhmässä oppimisen on havaittu auttavan opiskelijoita oppimaan nopeammin ja tehokkaammin [37].

Ohjaajat tarjoavat lisäopetusta kurssin tueksi ja auttavat kurssin opiskelijoita esimerkiksi tehtävien kanssa. Ohjaajilla ei tarvitse olla omaa vastuuryhmäänsä, vaan he voivat antaa yleistä neuvontaa tietokonealuokissa [36, 35]. Joissakin laitoksissa opiskelijat käyttävät enemmän aikaa ohjaajien kuin luennoijan kanssa [20].

Ryhmänohjaajien lisäksi opiskelijoita voi käyttää vertaistukena [34]. Vertaistukea voi saada opiskelija, joka tarvitsee lisää tukea opintoihinsa, tai joka tarvitsee ylimääräisiä haasteita. Tukihenkilön tulee osata opettamansa kurssin sisältö hyvin. Tukihenkilö tapaa neuvottavaansa ja avustaa häntä tehtävien kanssa. Neuvonta voi mennä myös ristiin; opiskelija voi olla tukihenkilönä toiselle opiskelijalle yhdellä kurssilla ja saman henkilön neuvottavana jollakin toisella.

Opiskelijoiden käyttäminen ohjaajina vaatii laitokselta sitoutumista [15]. Koska ohjaajien käyttämiseen tarvitaan ennakkoinvestiointeja, kannattaa laitoksen järjestää ohjaajia vain kursseille, joita siellä on aiemmin opetettu. Esimerkiksi Helsingin yliopistossa jopa 20 prosenttia tietojenkäsittelytieteen opiskelijoista toimii ohjaajana alkeiskursseilla [35]. Joissakin laitoksissa, esimerkiksi Kalifornian yliopistossa, ohjaajana toimiminen voi olla pakollinen osa tutkintoa [13].

3.1 Ohjaajien työtehtäviä

Ryhmänohjaajien tehtäviin kuuluu esimerkiksi viikottaisten keskustelusessioiden pitäminen [25]. Keskustelusessiossa käydään läpi kurssin sisältöä, mutta sen ei tarvitse noudattaa mitään tiukkaa kaavaa, vaan ryhmänohjaaja voi muokata sen vastaamaan ryhmänsä tarpeita. Ryhmänohjaaja voi esimerkiksi kerrata luentojen asioita, esittää lisäesimerkkejä, tai vastata opiskelijoiden esittämiin kysymyksiin. Viikkotapaamisten ei tarvitse olla vain ohjaajan esiintymistä, vaan ne voivat sisältää opiskelijoiden ryhmäkeskusteluja [8]. Tarvittaessa ryhmänohjaajan on oltava valmis hylkäämään aiemmat suunnitelmansa, mikäli selviää, että ryhmä on ymmärtänyt jonkin tärkeän asian väärin ja tilanne on korjattava nopeasti [25].

Ryhmänohjaajan tehtäviin voi kuulua viikottainen yksilöllinen palautekeskustelu jokaisen ryhmänsä opiskelijan kanssa. Opiskelija voi selittää ratkaisunsa johonkin tehtävään, tai kertoa, että ei ole ymmärtänyt jotakin kurssin asiaa. Ohjaaja taas voi selittää, mistä jokin virhe johtuu, ja miten sen voi jatkossa välttää. Näissä keskusteluissa ryhmänohjaaja saa kuvan opiskelijan edistymi-

sestä kurssilla [25, 24].

Ohjaajalla voi olla vastaanottoaika, jonka aikana opiskelijat voivat käydä kysymässä häneltä yksilöllisiä vastauksia kurssiin liittyviin tehtäviin [8]. Ohjelmointikursseilla ohjaajat voivat päivystää tietokonealuokassa valmiina auttamaan kurssin opiskelijoita kurssiin liittyvissä ongelmissa. Ruuhka-ajoille päivystäviä ohjaajia pyritään järjestämään enemmän [25, 24], ja ohjaajat saattavat voida kutsua lisäapua, mikäli tietokonealuokka ruuhkautuu pahasti [16]. Ohjaajan ei tulisi antaa suoraa vastausta tehtävään, vaan ohjata ja neuvoa opetettavaa löytämään ratkaisu itse [36, 16]. Ohjelmointitehtävien yhteydessä ohjaaja voi kannustaa opetettavaa kirjoittamaan hyvää ja luettavaa koodia esimerkiksi muuttujien selkeällä nimeämisellä. Ohjaajat voivat erottuakseen joukosta täydessä tietokonealuokassa käyttää esimerkiksi turvaliivejä [35].

Ohjaajat voivat pitää ennen koetta kertaustilaisuuden kurssin sisällöstä. Kertaustilaisuudessa he voivat kertoa, mitkä kurssin asiat ovat tärkeitä ja mitä kokeessa todennäköisesti kysytään, tai esitellä vanhoja kokeita ja niiden ratkaisuja [8].

Ohjaajan tehtäviin kuuluu usein tehtävien tarkistaminen [9]. Kun ohjaajat osallistuvat tehtävien tarkastamiseen, voi tehtäviä olla enemmän, ja luennoija voi käyttää aikansa tehokkaammin esimerkiksi kirjoittamalla kommentteja tehtäviin.

Ohjaajan voidaan edellyttää osallistuvan kurssin luennoille pysyäkseen mahdollisimman hyvin perillä kurssin etenemisestä ja käsitellyistä esimerkeistä [24, 8]. Ohjaajat voivat ratkaista kurssin tehtävät ennen kuin ne annetaan opiskelijoille, mikä auttaa heitä valmistautumaan opiskelijoiden neuvontaan [35]. Ohjaajat voivat käydä kurssin materiaalin läpi ennen kuin luennoija esittää sen opiskelijoille, ja huomauttaa mahdollisista virheistä tai puutteista [36].

Ohjaaja voi päivittää kurssin verkkosivuja [9]. Tämä vähentää luennoijan työtaakkaa ja auttaa opiskelijoita, joilla on vaikeuksia omien muistiinpanojen tekemisen kanssa. Luennoille osallistuva ohjaaja voi päivittää esimerkkejä verkkosivuille reaaliajassa.

Ohjaajat voivat valvoa kurssikokeita [26].

3.2 Millainen on hyvä ohjaaja?

Ohjaajalta vaaditaan innostusta auttaa muita oppimaan ja vastuullisuutta noudattaa kurssin käytäntöjä. Ohjaajan ei ole välttämätöntä olla esimerkiksi ohjelmoinnin erityisosaaja, mutta hänen edellytetään ymmärtävän kurssin sisältö ja olevan valmis kehittämään itseään pystyäkseen auttamaan opetettaviaan riittävästi [25]. Ohjaaja voi hyötyä siitä, että hänellä on itsellään ollut oppimisvaikeuksia, sillä se voi auttaa häntä ymmärtämään paremmin opetettaviensa vaikeuksia samojen tehtävien kanssa [8].

Aito kiinnostus opetettavien auttamiseen on keskeistä ohjaajan menestyk-

selle [26]. Ohjaaja, joka on töissä vain palkan tai velvollisuuden vuoksi ja pyrkii pelkkään minimisuoritukseen, ei tue oppimista [26]. Huono ohjaaja ei valmistaudu opetustilaisuuteen riittävästi tai vähättelee opetettaviaan tai opetettavaa materiaalia. Opiskelijat tuntevat tulevansa kohdelluksi epäoikeudenmukaisesti, jos he tietävät, että muilla ryhmillä on parempi ohjaaja kuin heillä. Aluksi heikolta ohjaajalta vaikuttava ehdokas voi osoittautua hyväksi ohjaajaksi, jos hänellä on innostusta kehittyä työssään tai hän saa enemmän itseluottamusta kurssin edetessä [9].

Ohjaajan voi edellyttää olevan aktiivinen, vaikka opetettavat eivät kysyisi mitään [36]. Ohjaajan tulee osata huomioida erilaisista taustoista tulevat opiskelijat [14]. Jos ohjaajalla on vastuullaan jokin tietty ryhmä opiskelijoita, hänen tulisi opetella heidän nimensä ja tutustuttava heihin [4].

Pareittain työskenteleminen parantaa ohjaajien suoritusta [20]. Ohjaaja sitoutuu tehtäväänsä paremmin, jos hänellä on pari, joka odottaa, että työtehtävät jaetaan tasavertaisesti. Työparin läsnäolo luo turvallisuuden tunnetta, jos ohjaaja voi luottaa siihen, että pari auttaa ja tukee tarvittaessa. Pareittain työskentely parantaa ohjaajien viihtymistä työssään.

Ohjaajan olisi osattava opetuskieli hyvin [26]. Kielivaikeuksista seuraa kommunikaatio-ongelmia, minkä takia opiskelijat eivät välttämättä ymmärrä ohjaajaa.

Uudet opiskelijat ovat yleensä parempia ohjaajia kuin vanhemmat, opinnoissaan pidemmälle edenneet opiskelijat [9]. Uusilla ohjaajilla on muistissaan hyvä kuva kurssin pääasioista, koska he ovat itse käyneet kurssin vasta vähän aikaa sitten. Pitkälle edenneiden opiskelijoiden saattaa olla vaikeaa ymmärtää kurssin materiaalia, opetustekniikkaa tai alkeiskurssien opiskelijoiden ongelmia, ja heidän voi olla vaikeampi ymmärtää laitoksen tietokoneita [25]. Korkealla tutkintoarvolla ei välttämättä ole paljoa yhteistä opetuksen kanssa, ja materiaalin osaaminen on vain yksi osa opetuksesta, joten pitkälle edenneet opinnot eivät takaa menestystä opettajana [3]. Uusi ohjaaja oppii nopeasti, millainen on hyvä ohjaaja, mikäli hän on juuri itse ollut ohjattavana [35, 36]. Ohjattavien voi olla vaikea suhtautua ohjaajaan, jos tämä on heitä kovin paljon vanhempi [8].

Tohtorin tai maisterin tutkinnon suorittanutta henkilökuntaa voi pitää uusina ja kouluttamattomia opiskelijoita parempana vaihtoehtona ohjaajaksi tai opettajaksi [3]. Kokenut opettaja voi helpommin viitata eri kurssien sisältöihin ja kertoa, miten kurssit liittyvät toisiinsa. Näin hän voi auttaa opiskelijoita näkemään materiaalin laajassa mittakaavassa eikä vain kokelmana erillisiä kursseja. Laajat pohjatiedot voivat auttaa opettajaa tunnistamaan ja omaksumaan nopeasti pääkohdat materiaalista, jota hän ei tunne ennalta. Pitkälle koulutettu henkilökunta voi myös toimia roolimallina opiskelijoille, sillä tutkinnon suorittaminen viittaa kykyyn sitoutua tutkimukseen sekä ajatella luovasti ja itsenäisesti. Kokenut opettaja kannustaa opiskelijoita jatkamaan opinnoissaan ja ajattelemaan tietojenkäsittelytiedettä kiinnostavana tieteenalana teollisuuteen siirtymisen sijaan. Akateemisen uran valit-

sevat opiskelijat tarvitsevat opinnoista asiantuntevaa neuvontaa, jota osaa parhaiten antaa ohjaaja, jolla on kokemusta aiheesta. Tämä auttaa opettajapulaan. Toisaalta opinnoissaan pitkälle edenneitä opiskelijoita voi olla vaikea houkutella ohjaajiksi, koska tietojenkäsittelytieteen alalla on tarjolla paljon muita työmahdollisuuksia [14].

3.3 Ohjaajien koulutus

Uusien ohjaajien kouluttamiseen voi kuulua erillinen tietojenkäsittelytieteen opettamiseen keskittyvä kurssi [25, 28] tai aloitusseminaari [33]. Ohjaajilla voi olla paljon koulutusta opettamansa kurssin alussa, ja koulutustapaamiset vähenevät ja lopulta loppuvat kokonaan kurssin edetessä [28]. Myös teollisuuden edustajat voivat olla kiinnostuneita kouluttamaan ohjaajia [18]. Joskus ohjaajilla ei ole mitään virallista koulutusta, vaan he oppivat työtä tekemällä ja luennoijan ja opiskelijoiden palautteen avulla [32, 35]. Vaikka ohjaamisesta ei olisi varsinaista koulutusta, ohjaajana toimimisen palkitseminen opintopisteillä antaa opiskelijoille kuvan siitä, että kokemuksesta on tarkoitus oppia [36].

Ohjaajien kouluttamiseen liittyy usein keskustelutilaisuuksia. Niissä voidaan esimerkiksi harjoitella tehtävien pisteyttämistä, vaikeiden käsitteiden opettamista tai hankalien opiskelijoiden käsittelyä [24]. Koulutuksessa voidaan käydä läpi erilaisia oppimistyyliä ja keskustella, millainen on hyvä tai huono ohjaaja [13]. Uudet ohjaajat voivat pitää harjoitusesitelmiä toisilleen, mistä he saavat esiintymiskokemusta ja palautetta. Useita erilaisia esitelmiä nähdessään ohjaajat voivat oppia uusia lähestymistapoja aiheeseen.

Ohjaajat voivat harjoitella tehtävien ja kokeiden tekemistä [13]. Ohjaajille voidaan antaa tehtäväksi keskustella, millaisen tietojenkäsittelytieteen alkeiskurssin he suunnittelisivat. Kurssin suunnitteleminen voi saada ohjaajat ajattelemaan tietojenkäsittelytieteen opetukseen liittyviä laajoja kysymyksiä, kuten mitä sivuaineopiskelijoiden tulisi alasta oppia, tai millaiset pohjatiedot uusilla opiskelijoilla voi olettaa tietojenkäsittelytieteestä olevan.

Kokeneet ohjaajat voivat osallistua uusien ohjaajien koulutukseen kerrataksaan omia taitojaan, ja samalla he voivat jakaa kokemustaan ja toimia roolimalleina uusille ohjaajille [25]. Kokeneiden ohjaajien vertaisopetus ja rohkeisuus nopeuttaa uusien ohjaajien koulutusta [8].

Aina muodollista koulutusta ei juuri tarvita tai se ei ole mahdollista, jos uusia ohjaajia hankitaan jatkuvasti opetuksen edetessä. Kokeneet ohjaajat voivat kouluttaa epävirallisesti uusia ohjaajia neuvomalla heitä [16].

Kehittyäkseen työssään ohjaajien on tärkeää saada palautetta [21, 20]. Myönteinen palaute parantaa ohjaajan itseluottamusta, mikä puolestaan parantaa hänen opetustaitojaan.

Vanhat ohjaajat voivat antaa hyödyllistä palautetta ohjaajien koulutuksesta [8]. Ohjaajat voivat suositella opettamiaan opiskelijoita uusiksi ohjaajiksi ja ehdottaa, millaisilla valintaperusteilla saadaan valittua parhaat ohjaajaeh-

dokkaat.

Opiskellessaan opetusta ryhmänä ohjaajat oppivat tuntemaan toisensa ja voivat ystävystyä [28].

3.4 Käytännön organisaatio

Ohjaajia hyödyntävien kurssien henkilökunnan rakenne vaihtelee kurssittain ja yliopistoitain. Esimerkiksi pienessä yliopistossa voi toimia malli, jossa luennoijan lisäksi kurssin henkilökuntaan kuuluu vain muutama ohjaaja [9], kun taas suurissa laitoksissa kurseilla, joilla on paljon opiskelijoita, on hyödyllistä, että kurssin henkilökuntaan kuuluu ohjaajien työtä hallinnoiva assistentti [24]. Assistentti toimii ohjaajien ja luennoijan välisenä kontaktina ja jakaa kurssin materiaalin eteenpäin ohjaajille. Hän voi auttaa luennoijaa kurssin sisällön kanssa. Assistentti voi olla joku kurssin ohjaajista tai erillinen tehtävään palkattu henkilö.

Mikäli henkilökuntaa tarvitaan kurseille paljon, ohjaajien käyttäminen tarvitsee usein toimiakseen jonkinlaisen koordinaattorin [28]. Siinä missä assistentin tehtävä on välittää kurssiin liittyvää tietoa tietyn kurssin luennoijan ja ohjaajien välillä, koordinaattorin kuuluu pitää huolta siitä, että ohjaajatoiminnan rakenne toimii, eli esimerkiksi varmistaa ohjaajien palkkaus ja koulutus. Koordinaattori voi olla esimerkiksi opinnoissaan pitkälle edennyt opiskelija, joka on aiemmin toiminut ryhmänohjaajana. Jos ohjaajia on paljon, voidaan tarvita useampia koordinaattoreita. Ohjaajatoiminnan laadun takaamiseksi jonkun tiedekunnan jäsenen tulisi olla ylimmässä vastuussa koordinaattorien työstä.

Koordinaattorin tehtäviin kuuluu kurssin käytännön hallinnointi, eli esimerkiksi luokkahuoneiden varaaminen ja tarvittavien ohjaajien jakaminen kurseille. He myös hoitavat ohjaajien valinnan, palkkaamisen ja kouluttamisen [25, 28]. Koordinaattori voi huomauttaa ohjaajan puutteellisesta toiminnasta [25].

Koordinaattorien kuuluu ylläpitää tiedon kulkua ja kommunikointia henkilökunnan välillä. Heidän tulee myös järjestää tarvittavat kommunikointikanavat opiskelijoiden ja ohjaajien välille [25]. Koordinaattorien tehtäviin kuuluu järjestää tapaamisia henkilökunnalle, jotta luennoija saa kuvan opiskelijoiden etenemisestä. Viikkopalaverissa ohjaajat voivat kertoa mahdollisista ongelmista, ja luennoija voi tarpeen tullen hidastaa opetustahtia tai selittää jonkin epäselväksi jääneen asian uudestaan. Palaverissa ohjaajat pääsevät tapamaan toisiaan ja jakamaan tietoa.

Koordinaattorien tehtäviin kuuluu usein myös järjestää tapaamisia, joissa käydään läpi opetukseen liittyviä asioita [25, 28]. Koordinaattori voi kerätä opiskelijoilta palautetta ohjaustilaisuuksista, ja esitellä palautteen ohjaajille henkilökunnan tapaamisessa [22]. Palautteen avulla koordinaattori voi suunnitella parannuksia seuraavalle lukukaudelle. Ohjaajan voidaan edellyttää kirjoittavan kurssin koordinaattorille työstään viikottainen raportti, jossa

hän voi kertoa, mikä viikon opetuksessa meni hyvin ja mikä huonosti, ja miten kurssia voidaan parantaa [12].

4 Seurauksia

Opiskelijoiden hyödyntäminen tietojenkäsittelytieteen opetuksessa on mahdollistanut suurempien opiskelijamäärien opettamisen ilman, että kustannukset kohoavat [25]. Samalla opetuksen laatu on noussut, koska opiskelijat saavat aiempaa enemmän yksilöllistä huomiota. Ohjaajilla on myönteinen vaikutus kurssin ja koko yliopiston ilmapiiriin [9, 28, 34].

Vaikka opiskelijoiden käytöstä ohjaajina on havaittu olevan paljon hyötyä, siihen liittyy myös haasteita ja ennakkoluuloja.

4.1 Hyötyjä

Opiskelijoiden käyttämisestä ohjaajina on taloudellista etua yliopistolle, sillä heille maksettava palkka on yleensä matalampi kuin kokeneemmille assistenteille, tai heidät palkitaan opintopisteillä [25]. Arizonan yliopistossa uusien opiskelijoiden palkkaaminen ohjaajiksi kokeneiden assistenttien sijaan laski kustannuksia 85 000 dollarista 75 000 dollariin. Hankkeen alkuvaiheessa säästöt olivat jopa 30 prosenttia, koska päättäjät oli saatava vakuutettua siitä, että ohjaajien palkkaamiseen olisi varaa. Helsingin yliopistossa on havaittu, että ohjaajien hyödyntäminen skaalautuu hyvin myös suurille opiskelijamäärille [16]. Palkkaamalla enemmän ohjaajia luentoja voitiin karsia tai jättää kokonaan pois.

Ryhmänohjaajien käyttäminen voi toimia koulutuksena ja innoituksena uusille luennoijoille yliopiston tulevaisuudessa [28, 18]. Ryhmänohjaajia ohjaavat opettajat saavat tilaisuuden vaikuttaa ohjaajien näkemykseen opetusalaista [18].

Uusien opiskelijoiden käyttäminen ohjaajina on parantanut opiskelijoiden arvosanoja tietojenkäsittelytieteen alkeiskursseilla [8, 16]. Kasvattamalla ohjaajien määrää henkilökohtaisen ohjauksen määrää voi lisätä ja passiivisia luentoja vähentää tai poistaa kokonaan [16].

Koska opiskelijoiden palkkaaminen ohjaajiksi on halvempaa kuin muiden vaihtoehtojen, voi heitä palkata enemmän. Näin opetettavat saavat enemmän yksilöllistä huomiota. Kurssilla voi olla enemmän pieniä tehtäviä, jotka innostavat opiskelijoita paremmin lukemaan materiaalia, kun tehtävien tarkastamiseen on käytössä enemmän henkilökuntaa [9]. Kun ohjaajia on luokassa useita, opiskelijat saavat huomiota nopeammin [10]. Tehtävien suuri määrä vaikuttaa myös oppimiseen, sillä opiskelija oppii materiaalin sitä paremmin, mitä enemmän hän pääsee harjoittelemaan sitä käytännön tehtävissä [36]. Jonkun tietyn henkilökunnan edustajan mahdollinen puolueellisuus ei välttämättä pääse vaikuttamaan opiskelijoihin, jos luokassa on samaan aikaan useita erilaisia ohjaajia [18].

Ohjaajat pystyvät antamaan hyödyllistä palautetta kurssin toiminnasta ja antaa parannusehdotuksia [8] tai huomauttaa, jos materiaalissa on puutteita [9]. Ohjaaja voi olla jonkin tietojenkäsittelyn osa-alueen erityisosaja ja tuoda osaamisensa kurssin hyödyksi. Aiheesta aidosti kiinnostuneilla opiskelijoilla voi olla luennoijaa enemmän aikaa omistautua kurssille. Kun luennoija esittelee kurssin sisällön ohjaajille, hän saa tilaisuuden käydä opetettava materiaali läpi ennen sen esittämistä luennolla [15]. Ohjaajia hyödyntämällä kurssia voi kehittää ketterästi, kun ohjaajat voivat päivittää kurssimateriaalia silloin, kun luennoijalla ei ole aikaa siihen tai jos hän ei ole huomannut sen olevan jäljessä [9].

Yliopistolla työskentelevät ohjaajat voivat toimia roolimalleina muille opiskelijoille [28, 34], sillä opiskelijat tietävät, että ohjaajaksi pääsemiseen vaaditaan kelpoista menestymistä opinnoissa [4]. Opiskelijoiden voi olla vaikea nähdä itsensä esimerkiksi kokeneena luennoijana, kun taas ohjaaja tarjoaa helpommin lähestyttävämmän roolimallin [30]. Jos ohjaajat valitaan opiskelijoiden keskuudesta, voidaan henkilökunnasta saada aiempaa monipuolisempi ja siten tarjota paljon erilaisia roolimalleja [18]. Etenkin naisille ja vähemmistöjen edustajille voi olla tärkeää saada kaltaisansa roolimalli, jotta heidän on helpompi tuntea kuuluvansa joukkoon. Vähemmistöopiskelijat voivat jäädä opiskelun ulkopuolisten epämuodollisten ryhmien ulkopuolelle. Ryhmänohjaajien käyttäminen mahdollistaa paremmin pienryhmissä opiskelun, joka auttaa opiskelijoita tutustumaan toisiinsa. Yhteenkuuluvuuden tunne saa opiskelijat pysymään alalla paremmin.

Kurssin ilmapiiriä rentouttaa, jos opiskelijat näkevät luennoijan ja ohjaajan tulevan hyvin toimeen keskenään [9]. Mikäli ohjaaja on läsnä esimerkiksi luennolla ja esittää luennotsijalle kysymyksiä aiheesta, rohkaisee se opiskelijoitakin osallistumaan enemmän luennoilla. Kurssin opiskelijat hyötyvät, jos heidän keskuudessaan on ohjaaja, joka osaa heti kurssin alussa kertoa luennoijan opetustyylistä ja painotuksista. Motivoitunut ohjaaja motivoi opetettaviakin työskentelemään ahkerammin. Kurssin opiskelijoiden voi olla helpompi antaa palautetta ohjaajalle kuin luennoijalle [18].

Ohjaajat saavat palkkaa tai opintopisteitä, minkä lisäksi opettaminen parantaa heidän ryhmätyö- ja esiintymistaitojaan. Ohjaajat oppivat keskustelemaan tietokonetaidoiltaan eritasoisten opiskelijoiden kanssa [36]. Opettaessaan he oppivat itsekin materiaalin paremmin [24], koska jonkin asian osaa kunnolla vasta sitten, kun osaa opettaa sen toiselle [4, 9]. Opetettavien kannustaminen hyvään ohjelmointityyliin parantaa ohjaajankin ohjelmointityyliä [28]. Ohjaajana toimiminen opettaa vastuukantoa ja liittää opiskelijan paremmin mukaan yliopiston yhteisöön [9]. Koska ohjaajat yleensä ovat tekemisissä luennoijan kanssa tavallisia opiskelijoita enemmän, heillä on tilaisuus saada ylimääräistä ohjausta opintoihinsa [2]. Ohjaajana toimiminen opettaa ajan hallintaa. Opettamalla muita ohjaaja voi oppia omista opiskelutottumuksistaan [10].

Opettaessaan muita ohjaaja oppii paremmin ymmärtämään tarpeen jatku-

valle elinikäiselle opiskelulle [23]. Ohjaajana toimimisesta voi saada hyvän mielen päästessään auttamaan muita [2] ja saadessaan nähdä opetettaviensa edistyvän [16]. Ohjaajat voivat saada toisistaan seuraa ja osallistua keskenään erilaisiin vapaa-ajan aktiviteetteihin [28]. Kokemus ohjaajana toimimisesta näyttää hyvältä ansioluettelossa [10].

4.2 Haasteita

On pelätty, että uusien ja vähän koulutettujen opiskelijoiden käyttäminen ohjaajina ei olisi hyödyllistä opetettaville [11]. Ohjaajien puutteellinen kokemus tai koulutus opetuksesta voi vaikeuttaa ohjausta tai epäonnistumisen pelkoa ohjaajille [17]. Aiheesta kirjoitetut artikkelit kuitenkin esittävät, että hyvän ohjaajan ei tarvitse olla opinnoissaan kovin paljoa edellä opetettaviaan. Eri oppilaitosten välillä voi olla eroja siinä, kuinka päteviä tai luotettavia ohjaajat ovat tehtävissään [24].

Uusien opiskelijoiden palkkaaminen ohjaajiksi voi olla taloudellisesti epävarmaa, jos ohjaajat ilmoittavat työtuntinsa viikoittain [24]. Tämän vuoksi on vaikea etukäteen ennustaa tarkkaan, kuinka paljon ohjaajien käyttäminen tulee maksamaan, jolloin päättäjien vakuuttaminen ohjaajien käyttämisen eduista voi olla vaikeaa.

Lakisääteiset syyt voivat rajoittaa opiskelijoiden hyödyntämistä opetuksessa. Säädökset voivat esimerkiksi kieltää liian pienet kurssit. Vaikka jollakin tietojenkäsittelytieteen alkeiskurssilla olisi valtavat määrät opiskelijoita, voi kurssi tilastollisesti näyttää monelta pieneltä, jos opiskelijat on jaettu pieniin ryhmiin [24]. Säädökset voivat myös kieltää opiskelijoita arvostelemasta saman asteen opiskelijoiden tehtäviä varmistaakseen, että yliopistot eivät laiminlyö velvollisuuksiaan arvostelussa. Tätä sääntöä voi kiertää esimerkiksi arvostelemalla tehtävät laadullisesti siltä kannalta, kuinka ohjelma yleisesti ottaen toimii eikä keskittymällä koodin yksityiskohtiin [9]. Tehtäviä voi tarkistaa myös automaattisesti tietokoneen avulla [35].

Ohjaajien palkitseminen opintopisteillä palkan sijaan on jakanut mielipiteitä. Vaikka muillakin tieteenaloilla opiskelijoita palkitaan opintopisteillä tutkimusprojekteihin osallistumisesta, on esitetty, että opettamiseen osallistuminen ei olisi yhtä arvokasta kuin tutkimukseen osallistuminen [25].

Vaikka ohjaajat pääasiassa hyötyvät siitä, että he ovat suurin piirtein saman ikäisiä kuin opetettavansa, voi siitä seurata myös ongelmia tehtävien arvostelun suhteen [28]. Koska ohjaajalla ei mahdollisesti ole paljon kokemusta tai auktoriteettia, voi hänen olla vaikea käsitellä opiskelijoita, jotka ovat mielestään saaneet liian huonon arvosanan. Lisäksi uusien ohjaajien voi olla vaikea arvioida, minkä arvosanan antaa epätavalliselle ratkaisulle. Ohjaajilla pitäisikin olla selvät ohjeet tehtävien pisteyttämisen suhteen omaperäisten ratkaisujen tai kiistatilanteiden varalta. Toisaalta jos pidetään tärkeänä sitä, että opiskelijat mieltävät ohjaajat vertaisekseen, voi olla parempi, että ohjaajat eivät osallistu tehtävien pisteyttämiseen ja opiskelijoiden arvosteluun

[18]. Jotkut opiskelijat voivat pitää epämiellyttävänä sitä, että heitä opettaa heitä itseään huomattavasti nuorempi ohjaaja [33].

Jatkuva vaihtelu opettajan ja opiskelijan roolin välillä voi olla rasittavaa ohjaajalle [17]. Ulkoa opettelua korostavaan opettajakeskeiseen opetukseen tottuneen ohjaajan voi olla vaikeaa siirtyä opettajan ja opiskelijan vuorovai-
kutusta korostavaan opetustyyliin.

Koulutuksen ja kokemuksen puutteen takia ympäristön häiriöt voivat haitata ohjaajan työskentelyä kokenutta opettajaa enemmän [20]. Esimerkiksi luok-
kahuoneen valaistus, pöytien järjestely ja opetustilanteen kesto vaikuttavat ohjaajan työhön.

Aina ohjaajille ei ole tarjolla tarvittavaa koulutusta [32]. Syitä koulutuksen puutteeseen voivat olla rahan tai henkilökunnan puute tai ajatus, että ohjaa-
jat eivät tarvitse koulutusta. Henkilökunnalta voi puuttua kiinnostusta tai taitoa avustaa ohjaajia.

Jos tutkimusavustajana toimimista pidetään arvokkaampana toimintana kuin opettamista, opiskelijat voivat olla kiinnostuneempia tutkimukseen pyrkimi-
sestä kuin ohjaajana toimimisesta [32]. Usein tutkimusavustajia rahoitetaan ohjaajia paremmin ja heillä on enemmän tilaisuuksia työskennellä yliopiston
henkilökunnan kanssa. Uhkana voi olla, että parhaat opiskelijat päätyvät tutkimusavustajiksi opettamisen sijaan.

5 Yhteenveto

Tässä artikkelissa käsitelimme opiskelijoiden hyödyntämistä tietojenkäsittelytieteen opetuksessa. Opiskelijoiden käyttö opetuksessa mahdollistaa suu-
rempien opiskelijamäärien opettamisen ilman, että kustannukset nousevat tai että olisi turvauduttava passiivisiin massaluentoihin. Ohjaajien käyttö antaa
mahdollisuuden tarjota opiskelijoille aiempaa enemmän yksilöllistä palau-
tetta ja neuvontaa, mikä nostaa opetuksen laatua. Ohjaajat työskentelevät lähellä opiskelijoita, joten he pystyvät antamaan luennoijalle palautetta kurs-
sin etenemisestä. Ohjaajana toimiminen antaa opiskelijalle hyödyllisiä taitoja ja kokemusta, ja ohjaajien läsnäolo kursseilla rentouttaa kurssin ilmapiiriä
ja tarjoaa opiskelijoille helposti lähestyttäviä roolimalleja. Opiskelijoiden hyödyntämisen opetuksessa on havaittu toimivan niin suurissa [24] kuin pie-
nissäkin [9] oppilaitoksissa.

Vaikka opiskelijoita on hyödynnetty tietojenkäsittelytieteen opetuksessa jo pitkään, aiheetta on tutkittu niukasti. Opiskelijoiden käyttämistä ohjaajina
käsittelevät artikkelit antavat usein yksipuolisen myönteisen kuvan aiheesta, koska mahdollisista haitoista tai huonosti toimivan toteutuksen riskeistä on
kirjoitettu vain vähän.

6 Lähteet

- [1] *Deep, Surface and Strategic Learning*. 2013. <http://www.scu.edu.au/teachinglearning/index.php/17>.
- [2] Aminzadeh, R.: *Being a Teaching Assistant Can Play an Important Role in Your Future Career*. Potentials, IEEE, 31(4):14–16, Aug., ISSN 0278-6648.
- [3] Baldwin, Doug: *Some thoughts on undergraduate teaching and the Ph.D.* SIGCSE Bull., 32(4):60–62, joulukuu 2000, ISSN 0097-8418. <http://doi.acm.org/10.1145/369295.369327>.
- [4] Bernstein, G.H.: *The role of the lab TA [teaching assistants]*. Potentials, IEEE, 21(3):36–38, Aug/Sep, ISSN 0278-6648.
- [5] Biggs, J. ja C. Tang: *Teaching for quality learning at university: what the student does*. Society for Research into Higher Education. McGraw-Hill, 2007, ISBN 9780335221264. <http://books.google.fi/books?id=TBadAAAAMAAJ>.
- [6] Boyer, Kristy Elizabeth, Rachael S. Dwight, Carolyn S. Miller, C. Dianne Raubenheimer, Matthias F. Stallmann ja Mladen A. Vouk: *A case for smaller class size with integrated lab for introductory computer science*. SIGCSE Bull., 39(1):341–345, maaliskuu 2007, ISSN 0097-8418. <http://doi.acm.org/10.1145/1227504.1227430>.
- [7] Chamillard, A. T. ja Laurence D. Merkle: *Management challenges in a large introductory computer science course*. SIGCSE Bull., 34(1):252–256, helmikuu 2002, ISSN 0097-8418. <http://doi.acm.org/10.1145/563517.563440>.
- [8] Decker, Adrienne, Phil Ventura ja Christopher Egert: *Through the looking glass: reflections on using undergraduate teaching assistants in CS1*. SIGCSE Bull., 38(1):46–50, maaliskuu 2006, ISSN 0097-8418. <http://doi.acm.org/10.1145/1124706.1121358>.
- [9] Dickson, Paul E.: *Using undergraduate teaching assistants in a small college environment*. Teoksessa *Proceedings of the 42nd ACM technical symposium on Computer science education*, SIGCSE '11, sivut 75–80, New York, NY, USA, 2011. ACM, ISBN 978-1-4503-0500-6. <http://doi.acm.org/10.1145/1953163.1953187>.
- [10] Ferstl, A.: *Teaching and Learning Science with Learning Assistants*. ArXiv e-prints, heinäkuu 2010.

- [11] Harper, Kathleen Andre, David B. May ja Keith W. Oliver: *Using Undergraduate Students as Physics Lab Teaching Assistants*. The Physics Teacher, 40(4):226–228, 2002. <http://link.aip.org/link/?PTE/40/226/1>.
- [12] Hug, Sarah, Heather Thiry ja Phyllis Tedford: *Learning to love computer science: peer leaders gain teaching skill, communicative ability and content knowledge in the CS classroom*. Teoksessa *Proceedings of the 42nd ACM technical symposium on Computer science education*, SIGCSE '11, sivut 201–206, New York, NY, USA, 2011. ACM, ISBN 978-1-4503-0500-6. <http://doi.acm.org/10.1145/1953163.1953225>.
- [13] Kay, David G.: *Training computer science teaching assistants: a seminar for new TAs*. SIGCSE Bull., 27(1):53–55, maaliskuu 1995, ISSN 0097-8418. <http://doi.acm.org/10.1145/199691.199719>.
- [14] Kay, David G.: *Large introductory computer science classes: strategies for effective course management*. SIGCSE Bull., 30(1):131–134, maaliskuu 1998, ISSN 0097-8418. <http://doi.acm.org/10.1145/274790.273177>.
- [15] Kopp, Sacha E.: *Undergraduate peer assistants in a large lecture course*. Physics Education, 35(6):423, 2000. <http://stacks.iop.org/0031-9120/35/i=6/a=308>.
- [16] Kurhila, Jaakko ja Arto Vihavainen: *Management, structures and tools to scale up personal advising in large programming courses*. Teoksessa *Proceedings of the 2011 conference on Information technology education*, SIGITE '11, sivut 3–8, New York, NY, USA, 2011. ACM, ISBN 978-1-4503-1017-8. <http://doi.acm.org/10.1145/2047594.2047596>.
- [17] Mark, K., D.R. Thadani, D.S. Calonge, C.F.K. Pun ja P.H.P. Chiu: *In-service Teaching Assistant Training (InsTAT) for engineering and computer science graduate students in Hong Kong: A blended-learning approach*. Teoksessa *Frontiers in Education Conference (FIE)*, 2011, sivut F2E–1–F2E–6, 2011.
- [18] Morgan, J., A. Kenimer, T. Kohutek, J. Rinehart ja M. Lee: *Peer teacher from an instructor's perspective*. Teoksessa *Frontiers in Education, 2002. FIE 2002. 32nd Annual*, nide 3, sivut S2C–11 – S2C–15 vol.3, nov. 2002.
- [19] Murphy, Christian, Rita Powell, Kristen Parton ja Adam Cannon: *Lessons learned from a PLTL-CS program*. Teoksessa *Proceedings of the 42nd ACM technical symposium on Computer science education*, SIGCSE '11, sivut 207–212, New York, NY, USA,

2011. ACM, ISBN 978-1-4503-0500-6. <http://doi.acm.org/10.1145/1953163.1953226>.
- [20] Patitsas, Elizabeth: *A case study of environmental factors influencing teaching assistant job satisfaction*. Teoksessa *Proceedings of the ninth annual international conference on International computing education research*, ICER '12, sivut 11–16, New York, NY, USA, 2012. ACM, ISBN 978-1-4503-1604-0. <http://doi.acm.org/10.1145/2361276.2361280>.
 - [21] Patitsas, Elizabeth ja Patrice Belleville: *What can we learn from quantitative teaching assistant evaluations?* Teoksessa *Proceedings of the Seventeenth Western Canadian Conference on Computing Education*, WCCCE '12, sivut 36–40, New York, NY, USA, 2012. ACM, ISBN 978-1-4503-1407-7. <http://doi.acm.org/10.1145/2247569.2247582>.
 - [22] Patitsas, Elizabeth Ann ja Steven Andrew Wolfman: *Effective closed labs in early CS courses: lessons from eight terms of action research*. Teoksessa *Proceedings of the 43rd ACM technical symposium on Computer Science Education*, SIGCSE '12, sivut 637–642, New York, NY, USA, 2012. ACM, ISBN 978-1-4503-1098-7. <http://doi.acm.org/10.1145/2157136.2157318>.
 - [23] Paxton, John: *Undergraduate consultation: opportunities and challenges*. J. Comput. Sci. Coll., 21(1):231–238, lokakuu 2005, ISSN 1937-4771. <http://dl.acm.org/citation.cfm?id=1088791.1088832>.
 - [24] Reges, Stuart: *Using undergraduates as teaching assistants at a state university*. SIGCSE Bull., 35(1):103–107, tammikuu 2003, ISSN 0097-8418. <http://doi.acm.org/10.1145/792548.611943>.
 - [25] Reges, Stuart, John McGrory ja Jeff Smith: *The effective use of undergraduates to staff large introductory CS courses*. SIGCSE Bull., 20(1):22–25, helmikuu 1988, ISSN 0097-8418. <http://doi.acm.org/10.1145/52965.52971>.
 - [26] Richards, L.G.: *Teaching graduate teaching assistants (GTAs) how to teach*. Teoksessa *Frontiers in Education Conference, 2000. FIE 2000. 30th Annual*, nide 2, sivut F3F/14 –F3F/20 vol.2, 2000.
 - [27] Roberts, Eric: *Conserving the seed corn: reflections on the academic hiring crisis*. SIGCSE Bull., 31(4):4–9, joulukuu 1999, ISSN 0097-8418. <http://doi.acm.org/10.1145/349522.349363>.
 - [28] Roberts, Eric, John Lilly ja Bryan Rollins: *Using undergraduates as teaching assistants in introductory programming courses: an update on*

- the Stanford experience*. SIGCSE Bull., 27(1):48–52, maaliskuu 1995, ISSN 0097-8418. <http://doi.acm.org/10.1145/199691.199716>.
- [29] Roberts, Eric S.: *Meeting the challenges of rising enrollments*. ACM Inroads, 2(3):4–6, elokuu 2011, ISSN 2153-2184. <http://doi.acm.org/10.1145/2003616.2003617>.
 - [30] Roberts, Eric S., Marina Kassianidou ja Lilly Irani: *Encouraging women in computer science*. SIGCSE Bull., 34(2):84–88, kesäkuu 2002, ISSN 0097-8418. <http://doi.acm.org/10.1145/543812.543837>.
 - [31] Settle, Amber: *Turning the tables: learning from students about teaching CS1*. Teoksessa *Proceedings of the 13th annual conference on Information technology education*, SIGITE '12, sivut 133–138, New York, NY, USA, 2012. ACM, ISBN 978-1-4503-1464-0. <http://doi.acm.org/10.1145/2380552.2380594>.
 - [32] Shannon, David M., Darla J. Twale ja Mathew S. Moore: *TA Teaching Effectiveness: The Impact of Training and Teaching Experience*. The Journal of Higher Education, 69(4):pp. 440–466, 1998, ISSN 00221546. <http://www.jstor.org/stable/2649274>.
 - [33] Sperry, Rita A. ja Phyllis Tedford: *Implementing peer-led team learning in introductory computer science courses*. J. Comput. Sci. Coll., 23(6):30–35, kesäkuu 2008, ISSN 1937-4771. <http://dl.acm.org/citation.cfm?id=1352383.1352390>.
 - [34] Tashakkori, Rahman, James T. Wilkes ja Edward G. Pekarek: *A systemic mentoring model in computer science*. Teoksessa *Proceedings of the 43rd annual Southeast regional conference - Volume 1*, ACM-SE 43, sivut 371–375, New York, NY, USA, 2005. ACM, ISBN 1-59593-059-0. <http://doi.acm.org/10.1145/1167350.1167453>.
 - [35] Vihavainen, Arto, Thomas Vikberg, Matti Luukkainen ja Jaakko Kurhila: *Massive Increase in Eager TAs: Experiences from Extreme Apprenticeship-based CS1*.
 - [36] Vikberg, Thomas, Arto Vihavainen, Matti Luukkainen ja Jaakko Kurhila: *Early Start in Software Coaching*.
 - [37] Yadin, Aharon: *Reducing the dropout rate in an introductory programming course*. ACM Inroads, 2(4):71–76, joulukuu 2011, ISSN 2153-2184. <http://doi.acm.org/10.1145/2038876.2038894>.