

Opiskelijoiden hyödyntäminen tietojenkäsittelytieteen opettamisessa

Hanna Arpiainen

tutkielma
Helsingin Yliopisto
Tietojenkäsittelytieteen laitos

Helsinki, 7. 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 tutkielma	Aika — Datum — Month and year 7. huhtikuuta 2013	Sivumäärä — Sidoantal — Number of pages 14	
Tiivistelmä — Referat — Abstract tutkielma.			
Avainsanat — Nyckelord — Keywords avainsana 1, avainsana 2, avainsana 3			
Säilytyspaikka — Förvaringsställe — Where deposited			
Muita tietoja — Övriga uppgifter — Additional information			

Sisältö

1	Johdanto	3
2	Muutokset tietojenkäsittelytieteen opetuksessa	3
2.1	Opiskelijamäärien kasvu	3
2.2	Pula opettajista	3
2.3	Suurten luentojen ongelmat	4
3	Oppilaat opettajina	5
3.1	Ohjaajien työtehtäviä	5
3.2	Millainen on hyvä ohjaaja?	6
3.3	Ohjaajien koulutus	7
3.4	Käytännön organisaatio	8
4	Seurauksia	9
4.1	Hyötyjä	9
4.2	Haasteita	10
5	Yhteenveto	11
6	Lähteet	11

1 Johdanto

Tietojenkäsittelytieteen opetuksen tarve on kasvanut. Ongelmana on usein taloudellisten resurssien puute, mikä johtaa opiskelupaikkojen lukumäärän rajoittamiseen. Lisäksi etenkin 1980-luvulla osaavista opettajista oli pulaa, joten henkilökuntaan jouduttiin palkkaamaan riittämättömästi koulutettuja ulkopuolisia [21]. Teollisuuden menestys houkuttelee opiskelijoita siirtymään nopeasti työelämään opintojen jatkamisen sijaan, jolloin on yhä vaikeampi saada uusia opettajia uusille tietojenkäsittelytieteen opiskelijoille [19].

Ongelmaa voi ratkaista kehittämällä yhteistyötä teollisuuden kanssa, koska sillä on taloudellisia resursseja tukea opetusta [21]. Toinen ratkaisu on palkata alkeiskursseille mahdollisimman pätevää henkilökuntaa.

Kolmas ratkaisu ongelmaan on hyödyntää opiskelijoita opetuksessa, mihin tässä tutkielmassa keskitytään. 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 esimerkiksi nostavan opetuksen laatua ja parantavan kurssin ilmapiriä.

Vaikka opiskelijoita on hyödynnetty tietojenkäsittelytieteen opetuksessa jo pitkään, aihetta 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.

2 Muutokset tietojenkäsittelytieteen opetuksessa

2.1 Opiskelijamäärien kasvu

Tietojenkäsittelytieteen opiskelijoiden lukumäärän vaihtelu on syklistä [21]. Esimerkiksi kotitietokoneiden ja World Wide Webin yleistymisen kasvattivat alan suosiota ja opiskelijoiden määrä kasvoi, kun taas IT-kuplan puhkeaminen 2000-luvun alussa vähensi opiskelijamäärää. Ajoittaisista taantumisista huolimatta opiskelijoiden määrä on pitkällä ajanjaksolla kasvanut, ja teollisuus tarvitsee yhä enemmän osaajia. Muiden aineiden opiskelijat haluavat opiskella tietojenkäsittelytieteen perusteita, koska he uskovat, että ohjelmointitaidot auttavat heitä työmarkkinoilla. Tämä lisää tietojenkäsittelytieteen opetuksen tarvetta. Etenkin alkeiskursseille pitäisi siis pystyä järjestämään riittävästi opetusta.

2.2 Pula opettajista

Tietojenkäsittelytieteen alaa on vaivannut puute pätevistä opettajista [19]. Teollisuuden menestys houkuttelee opiskelijat jättämään opintonsa ja siir-

tymään töihin, jolloin on yhä vaikeampi kouluttaa uusia opettajia. Työvoimapula aiheuttaa ylitöitä ja stressiä opettajille, mikä lisää heidän ovat halukkuuttaan siirtyä muihin töihin. Opettajien puute vaikeuttaa kasvaviin opiskelijamääriin vastaamista.

2.3 Suurten luentojen ongelmat

Kurssikoon kasvaessa kasvaa myös kurssin tehtävien korjaukseen ja opiskelijoiden henkilökohtaiseen ohjaukseen tarvittava työmäärä [7]. Suurissa opiskelijamäärissä voi parhaiden ja heikoimpien opiskelijoiden ero olla hyvin suuri, joten ääripäät tarvitsisivat erityisen paljon yksilöllistä huomiota. Kokeiden ja tehtävien tekeminen vaikeutuu oppilasmäärän kasvaessa, jos tehtävistä pitää saada mahdollisimman nopeasti tarkistettavia. Tehtävät tulisi myös olla määritellä mahdollisimman selkeästi, sillä epäselvien kysymysten aiheuttamia väärinymmärryksiä tulee suurten opiskelijamäärien kanssa enemmän. Suurella kurssilla voi olla haasteellista taata, että kaikki opiskelijat arvostellaan tasavertaisesti [2]. Kurssikoon kasvaessa vaikeutuu myös sen hallinnointi, kuten kurssinateriaalin valmistelu ja jakaminen opiskelijoille, neuvonnan tarjoaminen tehtävissä apua tarvitseville opiskelijoille ja henkilökunnan hallinnointi.

Suurten ryhmäkokojen seurauksena opiskelijat uhkaavat hukkua massaan ja voivat sen seurauksena menettää mielenkiintonsa alaa kohti [7]. Pelkkä luentojen passiivinen kuuntelu ei rohkaise opiskelijoita osallistumaan [8]. Uusien opiskelijoiden voi olla hankala lähestyä kokenutta luennoijaa, ja luentoihin painottuvilla kursseilla voi olla vaikeuksia tarjota opiskelijoille roolimalleja. Monilla opiskelijoilla ei ole tietojenkäsittelytieteestä aiempaa kokemusta, ja suuret luennot ja yksin tehtävät harjoitukset voivat tuottaa heille hankaluuksia [12].

Välttääkseen syrjäytymisen tunnetta opiskelijan olisi tärkeää päästä opiskeluun mukaan heti opintojen alussa [23]. Yliopisto-opintojen itsenäisyys voi olla uutta monille opiskelijoille, ja heidän voi olla helppo oppia huonoja opiskelutapoja ja olla välittämättä läsnäolon tärkeydestä kursseilla.

Suurten luentojen passiivisella kuuntelemisella ja asioiden ulkoa opettelulla opiskelija voi yltää pintapuoliseen osaamiseen, mutta syvempään osaamiseen vaaditaan aktiivista opiskelua [?]. Aktiivinen osallistuminen ehkäisee opintojen kesken jättämistä. Opiskelijoilla voi olla vaikeuksia keskittyä ja seurata opetusta suurilla luennoilla. Suurten luentokurssien opiskelijoilla voi olla heikompi itseluottamus omien kykyjensä, opintomenestyksensä tai tietojenkäsittelytieteen hyödyllisyyden tulevaisuudessa suhteen kuin pienryhmissä opiskelevilla.

3 Oppilaat opettajina

Kasvaviin opiskelijamääriin, suuriin luentokursseihin ja työvoimapulaan liittyviä ongelmia voi ratkaista käyttämällä opiskelijoita tietojenkäsittelytieteen opetuksessa. Opiskelijat voivat toimia ohjaajana ryhmälle alkeiskurssin opiskelijoita [17]. Ohjaajat tarjoavat lisäopetusta kurssin varsinaisten luentojen tueksi ja auttavat kurssin opiskelijoita tehtävien kanssa. Ohjaajilla ei kuitenkaan välttämättä tarvitse olla omaa vastuuryhmäänsä, vaan he voivat antaa yleistä neuvontaa tietokoneluokissa [26, 25]. Joissakin laitoksissa opiskelijat käyttävät enemmän aikaa ohjaajien kuin luennoijan kanssa [13].

Ryhmänohjaajien lisäksi opiskelijoita voi käyttää opetuksessa vertaistukena [24]. Vertaistukea voi saada opiskelija, joka tarvitsee lisää tukea opintoihinsa, tai joka tarvitsee ylimääräisiä haasteita. Vertaismentorin tulee osata opettamansa kurssin sisältö hyvin. Mentori tapaa neuvottavaansa ja avustaa tehtävien kanssa. Mentorointi voi mennä myös ristiin; opiskelija voi olla mentorina toiselle yhdellä kurssilla ja neuvottavansa mentoroitavana jollakin toisella.

Opiskelijoiden käyttäminen ohjaajina vaatii laitokselta sitoutumista [8]. Koska ohjaajien käyttämiseen tarvitaan ennakkoinvestioita, kannattaa laitoksen järjestää ohjaajia vain kursseille, joita siellä on aiemmin opetettu.

Joissakin laitoksissa voi ohjaajana toimiminen olla pakollinen osa tutkintoa [6].

3.1 Ohjaajien työtehtäviä

Ryhmänohjaajien tehtäviin kuuluu esimerkiksi viikottaisten keskustelusesioden pitäminen [17]. Keskustelusesiossa 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 [3]. Tarvittaessa ryhmänohjaajan on oltava valmis hylkäämään aiemmat suunnitelmansa, mikäli selviää, että ryhmä on ymmärtänyt jonkin tärkeän asian väärin ja tilanne on korjattava nopeasti [17].

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 edistymisestä kurssilla [17, 16]. Opiskelijan on tärkeä saada mielekästä palautetta edistymisestään [9].

Ryhmänohjaajat voivat päivystää tietokoneluokassa valmiina auttamaan kurssin opiskelijoita kurssiin liittyvissä ongelmissa. Ruuhka-ajoille päivystäviä

ohjaajia pyritään järjestämään enemmän [17, 16], ja ohjaajat saattavat voida kutsua lisääpua, mikäli tietokoneluokka ruuhkautuu pahasti [9]. Ohjaajan ei tulisi antaa suoraa vastausta tehtävään, vaan ohjata ja neuvoa opetettavaa löytämään ratkaisu itse [26, 9]. Ohjelmointitehtävien yhteydessä ohjaaja voi kannustaa opetettavaa kirjoittamaan hyvää ja luettavaa koodia esimerkiksi nimeämällä muuttujat selkeästi. Ohjaajat voivat erottuakseen joukosta täydessä tietokoneluokassa käyttää esimerkiksi turvaliivejä [25].

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 [3].

Usein ohjaajan tehtäviin kuuluu tehtävien tarkistaminen [4]. Kun ohjaajat osallistuvat tehtävien tarkastamiseen, voi tehtäviä olla enemmän, ja luennoija voi käyttää aikansa tehokkaammin esimerkiksi kirjoittamalla kommentteja tehtäviin. Ohjaajat voivat käydä kurssin materiaalin läpi ennen, kuin luennoija esittää sen opiskelijoille, ja huomauttaa mahdollisista virheistä tai puutteista [26]. Ohjaajat voivat ratkaista kurssin tehtävät ennen kuin ne annetaan opiskelijoille, mikä auttaa heitä valmistautumaan neuvontaan [25]. Ohjaajan voidaan edellyttää osallistuvan kurssin luennoille pysyäkseen mahdollisimman hyvin perillä kurssin etenemisestä ja käsitellyistä esimerkeistä [16, 3]. Ohjaaja voi päivittää kurssin verkkosivuja [4], mikä vähentää luennoijan työtaakkaa ja auttaa opiskelijoita, joilla on vaikeuksia omien muistiinpanojen tekemisen kanssa. Luennoille osallistuva ohjaaja voi päivittää esimerkkejä verkkosivuille reaaliajassa.

3.2 Millainen on hyvä ohjaaja?

On vaikea määritellä tarkkaan, millainen on hyvä ohjaaja, koska niin monet erilaiset ohjaajat ovat onnistuneet tehtävässään. Ohjaajalta vaaditaan kuitenkin aitoa innostusta auttaa muita oppimaan ja vastuullisuutta noudattaa kurssin käytäntöjä. Ohjaajan ei ole välttämätöntä olla ohjelmoinnin erityisosaaja, mutta hänen edellytetään ymmärtävän kurssin sisältö ja olevan valmis kehittämään itseään pystyäkseen auttamaan opetettaviaan riittävästi [17]. Ohjaaja voi jopa hyötyä siitä, että hänellä on itsellään ollut oppimisvaikeuksia, sillä se voi auttaa häntä ymmärtämään paremmin opetettaviensa vaikeuksia samojen tehtävien kanssa [3].

On havaittu, että uudet opiskelijat ovat yleensä parempia ohjaajia kuin vanhemmat, opinnoissaan pidemmälle edenneet opiskelijat [4]. Uusilla ohjaajilla on muistissaan hyvä kuva kurssin pääasioista, koska he ovat itse käyneet kurssin vasta vähän aikaa sitten. Pitkälle edenneiden opiskelijoiden taas saattaa olla vaikeaa ymmärtää kurssin materiaalia, opetustekniikkaa tai alkeiskurssien opiskelijoiden ongelmia, ja heidän voi olla vaikeampi ymmärtää laitoksen tietokoneita [17]. Korkealla tutkintoarvolla ei välttämättä ole paljoa yhteistä opetuksen kanssa, ja materiaalin osaaminen on vain yksi osa ope-

tuksesta, joten pitkälle edenneet opinnot eivät takaa menestystä opettajana [1]. Uusi ohjaaja oppii nopeasti, millainen on hyvä ohjaaja, mikäli hän on juuri itse ollut ohjattavana [25, 26]. Ohjattavien voi olla vaikea suhtautua ohjaajaan, jos tämä on heitä kovin paljon vanhempi [3].

Tohtorin tai maisterin tutkinnon suorittanutta henkilökuntaa voi pitää uusia ja kouluttamattomia opiskelijoita parempana vaihtoehtona ohjaajaksi tai opettajaksi [1]. 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 yksittäisiä ja 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 tutkintonsuorittaminen viittaa kykyyn sitoutua tutkimukseen ja ajatella luovasti ja itsenäisesti. Kokenut opettaja kannustaa opiskelijoita jatkamaan opinnoissaan ja ajattelemaan tietojenkäsittelytiedettä kiinnostavana tieteenalana pelkän teollisuuteen siirtymisen sijaan. Tämä auttaa opettajapulaan.

Aito kiinnostus opetettavien auttamiseen on keskeistä ohjaajan menestykselle. Ohjaaja, joka on töissä vain palkan vuoksi ja pyrkii pelkkään minimisuoritukseen eikä valmistaudu opetustilaisuuteen riittävästi tai vähättelee opetettaviaan tai opetettavaa materiaalia, ei tue oppimista [18]. Ohjaajan voi edellyttää olevan aktiivinen, vaikka opetettavat eivät itse kysyisikään mitään [26].

Pareittain työskenteleminen parantaa ohjaajien suoritusta [13]. 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.

3.3 Ohjaajien koulutus

Uusien ohjaajien kouluttamiseen voi kuulua erillinen tietojenkäsittelytieteen opettamiseen keskittyvä kurssi [17, 20]. Myös teollisuuden edustajat voivat olla kiinnostuneita kouluttamaan ohjaajia [11].

Ohjaajien kouluttamiseen liittyy usein keskustelutilaisuuksia. Niissä voidaan esimerkiksi harjoitella tehtävien pisteyttämistä, vaikeiden käsitteiden opettamista tai hankalien opiskelijoiden käsittelyä [16]. Koulutuksessa voidaan käydä läpi erilaisia oppimistyyplejä ja keskustella, millainen on hyvä tai huono ohjaaja [6]. Uudet ohjaajat voivat pitää harjoitusesitelmää toisilleen, mistä he saavat esiintymiskokemusta ja palautetta. Useita erilaisia esitelmää nähdessään ohjaajat voivat oppia uusia lähestymistapoja aiheeseen. Ohjaajat voivat harjoitella tehtävien ja kokeiden tekemistä. Ohjaajille voidaan antaa tehtäväksi keskustella, millaisen tietojenkäsittelytieteen alkeiskurssin he suunnittelisivat. Kurssin suunnitteleminen itse voi saada ohjaajat ajat-

telemaan 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 kerratukseen omia taitojaan, ja samalla he voivat jakaa kokemustaan ja toimia roolimalleina uusille ohjaajille [17]. Kokeneiden ohjaajien vertaisopetus ja rohkaisu nopeuttaa uusien ohjaajien koulutusta [3]. Aina muodollista koulutusta ei juuri tarvita, vaan kokeneet ohjaajat kouluttavat uusia neuvomalla [9].

Kehittyäkseen työssään ohjaajien on tärkeää saada palautetta [14, 13]. Myönteinen palaute parantaa ohjaajan itseluottamusta, mikä puolestaan parantaa hänen opetustaitojaan.

Vanhat ohjaajat voivat antaa hyödyllistä palautetta ohjaajien koulutuksesta [3]. Ohjaajat voivat suositella opettamiaan opiskelijoita uusiksi ohjaajiksi ja ehdottaa, millaisilla valintaperusteilla saadaan valittua parhaat ohjaajaehdokkaat.

3.4 Käytännön organisaatio

Ohjaajia hyödyntävien kurssien henkilökunnan rakenne vaihtelee kurssittain ja yliopistittain. Esimerkiksi pienessä yliopistossa voi toimia malli, jossa luennoijan lisäksi kurssin henkilökuntaan kuuluu vain muutama ohjaaja [4], kun taas suurissa laitoksissa kursseilla, joilla on paljon opiskelijoita, on hyödyllistä, että kurssin henkilökuntaan kuuluu ohjaajien työstä vastaava assistentti [16]. 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 kursseille paljon, ohjaajien käyttäminen tarvitsee usein toimiakseen jonkinlaisen koordinaattorin. Koordinaattori voi olla esimerkiksi opinnoissaan pitkälle edennyt opiskelija, joka on aiemmin toiminut ryhmänohjaajana. Jos ohjaajia on paljon, voidaan tarvita useampia koordinaattoreita [20].

Koordinaattorin tehtäviin kuuluu kurssin käytännön hallinnointi, eli esimerkiksi luokkahuoneiden varaaminen ja tarvittavien ohjaajien jakaminen kursseille. He myös hoitavat ohjaajien valinnan, palkkaamisen ja kouluttamisen [17, 20]. Koordinaattori voi huomauttaa ohjaajan puutteellisesta toiminnasta [17].

Koordinaattorien kuuluu ylläpitää tiedon kulkua ja kommunikointia henkilökunnan välillä. Heidän tulee myös järjestää tarvittavat kommunikointikanavat opiskelijoiden ja ohjaajien välille [17]. 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 paremmin. Palaverissa ohjaajat pääsevät tapaamaan toisiaan ja jakamaan tietoa. Koordinaattorien kuuluu myös järjestää tapaamisia, joissa käydään läpi opetukseen liittyviä asioita [17, 20]. Koordinaattori voi kerätä opiskelijoilta palautetta ohjaus-tilaisuuksista, ja esitellä palautteen ohjaajille henkilökunnan tapaamisessa [15]. Palautteen avulla koordinaattori voi suunnitella parannuksia seuraavalle lukukaudelle.

4 Seurauksia

4.1 Hyötyjä

Opiskelijoiden käyttämisestä ohjaajina on taloudellista etua yliopistolle, sillä heille maksettava palkka on yleensä matalampi kuin kokeneemmille assistenteille, tai heidät palkitaan opintopisteillä [17].

Yliopistolla työskentelevät opiskelijat voivat toimia roolimalleina muille opiskelijoille [20, 24], sillä opiskelijoiden voi olla vaikea nähdä itsensä esimerkiksi kokeneena luennoijana, kun taas ohjaaja tarjoaa helpommin lähestyttävämmän roolimallin [22]. Jos ohjaajat valitaan opiskelijoiden keskuudesta, voidaan henkilökunta saada paremmin vastaamaan opiskelijoita. Etenkin naisille ja vähemmistöjen edustajille voi olla tärkeää saada kaltaisansa roolimalli, jotta heidän on helpompi tuntea kuuluvansa joukkoon [11]. Ryhmänohjaajien käyttäminen mahdollistaa paremmin pienryhmissä opiskelun, joka auttaa opiskelijoita tutustumaan toisiinsa. Yhteenkuuluvuuden tunne saa opiskelijat pysymään alalla paremmin.

Ryhmänohjaajien käyttäminen voi toimia koulutuksena ja innoituksena uusille luennoijoille yliopiston tulevaisuudessa [20, 11]. Ryhmänohjaajia ohjaavat opettajat saavat tilaisuuden vaikuttaa ohjaajien näkemykseen opetusalaista [11].

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 [4]. Tehtävien suuri määrä vaikuttaa myös oppimiseen, sillä opiskelija oppii materiaalin sitä paremmin, mitä enemmän hän pääsee harjoittelemaan sitä käytännön tehtävissä [26]. Jonkun tietyn henkilökunnan edustajan mahdollinen puolueellisuus ei välttämättä pääse vaikuttamaan opiskelijoihin, jos luokassa on samaan aikaan useita erilaisia ohjaajia [11].

Kurssin ilmapiiriä rentouttaa, jos opiskelijat näkevät luennoijan ja ohjaajan tulevan hyvin toimeen keskenään. 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. Ohjaaja voi olla jonkin tietojenkäsittelyn osa-alueen erityisosaaja ja tuoda osaamisensa kurssin hyödyksi. Motivoitunut ohjaaja motivoi opetettaviakin työskentelemään ahkerammin [4]. Kurssin opiskelijoiden voi olla helpompi antaa palautetta ohjaajalle kuin luennoijalle [11]. Ohjaajat pystyvät antamaan hyödyllistä palautetta kurssin toiminnasta ja antaa parannusehdotuksia [3].

Ohjaajat saavat palkkaa tai opintopisteitä, minkä lisäksi opettaminen parantaa heidän ryhmätyö- ja esiintymistaitojaan. Ohjaajat oppivat keskustelemaan tietokonetaidoiltaan eritasoisten opiskelijoiden kanssa [26]. Opettaessaan he oppivat itsekin materiaalin paremmin [16], ja opetettavien kannustaminen hyvään ohjelmointityyliin parantaa ohjaajankin ohjelmointityyliä [20]. Ohjaajana toimiminen opettaa vastuukantoa ja liittyy opiskelijan paremmin mukaan yliopiston yhteisöön [4]. Ohjaajat voivat saada toisistaan seuraa ja osallistua keskenään erilaisiin vapaa-ajan aktiviteetteihin [20].

Kun luennoija esittelee kurssin sisällön ohjaajille, hän saa tilaisuuden käydä opetettava materiaali läpi ennen sen esittämistä luennolla [8].

4.2 Haasteita

On pelätty, että uusien ja vähän koulutettujen opiskelijoiden käyttäminen ohjaajina ei olisi hyödyllistä opetettaville [5]. Ohjaajien puutteellinen kokemus tai koulutus opetuksesta voi vaikeuttaa ohjausta tai epäonnistumisen pelkoa ohjaajille [10]. Aiheesta kirjoitetut artikkelit kuitenkin väittävät, että hyvän ohjaajan ei tarvitse olla opinnoissaan kovin paljoa edellä opetettaviaan.

Vaikka ohjaajat pääasiassa hyötyvät siitä, että he ovat suurin piirtein saman ikäisiä kuin opetettavansa, voi siitä seurata myös ongelmia tehtävien arvostelun suhteen. Koska ohjaajalla ei mahdollisesti ole paljon kokemusta tai auktoriteettia, voi hänen olla vaikea käsitellä opiskelijoita, jotka ovat mielestään saaneet liian huonon arvosanan. Ohjaajilla pitäisikin olla selvät ohjeet tehtävien pisteyttämisen suhteen kiistatilanteiden varalta [20]. Toisaalta jos pidetään tärkeänä sitä, että opiskelijat mieltävät ohjaajat vertaisekseen, voi olla parempi, että ohjaajat eivät osallistu tehtävien pisteyttämiseen ja opiskelijoiden arvosteluun [11].

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 [17].

Jatkuva vaihtelu opettajan ja opiskelijan roolin välillä voi olla rasittavaa ohjaajalle [10].

Lakisääteiset syyt voivat rajoittaa opiskelijoiden hyödyntämistä opetuksessa. Säädökset voivat esimerkiksi kieltää opiskelijoita arvostelemasta saman asteen opiskelijoiden tehtäviä varmistaakseen, että yliopistot eivät laiminlyö velvollisuuksiaan arvostelussa. Säädökset voivat myös kieltää liian pienet kurssit. Vaikka jollakin tietojenkäsittelytieteen alkeiskurssilla olisi valtavat

määrät opiskelijoita, jos heidät on jaettu pieniin ryhmiin, voi yksi iso kurssi tilastollisesti näyttää monelta pieneltä [16].

Koulutuksen ja kokemuksen puutteen takia ympäristön häiriöt voivat haitata ohjaajan työskentelyä koenutta opettajaa enemmän [13]. Esimerkiksi luokahuoneen valaistus, pöytien järjestely ja opetustilanteen kesto vaikuttavat ohjaajan työhön.

5 Yhteenveto

Useissa yliopistoissa on todettu, että opiskelijoiden käyttö opetuksessa nostaa opetuksen laatua ja mahdollistaa suurempien opiskelijamäärien opettamisen ilman, että kustannukset nousevat. Ohjaajana toimiminen antaa opiskelijalle hyödyllisiä taitoja ja kokemusta, ja ohjaajien läsnäolo kursseilla rentouttaa kurssin ilmapiiriä. Mallin on havaittu toimivan niin suurissa kuin pienissäkin oppilaitoksissa.

6 Lähteet

- [1] 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>.
- [2] 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>.
- [3] 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>.
- [4] 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>.
- [5] 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>.

- [6] 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>.
- [7] 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>.
- [8] 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>.
- [9] 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>.
- [10] 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.
- [11] 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.
- [12] 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>.
- [13] 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>.
- [14] Patitsas, Elizabeth ja Patrice Belleville: *What can we learn from quantitative teaching assistant evaluations?* Teoksessa *Proceedings of the Seventeenth Western Canadian Conference on Compu-*

ting 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>.

- [15] 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>.
- [16] 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>.
- [17] 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>.
- [18] 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.
- [19] 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>.
- [20] 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>.
- [21] 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>.
- [22] 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>.
- [23] 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>.

- [24] 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>.
- [25] Vihavainen, Arto, Thomas Vikberg, Matti Luukkainen ja Jaakko Kurhila: *Massive Increase in Eager TAs: Experiences from Extreme Apprenticeship-based CS1*.
- [26] Vikberg, Thomas, Arto Vihavainen, Matti Luukkainen ja Jaakko Kurhila: *Early Start in Software Coaching*.