



Tuloksia ja kokemuksia ohjelmasuorituksen visualisoinnista

Erkki Kaila, Teemu Rajala, Mikko-Jussi Laakso & Tapio Salakoski
Turun Yliopisto, Informaatioteknologian laitos

&

TUCS – Turku Centre for Computer Science

{ertaka,temira,milaak,sala}@utu.fi

Tiivistelmä

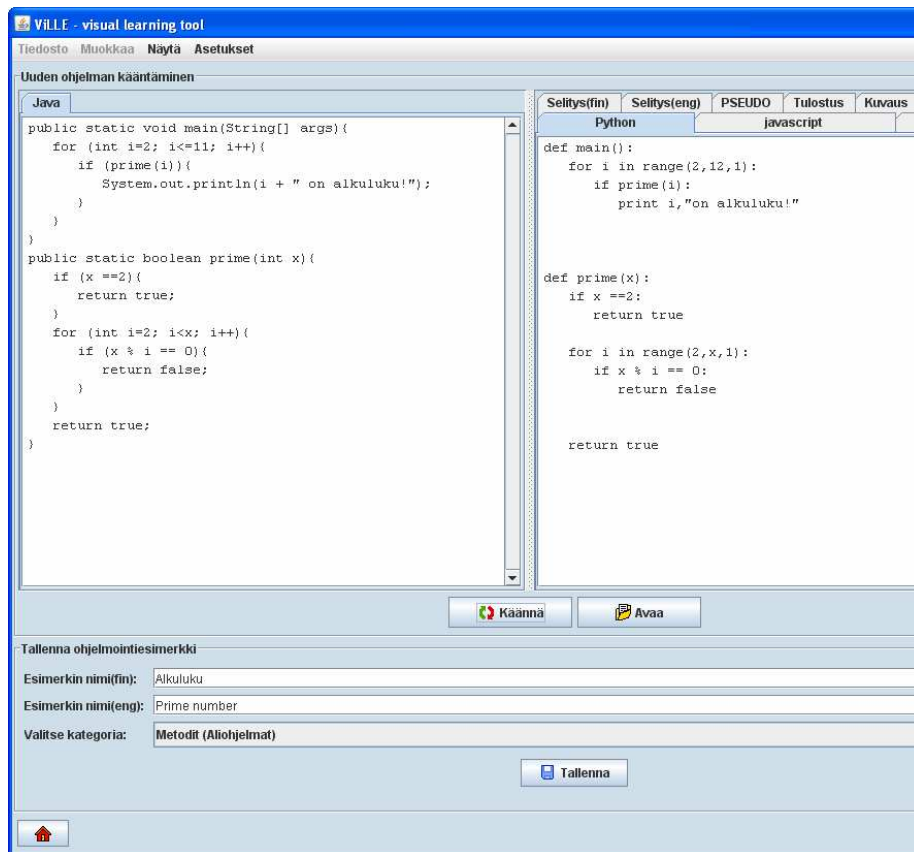
Ohjelmasuorituksen visualisointi on potentiaalisesti käyttökelpoinen menetelmä ohjelmoinnin perusteiden opettamiseksi aloittelijoille. Sen vaikutuksia on kuitenkin tutkittu melko vähän. ViLLE on Turun yliopistossa kehitetty ohjelmasuorituksen visualisointityökalu. Tässä artikkelissa esittelemme järjestelmän vaikutuksista tehtyjä tutkimuksia tuloksineen. Lisäksi esittelemme uusina kvalitatiivisina tuloksina opiskelijoiden palautetta työkalun käyttöön liittyen. Tulokset ja palaute osoittavat, että ViLLEä voidaan hyödyntää tehokkaasti ohjelmoinnin alkeiden opettamiseen aloitteleville ohjelmoijille.

1 Johdanto

Ohjelmointitaitoa voidaan pitää eräänä tietojenkäsittelytieteen opintojen tärkeimpänä yksittäisenä päämääränä — linjasta tai opintojen suuntauksesta riippumatta. Useiden tutkimusten (ks. esim. [8] tai [1]) mukaan opiskelijoilla on kuitenkin huomattavia vaikeuksia peruskäsitteiden omaksumisessa. Niukkojen opettajaresurssien ja suurten ryhmäkokojen vuoksi henkilökohtaisen ohjauksen mahdollisuudet ovat yleensä vähäiset. Tästä johdetaan tarve erilaisille opetuksen apuvälineille — ja erityisesti sellaisille apuvälineille, joita opiskelijat voivat käyttää itsenäisesti oppimisen apuna — on ilmeinen.

Visualisoinnilla tarkoitetaan ohjelman tai algoritmin esittämistä graafisesti. Wig-

ginsin [11] mukaan visualisoinnin tarkoituksena on auttaa käyttäjää ymmärtämään mitä ohjelma tekee, miksi se tekee niin, miten se toimii ja mitä suorituksesta seuraa. Visualisointijärjestelmien päämääränä on siis konkretisoida ohjelmien ja algoritmien suorituksen muuten abstrakteja piirteitä. Oletettavasti järjestelmien avulla voidaan helpottaa ohjelmien ymmärtämistä ja näin ollen edelleen parantaa oppimistuloksia. Algoritmeja visualisoivien järjestelmien käyttöä opetuksen apuna onkin tutkittu kohtuullisen laajasti (ks. esim. [2] tai [4]); ohjelmasuorituksen visualisoinnissa tilanne on toinen: vaikka erilaisia järjestelmiä on kehitetty suhteellisen paljon, löytyy tutkimuksia oppimisvaikutuksista niukasti.



Kuva 1: ViLLEn esimerkkieditori.

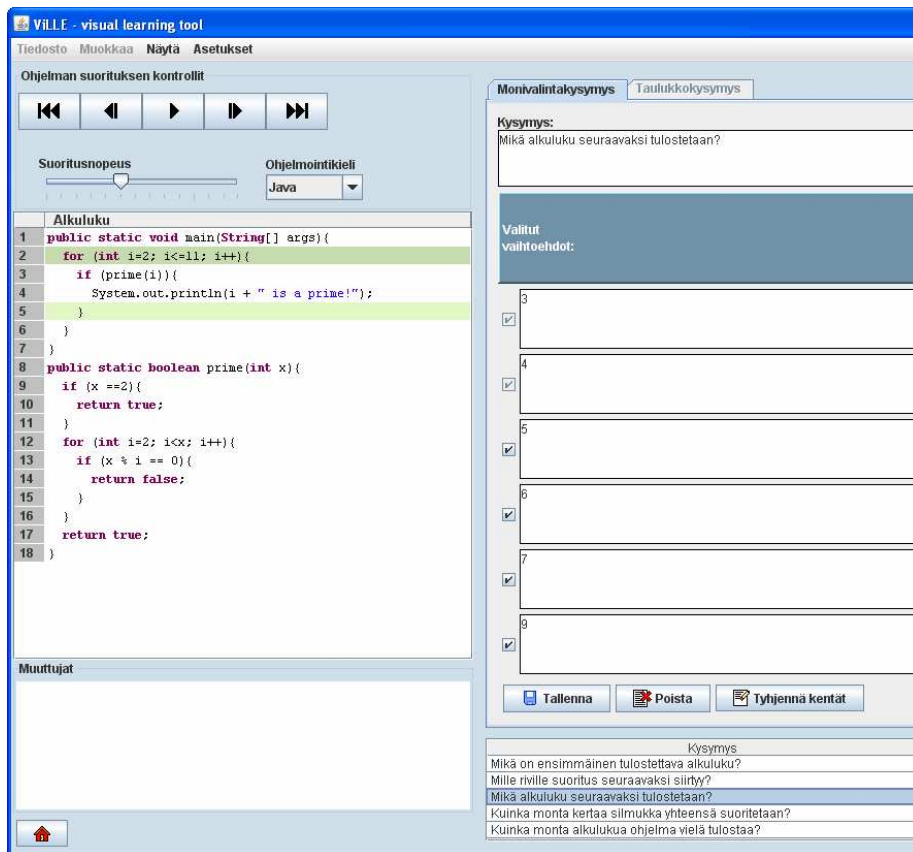
Tieteellisesti todistettujen oppimistulosten saavuttaminen oli eräs lähtökohdista kehittäessämme ViLLE-nimistä ohjelmasuorituksen visualisointijärjestelmää Turun Yliopistossa. Järjestelmää on pyritty testaamaan mahdollisimman monipuolisesti, opiskelijoiden erilaiset lähtökohdat huomioiden. Tähän artikkeliin on koottu tähän mennessä julkaistuja tutkimuksia järjestelmän käytöstä ohjelmoinnin opetuksessa tuloksineen, minkä lisäksi esittelemme uusia tuloksia opiskelijoilta kerätyn palautteen pohjalta. Lisäksi esittelemme järjestelmän yleiset ominaisuudet

suhteellisen kattavasti, niin opettajan kuin oppijan kannalta, ja pohdimme järjestelmän tulevaisuutta. Artikkelin on julkaistu englanniksi käännettynä lehdessä "Informatics in Education" [3].

2 ViLLE

2.1 Ominaisuudet yleisesti

ViLLE on Turun Yliopistossa kehitetty ohjelmasuorituksen visualisointijärjestelmä. Sen tarkoituksena on havainnollistaa ohjelman tiloissa sen suorituksen aikana tapahtuvat muutokset erilaisten graafis-



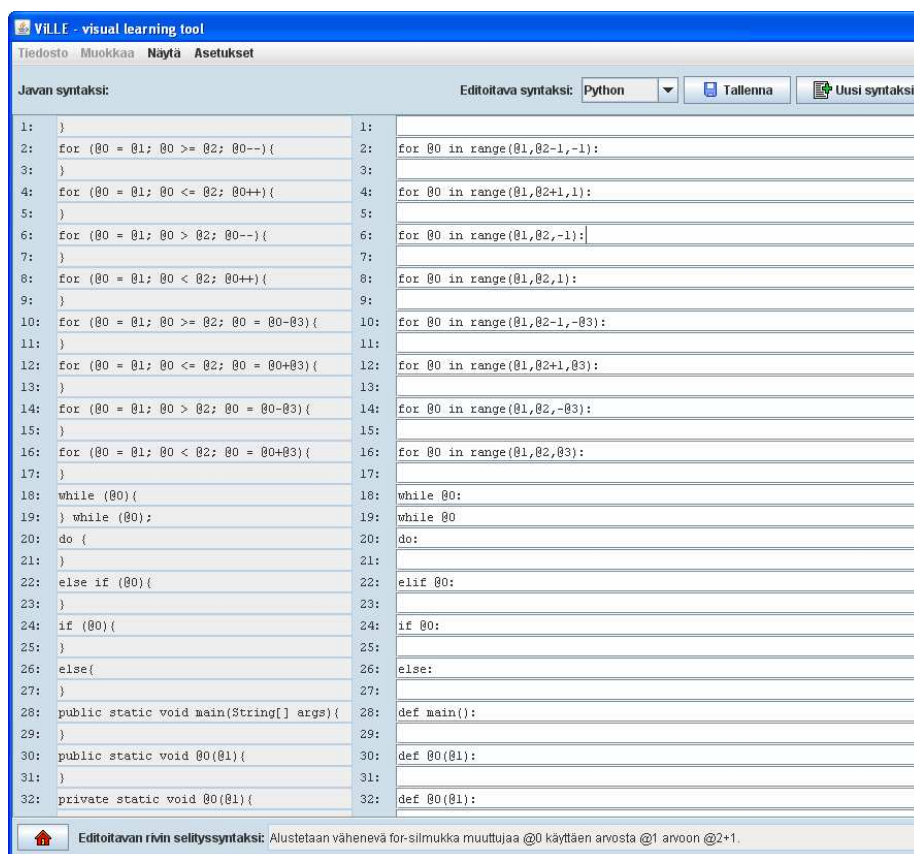
Kuva 2: Kysymyseditori.

ten ja tekstimuotoisten elementtien avulla. Järjestelmä tukee joustavasti eri ohjelmointikieliä ja mahdollistaa uusien syntaksien ja esimerkkien määrittelyn sisäänrakennettujen editorien avulla. Opettaja voi lisäksi muodostaa esimerkkeihin liittyen harjoitustehtäviä. Esimerkit ja harjoitukset voidaan järjestelmän viennitoimintona hyväksikäyttäen tarjota helposti opiskelijoiden saataville esimerkiksi verkkoon tai muistitikulle. Integroimalla harjoitukset TRAKLA2-palvelimeen [7] voidaan automaattisesti tarkastettavat tehtävät ottaa osaksi ohjelmointikurs-

sia, esimerkiksi korvaamaan (tai täydentämään) perinteiset demonstraatiot tai laskeharjoitukset. Lisätietoa ViLLEstä löytyy järjestelmän kotisivuilta osoitteesta <http://ville.cs.utu.fi>.

2.2 Ominaisuudet opettajan kannalta

Opettajan näkökulmasta ViLLEn tärkeimpänä ominaisuutena voidaan pitää sen soveltuvuutta erilaisiin kursseihin: käytettävä ohjelmointikieli (tai ohjelmointikielet), esimerkit ja tehtävät ovat kaikki



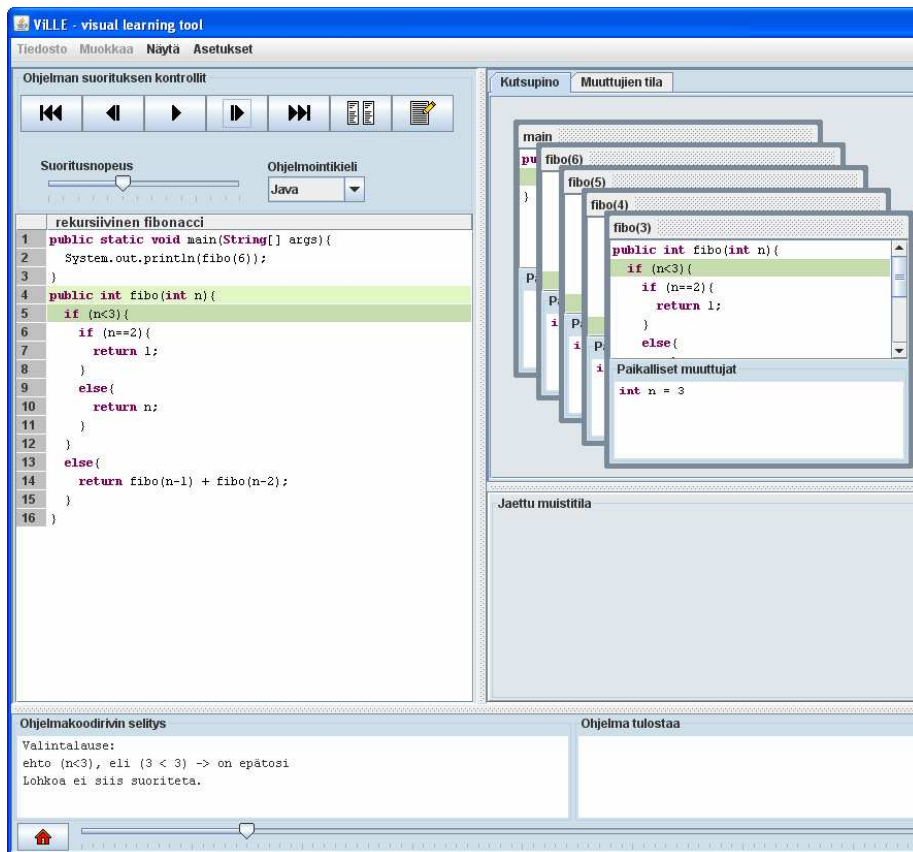
Kuva 3: ViLLEn syntaksieditori.

muokattavissa sisäänrakennettujen editoreiden avulla. Tarkoituksena on sekä mahdollistaa helppo integrointi mille tahansa ohjelmointikurssille että tarjota työkalu, jolla opettaja voi toteuttaa joustavasti omaa opetusfilosofiaansa tarvitsematta sitoutua esimerkiksi määrättyyn ohjelmointikielen tai valmiisiin esimerkkeihin tai tehtäviin.

2.2.1 Esimerkkien muokkaaminen

Esimerkit on jaettu aiheittain kategorioihin. Opettaja voi sekä muokata valmiita esimerkkejä että luoda kokonaan uusia

esimerkkejä esimerkkieditorin (kuva 1) avulla. ViLLE kääntää Javalla kirjoitetun ohjelmakoodin automaattisesti kaikille määritellyille kielille ja muodostaa jokaisesta rivistä selitykset sekä visualisoinnin kannalta tarpeelliset suoritus-tapahtumamäärittelyt. Työkalu tukee Javan määrittelyistä peruskurssien kannalta olennaisimpia ominaisuuksia: painopiste on niissä ominaisuuksissa, jotka pystytään havainnollisesti visualisoimaan, ja jotka esiintyvät yleisesti peruskursseilla käytetyissä ohjelmointikielissä; näin ollen esimerkiksi oliotuki on rajallinen.



Kuva 4: ViLLEn visualisointinäkömä.

2.2.2 Kysymykset ohjelman suorituksesta

Opettaja voi määrittellä kysymyksiä ohjelman suorituksesta sitouttaakseen opiskelijat seuraamaan visualisointia tiiviimmin. Nykyisellään järjestelmä tukee monivalintakysymyksiä ja graafisia taulukkokysymyksiä. Kysymykset määritellään kysymyseditorin (kuva 2) avulla.

Kysymykset liitetään esimerkkiohjelman suoritukseen ja ne esitetään opiskelijalle automaattisesti tämän suoritettua ohjelman määrittäytyn pisteeseen asti. TRAKLA-palvelinta käyttämällä voidaan

lisäksi pitää automaattisesti kirjaa opiskelijan läpikäymistä tehtävistä ja oikeiden vastausten määrästä (ks. aliluku 2.4).

2.2.3 Ohjelmointikielten määrittely

Ohjelmointikielten syntakseja voidaan määrittellä ViLLEn syntaksieditorin (kuva 3) avulla. Määriteltävä syntaksi on esitetty editorin oikeassa ikkunassa ja vastaavat rivit Javan syntaksissa vasemmassa. Lisäksi alhaalla olevassa ikkunassa on esitetty editoitavan rivin syntaksin selitys.

Rivivastaavuus Java-kielen on kanssa on pakollinen kaikissa määritellyissä syn-

takseissa, jotta ohjelman suoritus ja määritellyt kysymykset sijoittuvat oikeille paikoilleen valitusta kielestä riippumatta. Rivien määrittely ei kuitenkaan ole pakollista: näin esimerkiksi Pythonin lohkot voivat loppua tyhjään riviin Javan vastavien lohkojen päättyessä sulkevaan aaltosulkeeseen.

2.3 Ominaisuudet opiskelijan kannalta

ViLLEn avainominaisuudet opiskelijan kannalta voidaan jakaa seuraaviin kategorioihin:

Ohjelman suorituksen visualisointi: ViLLE visualisoi esimerkkihjelman suorituksen rivi kerrallaan (kuva 4). Ohjelmavien korostuksen lisäksi järjestelmä esittää omilla alueillaan mm. tiedon muuttujien arvoista ja näkyvyydestä, selityksen suoritettavasta rivistä ja ohjelman tulosteet. Aliohjelmat ja niiden paluuarvot (samoin kuin aliohjelmille kuuluvat paikalliset muuttujat) esitetään kutsujärjestyksessä omissa kehyksissään erillisessä kutupinossa. Lisäksi järjestelmä esittää halluissa globaalit muuttujat (käytännössä taulukot) arvoineen omissa kehyksessään.

Kieliriippumattomuus: ViLLE visualisoi ohjelmat samalla tavalla ohjelmointikielestä riippumatta. Opiskelija voi vaihtaa käytettävää ohjelmointikieltä missä tahansa suorituksen vaiheessa. Lisäksi järjestelmässä on erillinen rinnakkaisnäkyvä (kuva 5), jossa saman esimerkkihjelman suoritus esitetään yhtäaikaaisesti kahdella valinnaisella ohjelmointikielellä.

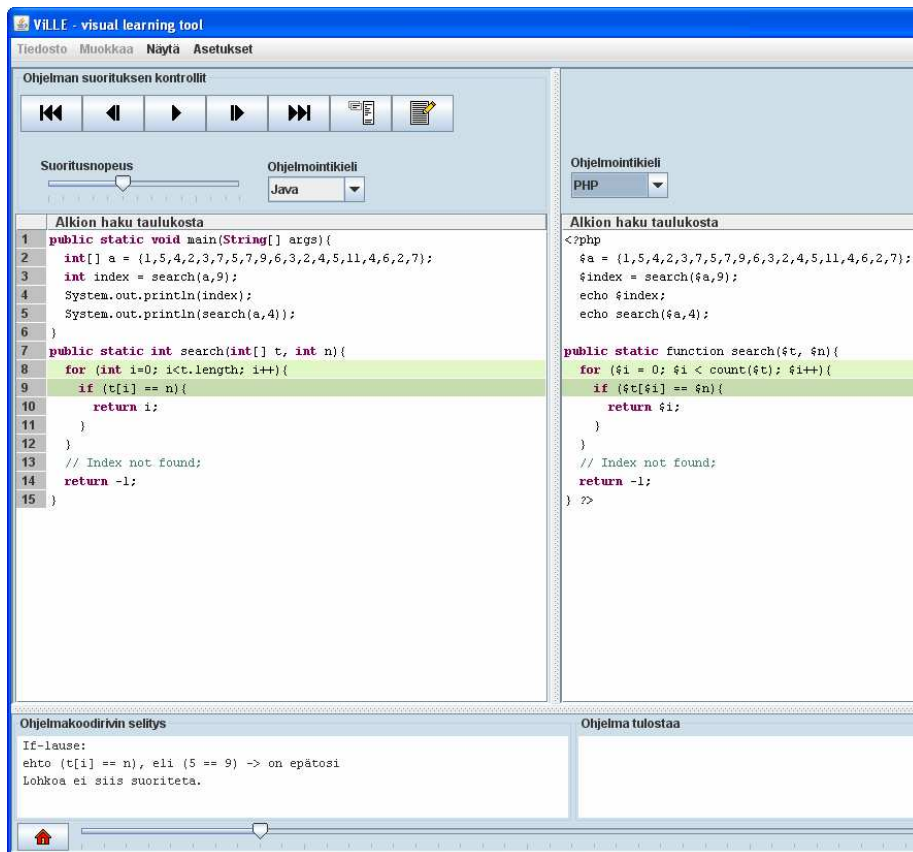
Visualisoinnin kontrollit: Opiskelija voi kontrolloida suorituksen etenemistä monipuolisesti: suorituksessa on mah-

dollista liikkua askel kerrallaan eteen- tai taaksepäin, minkä lisäksi ohjelma (tai osa siitä) voidaan suorittaa automaattisesti jatkuvana halutulla nopeudella. Ikkunan alalaidassa olevalla liukusäätimellä opiskelija voi lisäksi halutessaan siirtyä suorituksessa mihin tahansa vaiheeseen nopeasti. Liukusäätimellä on myös sekundäärinen funktio: sen askelista voidaan nopeasti päätellä ohjelman suorituksen kuuluvien askelten määrä, ja sopivilla esimerkeillä vertailla algoritmien (esimerkiksi kaksi erilaista järjestelyalgoritmia) tehokkuutta vastaavilla syötteillä.

Vuorovaikutus: Kysymyksiin vastaamisen (kuva 6) lisäksi opiskelijat pystyvät opettajan niin halutessa muokkaamaan esimerkin koodia visualisointitilassa. Muutokset ohjelmakoodissa voidaan visualisoida välittömästi kääntämisen jälkeen. Koska ohjelmien muokkaus on kuitenkin nykyisessä versiossa tehtävä aina esimerkkihjelman Java-kieliseen versioon, ei ominaisuus luonnollisesti sovellu kaikkien kurssien käyttöön.

2.4 Automaattisesti arvioidut tehtävät

TRAKLA-palvelinta käyttämällä ViLLEn tehtävät voidaan siirtää verkossa suoritettavaksi kokonaisuudeksi (kuva 7). Palvelin huolehtii opiskelijoiden kirjautumisista, pistemäärien ylläpidosta ja tehtäväkierrosten avautumisesta ja sulkeutumisesta haluttuina ajankohtina. Opiskelijat voivat suorittaa yksittäiset tehtävät niin monta kertaa kuin haluavat. Lisäksi opettaja voi asettaa kierrokselle minimipistemäärän, joka opiskelijoiden on vähintään saavutettava kurssin läpäistäkseen. Verkossa suoritettavat ViLLE-tehtävät ovat nykyisellään käytössä useissa korkeakouluissa Suomessa, ja palaute (ks. aliluku



Kuva 5: ViLLE:n rinnakkaisnäkyä.

3.4) opiskelijoilta on ollut pääsääntöisesti erinomaista: järjestelmä näyttää täyttävän paikkansa ohjelmakoodin lukutaidon parantamisessa erityisesti ohjelmoinnin alkutaipaleella.

3 Tutkimuksia ViLLE:n käytöstä

3.1 Järjestelmän tehokkuus opetuskäytössä

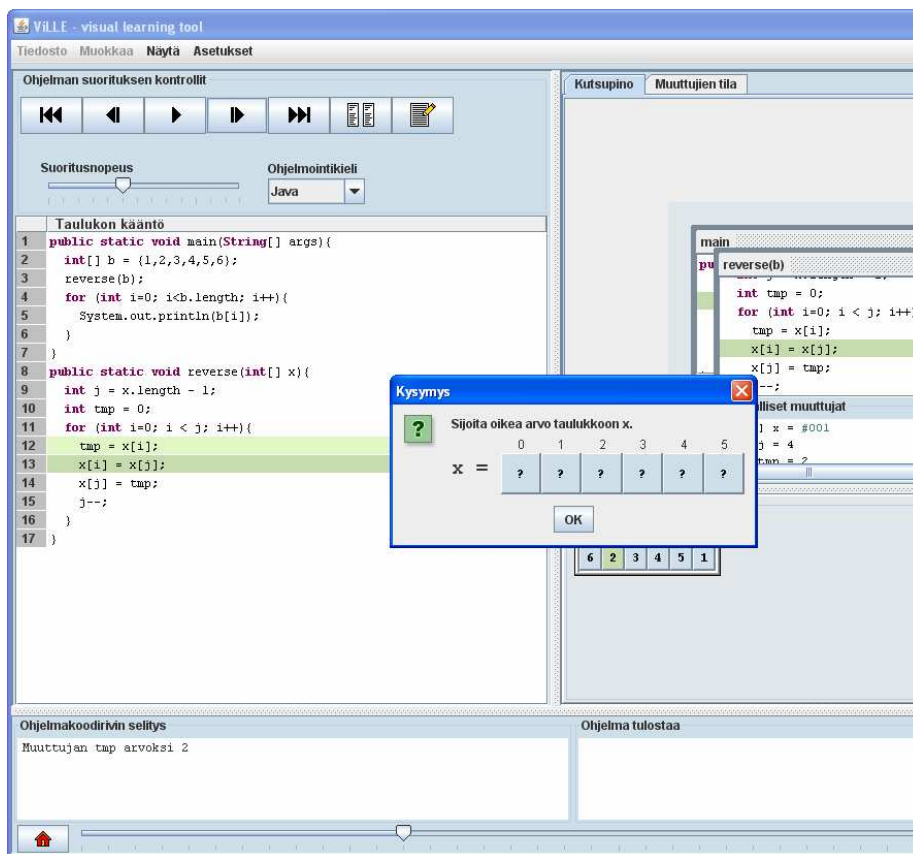
ViLLE:n tehokkuutta opetuskäytössä tutkittiin Turun Yliopiston ”Johdatus informaatioteknologiaan” -kurssilla. Tutkimukseen osallistui 72 opiskelijaa (N=72), ja siinä haettiin vastausta kahteen tutkimusongelmaan: 1) ”Onko ViLLEstä apua ohjelmoinnin peruskäsitteiden oppimisessa?” ja 2) ”Onko oppimisessä eroa, mi-

Taulukko 1: Esitestin tulokset.

<i>Kysymys</i>	<i>Vertailuryhmä (N = 40)</i>	<i>Koeryhmä (N = 32)</i>	<i>p-arvo</i>
Kysymys 1 (K1)	5,20 (2,67)	6,19 (2,46)	0,111
Kysymys 2 (K2)	2,70 (3,53)	2,13 (3,53)	0,494
Kysymys 3 (K3)	2,68 (4,15)	2,09 (3,88)	0,546
Yhteensä	10,58 (8,64)	10,41 (7,18)	0,930

Taulukko 2: Jälkitestin tulokset.

<i>Kysymys</i>	<i>Vertailuryhmä (N = 40)</i>	<i>Koeryhmä (N = 32)</i>	<i>p-arvo</i>
JK1 (K1)	6,30 (2,81)	6,13 (2,69)	0,790
JK2 (K2)	5,10 (4,35)	5,50 (4,50)	0,704
JK3	6,28 (3,75)	5,88 (3,75)	0,654
JK4 (K3)	6,15 (4,56)	6,50 (4,42)	0,744
JK5	7,05 (3,78)	6,69 (4,08)	0,698
Yhteensä (yhteiset)	17,55 (9,08)	18,13 (8,81)	0,788
Yhteensä (kaikki)	30,88 (15,20)	30,69 (15,08)	0,959
Erotus JK1–K1	1,10 (2,60)	-0,06 (2,81)	0,073
Erotus JK2–K2	2,40 (3,30)	3,38 (4,02)	0,262
Erotus JK4–K3	3,48 (4,81)	4,41 (4,53)	0,405
Erotus yhteensä	6,98 (6,81)	7,72 (6,76)	0,646



Kuva 6: Taulukkokysymys ViLLEn visualisointinäkyvässä.

käli aikaisempi ohjelmointikokemus otetaan huomioon?”. Vastaavat nollahypoteesit olivat, että ViLLEstä ei ole hyötyä ohjelmoinnin oppimisessa, ja että vaikutus on sama aloittelijoille ja kokeneemmille ohjelmoijille. Tutkimus järjestettiin kurssin kolmannella viikolla, jolloin varsinaista ohjelmointiasiaa ei oltu vielä käsitelty. Opiskelijoille järjestettiin kuitenkin erillinen perehdytysluento, jonka aikana muun muassa esiteltiin käytetty ohjelmointikieli. Lisäksi kurssin toisella viikolla kurssin kotisivuille lisättiin linkki ViLLEen, jotta järjestelmä ja sen käyttö olisi opiskeli-

joille tuttua ennen tutkimusta. Tutkimusta varten opiskelijat jaettiin satunnaisesti kahteen eri ryhmään: koeryhmään kuului 32 oppilasta (N=32) ja vertailuryhmään 40 (N=40). Molemmat ryhmät aloittivat kahden tunnin tietokoneluokkaharjoituksen suorittamalla esitestin viidentoista minuutin aikana. Testin kolmessa kysymyksessä kussakin opiskelijoille esitettiin ohjelmakoodilohko, ja pyydettiin määrittelemään ohjelman tulostus. Esitestin jälkeen molemmat ryhmät opiskelivat testissä esitettyjä aiheita käymällä vapaasti läpi html-muotoisen ohjelmointitutoriaalin.

1	Muuttujat ja ehtolauseet Sulkeutuu 31 joul. 2009 00:00:00 GMT+0200	138/650 Pistettä	6 Palautusta
2	Merkkijonot Sulkeutuu 31 joul. 2009 00:00:00 GMT+0200	0/250 Pistettä	0 Palautusta
3	Toistorakenteet (Pillota) Sulkeutuu 31 joul. 2009 00:00:00 GMT+0200	Pisteet	Palautukset
Esimerkit havainnollistavat ehtolauseiden toimintaa			
1:	For-lause	0 / 50	1
2:	For-lause 2	0 / 50	0
3:	While-lause	25 / 50	1
4:	While-lause 2	33 / 50	1
5:	Do-while -lause	0 / 50	1
6:	For- ja while-lauseet	50 / 50	1
7:	Toisto- ja valintarakenteet	33 / 50	1
8:	Toisto- ja valintarakenteet 2	0 / 50	1
9:	Sisäkkäiset toistorakenteet	0 / 50	0
10:	Sisäkkäiset toistorakenteet 2	0 / 50	0
11:	Break -lause	0 / 50	0
12:	continue-lause	0 / 50	0
13:	Fibonacci luku	0 / 50	0
14:	Suurin yhteinen tekijä	0 / 50	0
15:	Merkkijonon kääntö	0 / 50	0
16:	Merkkijonon kääntö 2	0 / 50	0
17:	Merkkijonon alijonot 2	0 / 50	0
		Pisteesi: 141/850	

Kuva 7: ViLLE-tehtäviä verkossa.

Koeryhmään kuuluvat opiskelijat pystyivät lisäksi suorittamaan tutoriaalissa olevat ohjelmointitehtävät ViLLEn avulla. Tutoriaalini läpikäynnin (45 minuuttia) jälkeen molemmat ryhmät suorittivat jälkitestin puolen tunnin aikana. Jälkitestin kuului esitettien kysymysten lisäksi kaksi vaativampaa tehtävää, joista ensimmäisessä opiskelijoiden piti itse muodostaa annetun tehtävän suorittava ohjelma, ja toisessa päätellä rekursiivisen algoritmin sisältävän ohjelman tuloste. Alku- ja jälkitestien tehtävät pisteytettiin nolasta kymmeneen siten, että nolla tarkoitti täysin väärää vastausta, ja jokainen piste tarkoitti 10% nousua vastauksen oikeellisuudessa. Maksimipistemäärä esitettissä oli näin ollen 30 ja jälkitestissä 50 pistettä. Esitettien pisteet molemmille ryhmille on esitetty taulukossa 1, mukaan lukien pisteiden keskiarvot, keskihajonta (suluissa) ja kaksisuuntaisen t-testin avulla lasketut p-arvot ryhmien välillä.

Yhdessäkin kysymyksessä ei esiin-

tynyt tilastollisesti merkitsevää eroa. Absoluuttisella asteikolla vertailuryhmä suoriutui paremmin kysymyksissä K2 ja K3, ja koeryhmä kysymyksessä K1.

Jälkitestin tulokset on esitetty taulukossa 2. Jokaisen kysymyksen yhteydessä on ilmoitettu suluissa vastaavan kysymyksen numero esitettissä. Yhteispisteistä on esitetty sekä jaettujen (eli kysymysten, jotka esiintyivät sekä esi- että jälkitestissä) että kaikkien pisteiden keskiarvo. Lisäksi on esitetty vastaavien kysymysten esi- ja jälkitestin pisteiden erotus.

Vertailtaessa sekä esi- että jälkitestissä esiintyneitä (l. yhteisiä) kysymyksiä huomataan, että vertailuryhmä suoriutui paremmin kysymyksessä JK1, ja koeryhmä kysymyksissä JK2 ja JK3. Joka tapauksessa erot ovat liian pienet, jotta nollahypoteesi voitaisiin hylätä.

Taulukossa 3 on vertailtu esi- ja jälkitestien tuloksia ryhmien sisällä. Kuten taulukosta nähdään, molemmat ryhmät suoriutuivat tilastollisesti merkitse-

Taulukko 3: Esi- ja jälkitestin tulokset.

<i>Pisteet</i>	<i>Vertailuryhmä</i>	<i>Koeryhmä</i>
Esitesti	10,58	10,41
Jälk testi	17,55	18,13
Erotus yhteensä	6,98	7,72
p-arvo	0,000	0,000

västi paremmin jälkitestissä kuin esitestissä (p-arvo < 0,01). Tästä voidaan päätellä, että järjestetyn kaltaisten tilaisuuksien avulla on mahdollista opiskella itsenäisesti hyvinkin tehokkaasti ohjelmoinnin peruskäsitteitä.

Toisena tutkimusongelmana selvitetiin onko aikaisemmalla ohjelmointikokemuksella vaikutusta oppimistuloksiin. Tätä varten molemmat ryhmät jaettiin tuloksia tarkastellessa edelleen kahtia: niihin, joilla ei ollut yhtään aikaisempaa ohjelmointikokemusta (EAO), ja niihin, joilla oli ennestään jonkin verran ohjelmointikokemusta (JAO). Tuloksia tarkasteltiin aluksi ryhmien sisällä (inside groups). Esitestin tulokset ryhmittäin ohjelmointikokemus huomioiden on esitetty taulukossa 4. Taulukosta nähdään, että aikaisempaa kokemusta omaavien tulokset ovat tilastollisesti merkitsevästi parempia vertailuryhmässä (p ~0,001) ja lähes tilastollisesti merkitsevästi parempia koeryhmässä (p ~0,051). Voidaan olettaa, että kokeneiden ohjelmoijien tulisi saada parempia pisteitä, joten tässä voitaisiin periaatteessa käyttää myös luottamisväliä (p < 0,1). Vastaavat tulokset jälkitestistä on esitetty taulukossa 5. Taulukosta nähdään, että vertailuryhmissä tilastollisesti merkitsevä ero noviisien ja aikaisemmin ohjelmoineiden välillä säilyi myös lopputestissä. Koeryhmässä ero on kuitenkin kaventunut siten, että tilastollinen merkitsevyys on hävinnyt (p-arvot 0,212 ja 0,151).

Koska tulos ei ole tilastollisesti vedentävä (koeryhmässä p ~0,051 esitestissä, vaadittava p < 0,05) vertailtiin tuloksia myös ryhmien yli. Yksisuuntainen ANOVA-varianssianalyysi kaikkien neljän ryhmän yli esi- ja jälkitestin yhteisten kysymysten summille antoi $F(3,68) = 8,168$ (esi) ja $F(3,68) = 3,139$ (jälki), p < 0,05. Tämän jälkeen laskettiin post-hoc-analyysin avulla homogeeniset osajoukot esi- ja jälkitestin tuloksille. Esitestissä luonnollisesti kahdeksi homogeeniseksi osajoukoksi muodostuivat ensinnäkin koeryhmän EAO ja vertailuryhmä EAO (KEAO ja VEAO), ja toiseksi koeryhmän JAO ja vertailuryhmän JAO (KJAO ja VJAO). Jälkitestissä tilanne muuttui, sillä KEAO kuului kumpaankin osajoukkoon, ja otti näin molempia JAO-ryhmiä kiinni. Näin ollen näyttää siltä, että ViLLEstä on erityisesti hyötyä kokemattomille ohjelmoijille ja tulosta voidaan pitää merkitävänä, koska järjestelmän käyttöön varattu aika oli verrattain lyhyt. Esi- ja jälkitestin kysymyksille laskettiin Cronbachin alfa-luotettavuusarvot: arvot (esitestin = 0,667 ja jälkitestin = 0,831) osoittavat, että instrumenttia voidaan pitää erittäin luotettavana. Tutkimus on esitetty kokonaisuudessaan artikkelissa [10].

Taulukko 4: Esitestin tulokset aikaisempi ohjelmointikokemus huomioiden.

Kysymys	Vertailuryhmä			Koeryhmä		
	EAO (N=23)	JAO (N=17)	p-arvo	EAO (N=20)	JAO (N=12)	p-arvo
K1	4,17 (2,33)	6,59 (2,53)	0,003	5,60 (2,11)	7,17 (2,76)	0,107
K2	1,22 (1,78)	4,71 (4,31)	0,005	1,00 (2,22)	4,00 (4,51)	0,049
K3	1,00 (2,86)	4,94 (4,62)	0,005	1,65 (3,62)	2,83 (4,34)	0,414
Yhteensä	6,39 (4,68)	16,24 (9,63)	0,001	8,25 (5,44)	14,00 (8,48)	0,051

Taulukko 5: Jälkitestin tulokset aikaisempi ohjelmointikokemus huomioiden.

Kysymys	Vertailuryhmä			Koeryhmä		
	EAO (N=23)	JAO (N=17)	p-arvo	EAO (N=20)	JAO (N=12)	p-arvo
JK1 (K1)	5,74 (2,78)	7,06 (2,75)	0,144	5,90 (2,86)	6,50 (2,43)	0,533
JK2 (K2)	3,39 (3,97)	7,41 (3,81)	0,003	4,70 (4,58)	6,83 (4,22)	0,199
JK3	5,30 (4,06)	7,59 (2,90)	0,045	5,05 (3,65)	7,25 (3,65)	0,109
JK4 (K3)	5,22 (4,83)	7,41 (3,94)	0,122	6,00 (4,71)	7,33 (3,94)	0,418
JK5	6,09 (4,09)	8,35 (2,96)	0,049	6,05 (4,20)	7,75 (3,82)	0,261
Yhteensä (yhteiset)	14,35 (8,27)	21,88 (8,51)	0,008	16,60 (9,29)	20,67 (7,64)	0,212
Yhteensä (kaikki)	25,74 (14,44)	37,82 (13,68)	0,011	27,70 (15,49)	35,67 (13,53)	0,151
Erotus JK1–K1	1,57 (2,48)	0,47 (2,70)	0,198	0,30 (2,62)	-0,67 (3,11)	0,354
Erotus JK2–K2	2,17 (3,07)	2,71 (3,65)	0,620	3,70 (4,38)	2,83 (3,46)	0,564
Erotus JK4–K3	4,22 (4,73)	2,47 (4,87)	0,261	4,35 (4,73)	4,50 (4,38)	0,929
Erotus yhteensä	7,96 (5,80)	5,65 (7,98)	0,295	8,35 (7,98)	6,67 (4,14)	0,439

3.2 Vuorovaikutuksen merkitys oppimisvaikutuksiin

Naps ym. [9] esittelivät vuorovaikutuksen taksonomian, jonka tarkoitus on jäsentää käyttäjän sitoutumisen määrää visualisointijärjestelmään. Taksonomia koostuu kuudesta tasosta:

1. **Ei-katsominen:** Ei visualisointijärjestelmää käytössä
2. **Katsominen:** Käyttäjä seuraa passiivisesti visualisointia. Nimestään huolimatta tasoon kuuluvaksi lasketaan kaikki havainnointi katsomisen lisäksi. Käyttäjä voi kontrolloida visualisoinnin kulkua, muttei osallistu muuten aktiivisesti.
3. **Vastaaminen:** Käyttäjä vastaa visualisoinnista esitettyihin kysymyksiin.
4. **Muokkaaminen:** Käyttäjä muuttaa visualisointia, esimerkiksi muokkaamalla visualisoitavaa ohjelmakoodia tai algoritmia.
5. **Rakentaminen:** Käyttäjä osallistuu aktiivisesti visualisoinnin muodostamiseen, esimerkiksi kirjoittamalla ohjelmakoodin.
6. **Esittäminen:** Käyttäjä esittää visualisoinnin muiden arvioitavaksi ja evaluoi sitä yhdessä ryhmän ja opettajan kanssa.

Luvussa 3.1 esitettyä tutkimusta laajennettiin siten, että esitettyjen ryhmien lisäksi mukaan otettiin kolmas ryhmä, joka tutoriaalini lisäksi käytti ViLLEä koe-ryhmää alemmalla taksonomian tasolla. Käytännössä mukana oli siis kolme ryhmää: ei-katsominen ($N = 40$), katsominen ($N=65$) ja vastaaminen ($N = 32$). Katsomisryhmän käyttämästä ViLLE:n versiosta

oli poistettu kokonaan kysymykset ohjelman suorituksesta. Tarkoituksena oli selvittää, saadaanko tutkimuksen avulla vahvistettua Napsin ym. [9] hypoteesi, jonka mukaan visualisointijärjestelmän käytöllä voidaan saavuttaa oppimisvaikutuksia vain jos sitä käytetään vuorovaikutuksen tasoilla kolmesta ylöspäin. Esi- ja jälkitestit ja tutkimuksen kulku olivat samansisältöiset kuin luvussa 3.1.

Yksisuuntaisen ANOVA-testin avulla lasketut ryhmien väliset tilastolliset erot esi- ja jälkitestissä on esitetty taulukossa 6. Kuten taulukosta nähdään — ja luvussa 3.1 jo todettiin — ei tilastollisia eroja ryhmien välillä esiintynyt. Myös katsomisryhmän opiskelijat paransivat tuloksiaan tilastollisesti merkitsevästi tilaisuuden aikana (p -arvo esi- ja jälkitestin tuloksia verratessa ryhmän sisällä $< 0,01$). Tarkastellaan seuraavaksi noviisien ja ohjelmointikokemusta omaavien opiskelijoiden oppimistulosten eroja kaikkien ryhmien sisällä. Tulokset on esitetty taulukossa 7. Esitestissä ei-katsomis- ja katsomisryhmien sisällä noviisien ja aikaisemmin ohjelmoineiden kesken oli tilastollisesti merkitsevä ero ja vastaamisryhmässä lähes tilastollisesti merkitsevä ero. Kuten edellisessä luvussa huomattiin, vastaamisryhmältä ero oli hävinnyt lopputestissä. Katsomisryhmällä ero kuitenkin säilyi (p -arvo $< 0,001$). Tästä voidaan päätellä, että ViLLEstä hyötyäkseen noviisien on käytettävä sitä katsomistasoa korkeammalla vuorovaikutuksen tasolla. Edellisen varmistamiseksi suoritettiin yksisuuntainen ANOVA-varianssianalyysi jälkimittauksiin. Analyysin mukaan esitestin pisteiden perusteella muodostuu kaksi homogeenista osajoukkoa siten, että kaikki noviisit kuuluvat eri joukkoon kuin kokeenemat (taulukko 8). Lopputestin tuloksia vastaavalla tavalla analysoitaessa vastaamisryhmän noviisit saivat kuitenkin ai-

Taulukko 6: Tilastolliset erot ryhmien välillä.

	Ei- katsominen (N=40)	Katsominen (N=65)	Vastaaminen (N=32)	p-arvo
Esitesti yhteensä	10,58 (8,64)	10,85 (8,89)	10,41 (7,18)	0,968
Jälkitesti yhteensä (yhteiset)	17,55 (9,08)	17,94 (9,53)	18,13 (8,81)	0,963
Erotus yhteensä	6,97 (6,81)	7,09 (6,63)	7,72 (6,76)	0,881

Taulukko 7: Esi- ja jälkitestien tulokset aikaisempi ohjelmointikokemus huomioiden.

Kysymys	<i>Ei- katsominen</i>			<i>Katsominen</i>			<i>Vastaaminen</i>		
	EAO (N=23)	JAO (N=17)	p- arvo	EAO (N=36)	JAO (N=29)	p- arvo	EAO (N=20)	JAO (N=12)	p- arvo
Esitesti yhteensä	6,39	16,24	0,001	6,81	15,86	0,000	8,25	14,00	0,051
Jälkitesti yhteensä (yhteiset)	14,35	21,88	0,008	13,72	23,17	0,000	16,60	20,67	0,212
Erotus yhteensä	7,96	5,65	0,320	6,92	7,31	0,812	8,35	6,67	0,439
Jälkitesti yhteensä (kaikki)	25,74	37,82	0,011	24,72	39,90	0,000	27,70	35,67	0,151

Taulukko 8: Esitestin tulokset jaettuna homogeenisiin osajoukkoihin.

Ryhmä	N	Osajoukko alfa-arvolle 0,05	
		1	2
Ei-katsominen EAO	23	6,39	
Katsominen EAO	36	6,81	
Vastaaminen EAO	20	8,25	
Vastaaminen JAO	12		14,00
Katsominen JAO	29		15,86
Ei-katsominen JAO	17		16,24

Taulukko 9: Jälkitestin tulokset jaettuna homogeenisiin osajoukkoihin.

Ryhmä	N	Osajoukko alfa-arvolle 0,05	
		1	2
Katsominen EAO	36	13,72	
Ei-katsominen EAO	23	14,35	
Vastaaminen EAO	20	16,60	16,60
Vastaaminen JAO	12		20,67
Ei-katsominen JAO	17		21,88
Katsominen JAO	29		23,17

kaisemmin ohjelmoineita ryhmiä kiinni (taulukko 9). Analyysi vahvistaa edellä esitettyä päätelmää: ViLLE on erityisen hyödyllinen noviiseille ohjelmoinnin perusteiden oppimisessa, mutta vain, jos sitä käytetään taksonomian korkeammalla tasolla. Pelkkää passiivista visualisointia käyttämällä vastaaviin oppimistuloksiin ei päästä.

3.3 Aikaisemman käyttökokemuksen merkitys oppimisvaikutuksiin

Halusimme myös tutkia järjestelmän aikaisemman käyttökokemuksen merkitystä oppimistuloksiin. Oletettavaa on, että järjestelmään aikaisemmin tutustuneet opiskelijat pystyvät paremmin keskittymään opetettavaan asiaan, koska järjestelmän käyttöön liittyvä kognitiivinen taakka on pienempi. Tutkimus järjestettiin kahdella samansisällöllisellä lukion perusohjelmointikurssilla. Kurssien ainoana erona oli se, että jälkimmäisellä kurssilla opiskelijat tutustutettiin alusta alkaen ViLLEN käyttöliittymään ja ominaisuuksiin. Oppimista testattiin samankaltaisella tilaisuudella kuin aiemmissa luvuissa mainituissa tutkimuksissa, tosin esi- ja jälkitestissä olevat kysymykset muotoiltiin osittain uudestaan (mm. rekursio koettiin lukioi-

käisille liian monimutkaiseksi käsitteeksi). ViLLEä ennen käyttämättömät oppilaat muodostivat vertailuryhmän (N=17) ja aikaisempaa kokemusta omaavat koe-ryhmän (N=7). Vaikka ryhmäkoot olivat melko pieniä, tilastollisesti tulokset muodostuivat kuitenkin merkitseviksi.

Oppilaiden lähtötason yhteneväisyyden varmistamiseksi tarkasteltiin heidän aikaisempien tietojenkäsittelyn ja matematiikan kurssiensa arvosanoja ennen tutkimusta. Arvosanojen keskiarvot ja -hajonnat on esitetty taulukossa 10. Ryhmien välillä ei esiintynyt arvosanoissa tilastollisesti merkitsevää eroa, ja absoluutisestikin erotus on vähemmän kuin yksi piste. Esi- ja jälkitestin tulokset ryhmittäin on esitetty taulukossa 11.

Esitestissä ryhmien välillä ei ollut tilastollisesti merkitsevää eroa, mutta jälkitestissä tilanne muuttui. Merkitsevä ero löytyy sekä jaetuissa (eli sekä esi- että jälkitestissä esiintyvissä) että kaikissa kysymyksissä. Tuloksista voidaan päätellä, että aikaisempi tutustuminen järjestelmään vahvasti selvästi oppimisvaikutuksia. Voidaankin sanoa, että erilaisten järjestelmien oppimisvaikutuksia tutkittaessa on tärkeä tutustuttaa opiskelijat järjestelmän käyttöön. Lisäksi tutkimus vahvisti edellisissä luvuissa mainittuja tuloksia, joiden mukaan ViLLEstä on hyötyä ohjelmoinnin peruskäsitteiden oppimisessa:

Taulukko 10: Matematiikan ja tietojenkäsittelyn kurssien arvosanat (asteikko 4. . . 10).

<i>Ryhmä</i>	<i>Matematiikka</i>	<i>Tietojenkäsittely</i>
Vertailuryhmä	6,75 (1,60)	7,94 (1,09)
Koeryhmä	7,67 (2,25)	8,57 (1,62)
p-arvo	0,330	0,220

Taulukko 11: Esi- ja jälkitestin tulokset ryhmittäin.

	<i>Esitesti yhteensä</i>	<i>Jälk testi yhteensä (yhteiset)</i>	<i>Jälk testi yhteensä (kaikki)</i>
Vertailuryhmä (N=17)	7,12	12,59	16,94
Koeryhmä (N=7)	9,43	19,57	26,43
p-arvo	0,515	0,047	0,046

molemmat ryhmät saivat jälkitestissä tilastollisesti merkitsevästi paremmat pisteet kuin esitestissä. Tulokset on esitetty kokonaisuudessaan artikkelissa Laakso ym. [5].

3.4 Opiskelijoiden mielipiteitä järjestelmästä

Kvantitatiivisten testien lisäksi halusimme selvittää opiskelijoiden mielipiteitä järjestelmän käytöstä. Mielipiteet kerättiin Turun yliopiston Johdatus informaatioteknologiaan -kurssin osallistujilta kurssin lopussa. ViLLE oli olennaisena osana kurssin suoritusta, sillä kurssin demonstraatiot suoritettiin kokonaisuudessaan ViLLEn automaattisesti tarkastettavien ohjelmakoodin lukutaitotehtävien avulla. Kyselyyn vastasi yhteensä 114 (N=114) opiskelijaa ja se jakaantui kolmeen osaan: yleiset kysymykset järjestelmästä, järjestelmän hyödyt eri käsitteiden oppimisessa ja mielipiteet järjestelmän eri ominaisuuksista.

Ensimmäisessä osuudessa esitettiin

kuusi väittämää järjestelmästä ja pyydettiin opiskelijoita arvioimaan seitsenportaisella asteikolla (1: täysin eri mieltä, 7: täysin samaa mieltä) niiden todenmukaisuutta. Vastausten perusteella opiskelijat näyttävät olevan sitä mieltä, että järjestelmä soveltuu hyvin ohjelmoinnin alkeisopetukseen (kaikkien vastaajien keskiarvo 5,64), on helppokäyttöinen (ka. 5,49) ja helpottaa ohjelmointiin liittyvien käsitteiden oppimista (ka. 5,41).

Toisessa osuudessa kysyttiin oliko ViLLEstä hyötyä erilaisten ohjelmointikäsitteiden oppimisessa (taulukko 12); käytössä oli sama seitsenportainen asteikko. Opiskelijoiden vastausten perusteella ViLLE koettiin hyödylliseksi kaikkien kysytyjen käsitteiden ymmärtämisessä — ainoastaan taulukoiden osalta vastausten keskiarvo jäi alle viiden (ka. 4,73). Avointen palautekommenttien perusteella, tämä johtuu suurelta osin taulukkokysymyksiin vastaamiseen liittyneestä käytettävyysongelma.

Kolmanneksi kysyttiin mielipiteitä ViLLEn eri ominaisuuksien hyödyllisyy-

Taulukko 12: ViLLEn hyöty ohjelmointikäsitteiden ymmärtämisessä.

<i>Ohjelmointikäsite</i>	<i>Hyöty (asteikolla 1... 7)</i>
Muuttujat ja asetuslauseet	5,41 (1,37)
Valintalauseet	5,52 (1,15)
Toistolauseet	5,61 (1,19)
Totuusarvolausekkeet	5,38 (1,23)
Aliohjelmien määrittely	5,38 (1,25)
Aliohjelmakutsut	5,34 (1,32)
Aliohjelmaparametrit	5,24 (1,33)
Taulukot	4,73 (1,58)

destä (taulukko 13). Vastausten keskihajonnat on merkitty sulkeisiin. Lukuun ottamatta ohjelmien visualisointia usealla eri kielellä, kaikkien kysytyjen ominaisuuksien hyödyllisyys arvioitiin yli viideksi. Erityisesti muuttujien tilojen visualisointi ja tehtävien automaattinen arviointi arvioitiin hyödyllisiksi (keskiarvot 5,90 ja 5,80). Useilla kielillä visualisoinnin muita heikompi keskiarvo johtunee ainakin osittain siitä, että opiskelijat eivät todennäköisesti käyttäneet koko ominaisuutta. Muutenkin ominaisuudesta on varmasti eniten hyötyä silloin, kun jonkin ohjelmointikielen perusteet ovat jo hallussa, ja aloitetaan uuden ohjelmointikielen opiskelu.

Lisäksi opiskelijoita pyydettiin arvioimaan asteikolla 1–7 (1 = liian vähän, 7 = liian paljon) ViLLE-tehtävien kokonaisuutta kurssilla (ka. 4,48, keskihajonta 0,96), ja kysymysten määrää yksittäistä tehtävää kohden (ka. 4,13, keskihajonta 0,76). Keskiarvojen perusteella opiskelijoiden mielestä kurssilla oli sopivasti ViLLE-tehtäviä ja tehtäviin liittyi sopiva määrä kysymyksiä.

Opiskelijoilla oli myös mahdollisuus antaa vapaamuotoista palautetta. Palaute oli pääosin positiivista, mutta myös joitain negatiivisia kommentteja saatiin. Positiivisina puolina mainittiin esimerkiksi seu-

raavat:

- ”Mielestäni oli tehokkaampaa oppimisen kannalta tehdä ViLLE-tehtäviä kuin käydä asioita luennolla läpi.”
- ”Koin ViLLEn kurssilla todella hyödylliseksi! Sen avulla pystyi hahmottamaan ohjelmointia aivan eri tavalla kuin luennolla.”
- ”Kynnys tehtävien tekemiseen on matala, erilaisia harjoituksia on paljon ja tehtävät voi tehdä missä ja milloin huvittaa.”

Negatiiviset kommentit liittyivät yleensä ohjelman käyttöliittymän toiminnallisuuteen:

- ”Kysymysten kohdalla olisi kätevää päästä lukemaan ohjelman suorituksen historiaa sekä muistitilojen kehityksen.”
- ”Kaikki oleellimmat asiat eivät mahdu kuvaruudulle. Miksi ihmeessä ViLLEen pitää loggautua?”
- ”Ärsyttävää, kun ViLLE ikkunaa ei voinut rullata alas- tai ylöspäin suorituksen aikaisten kysymysten ilmestyessä ruutuun.”

Taulukko 13: ViLLEn hyöty ohjelmointikäsitteiden ymmärtämisessä.

Ominaisuus	Hyöty (asteikolla 1...7)
Ohjelman visualisointi useilla eri kielillä	4,93 (1,46)
Aliohjelmakutsujen visualisointi kutsupinossa	5,35 (1,24)
Muuttujien tilojen visualisointi	5,90 (1,17)
Ohjelmakoodirivin selitysteksti	5,40 (1,49)
Suorituksen aikaiset kysymykset	5,50 (1,24)
Tehtävien automaattinen arviointi	5,80 (1,28)

Lisäksi osa opiskelijoista kaipasi ViLLE-harjoitusten rinnalle (tai sijasta) perinteisempää opiskelua, esim. yhteisten mikroluokkaharjoitusten muodossa:

- ”Hyvä työkalu opiskeluun, mutta toivoisin enemmän pienryhmäharjoittelua.”
- ”Ihan hyvä ohjelmoinnin alkeiden opetteluun, mutta luennoista sain silti enemmän irti.”

Kokonaisuutena opiskelijoilla tuntui olevan hyvinkin positiivinen kuva ViLLEn hyödyllisyydestä, mutta parannettavaa luonnollisesti löytyi. Mielenpitoisten perusteella opiskelijat näyttäisivät pitävän mielekkäämpänä kurssia, jossa ViLLE-tehtävät on nivottu perinteisempien opiskelumenetelmien sekaan. Kohtuullisen runsas palaute ohjelman käyttöliittymään liittyvistä ongelmista on myös otettu huomioon järjestelmää edelleen kehitettäessä, ja mm. taulukkokysymyksiin vastaaminen toimii nykyisellään (toivottavasti) loogisemmin kuin versiossa, josta mielipiteet on kerätty.

4 Järjestelmän tulevaisuus

ViLLE on käytössä ohjelmoinnin opetuksessa useassa eri yliopistossa Suomessa,

ja lisäksi kevään 2009 aikana se on otettu käyttöön Monashin yliopistossa Australiassa. Tarkoituksenamme on tulevaisuudessa kehittää järjestelmää käyttäjäkokemuksien ja mielipiteiden perusteella ja tutkia edelleen minkälaisia vaikutuksia sillä on ohjelmoinnin oppimiseen. Tavoitteena on lisätä uusia ominaisuuksia tukemaan myös vuorovaikutuksen taksonomian korkeampia tasoja. Lisäksi järjestelmään kehitetään parhaillaan tehtäväpohjia, joiden avulla tehtäviä voidaan satunnaistaa. Tehtäväpohjien avulla voidaan esimerkiksi ohjelmien muuttujat ja niiden lähtöarvot arpoa satunnaisesti, jolloin samojen tehtävien uudelleen tekeminen muuttuu mielekkäämmäksi. Myös erilaisten tehtävätyyppien lisääminen on työn alla: tarkoituksena on lisätä ainakin koodirivin järjestelytehtäviä, joissa koodirivit annetaan sekoitettuna, ja opiskelijan pitää järjestää ne tietyn tuloksen aikaansaavaan järjestykseen.

5 Johtopäätökset

Kerättyjen tutkimustulosten, kokemusten ja palautteen perusteella voidaan yhteenvedon esittää seuraavat johtopäätökset ViLLEstä:

- ViLLEä käyttämällä aloittelijat voivat saavuttaa parempia oppimistu-

loksia lyhyenkin harjoitustilaisuuden aikana ...

- ... mutta vain, mikäli työkalua käytetään korkeammalla vuorovaikutuksen taksonomian tasolla. Pelkkä visualisointien seuraaminen (eli työkalun käyttö katsomistasolla) ei näytä riittävän tulosten saavuttamiseen; tulokset siis vahvistavat Napsin ym. [9] hypoteesin.
- Oppimishyötyjen saavuttamiseksi opiskelijat pitää lisäksi tutustuttaa järjestelmään riittävän perinpohjaisesti etukäteen, jotta fokus voidaan suunnata asioiden oppimiseen järjestelmän käytön opetteluun sijasta.
- Suurin osa opiskelijoista pitää ViLLEä hyödyllisenä perusteiden opetuksen kannalta, osan mielestä sen paras käyttömuoto olisi kuitenkin työkalun käytön integrointi muuhun opetukseen ja riittävän henkilökohtaisen ohjauksen tarjoaminen sen rinnalla.
- ViLLE tarjoaa opiskelijoille mahdollisuuden opetella ja harjoitella itsenäisesti sellaisia perusasioita, joihin luennoilla ei aikarajoitusten takia voida keskittyä kaikkien opiskelijoiden kannalta riittäväällä laajuudella.
- ViLLEn fokus on vahvasti ohjelmakoodin lukutaidon opettamisessa. Lopezin ym. [6] mukaan opiskelijoiden koodinlukutaito korreloi suoraan ohjelmakoodin kirjoitustaidon kanssa.
- Opettajan kannalta järjestelmän tärkeimpänä ominaisuutena voidaan pitää sen joustavuutta, erityisesti käytettävän ohjelmointikielen vallinnan suhteen.

Kaiken kaikkiaan kokemukset järjestelmästä vaikuttavat tähän mennessä positiiviselta: sekä oppimistulokset että opiskelijoiden mielipiteet vahvistavat käsitystä siitä, että ViLLE on aidosti hyödyllinen apuväline opettaessa ohjelmoinnin perusteita aloittelijoille. Kun aloittavien opiskelijoiden ohjelmointitaidot vaikuttavat vuosi vuodelta huonommilla, on selvää, että järjestelmälle löytyy käyttöä tulevaisuudessakin.

Viitteet

- [1] Ala-Mutka, K. 2005. Ohjelmoinnin opetuksen ongelmia ja ratkaisuja. Tekniikan opetuksen symposium 20.-21.10.2005. Helsinki University of Technology. <http://www.dipoli.tkk.fi/ok/p/reflektori/verkkojulkaisu/index.php?p=verkkojulkaisu>.
- [2] Hundhausen, C.D., Douglas, S.A. & Stasko, J.D. 2002. A Meta-study of Algorithm Visualization Effectiveness. *Journal of Visual Languages and Computing*, 13, 259–290.
- [3] Kaila, E., Rajala, T., Laakso, M.-J. & Salakoski, T. 2009. Effects, Experiences and Feedback from Studies of a Program Visualization Tool. *Informatics in Education*, 8, 1, 17–34.
- [4] Laakso, M.-J., Myller, N. & Korhonen, A. 2008a. Comparing learning performance of students using algorithm visualizations collaboratively on different engagement levels. Ilmestyy teoksessa *Journal of Educational Technology and Society*.
- [5] Laakso, M.-J., Rajala, T., Kaila, E. & Salakoski, T. 2008b. The Impact Of Prior Experience In Using A Visualization Tool On Learning To Program. Teoksessa *Proceedings of CELDA 2008*, Freiburg, Germany.

- [6] Lopez, M., Whalley, J., Robbins, P. & Lister, R. 2008. Relationships between reading, tracing and writing skills in introductory programming. Teoksessa Proceeding of the fourth international workshop on Computing education research, September 06-07, 2008, Sydney, Australia, 101-112.
- [7] Malmi, L., Karavirta, V., Korhonen, A., Nikander, J., Seppälä, O. & Silvasti, P. 2004. Visual Algorithm Simulation Exercise System with Automatic Assessment: TRAKLA2. *Informatics in Education*, 3, 2, 267-288.
- [8] McCracken, M., Almstrum, V., Diaz, D., Guzdial, M., Hagan, D., Kolikant, Y., Laxer, C., Thomas, L., Utting, I. & Wilusz, T. 2001. A Multi-National, Multi-Institutional Study of Assessment of Programming Skills of First-year CS Students. *ACM SIGCSE Bulletin*, 33, 4, 125-140.
- [9] Naps, T. L., Rößling, G., Almstrum, V., Dann, W., Fleischer, R., Hundhausen, C., Korhonen, A., Malmi, L., McNally, M., Rodger, S. & Velázquez-Iturbide, J. Á. 2002. Exploring the Role of Visualization and Engagement in Computer Science Education. Teoksessa Working Group Reports from ITiCSE on Innovation and Technology in Computer Science Education, 35, 2, 131-152.
- [10] Rajala, T., Laakso, M.-J., Kaila, E. & Salakoski, T. 2008. Effectiveness of Program Visualization: A Case Study with the ViLLE Tool. *Journal of Information Technology Education: Innovations in Practice*, 7, 15-32.
- [11] Wiggins, M. 1998. An overview of program visualization tools and systems. Teoksessa Proceedings of the 36th annual Southeast regional conference, 194-200.