ePublishing
ΕΘΝΙΚΟ ΚΕΝΤΡΟ ΤΕΚΜΗΡΙΩΣΗΣ
NATIONAL DOCUMENTATION CENTRE

Συνέδρια της Ελληνικής Επιστημονικής Ένωσης Τεχνολογιών Πληροφορίας & Επικοινωνιών στην Εκπαίδευση

Τόμ. 1 (2012)

6ο Συνέδριο Διδακτική της Πληροφορικής

Συνεργατικός Προγραμματισμός από Απόσταση και Προσαρμοστική Υποστήριξη της Συνεργασίας

Δ. Τσομπανούδη, Μ. Σατρατζέμη

Βιβλιογραφική αναφορά:

Τσομπανούδη Δ., & Σατρατζέμη Μ. (2022). Συνεργατικός Προγραμματισμός από Απόσταση και Προσαρμοστική Υποστήριξη της Συνεργασίας. *Συνέδρια της Ελληνικής Επιστημονικής Ένωσης Τεχνολογιών Πληροφορίας & Επικοινωνιών στην Εκπαίδευση*, *1*, 073–082. ανακτήθηκε από https://eproceedings.epublishing.ekt.gr/index.php/cetpe/article/view/4426

https://epublishing.ekt.gr | **e-Εκδότης:** EKT | **Πρόσβαση:** 06/12/2025 00:10:14

Συνεργατικός Προγραμματισμός από Απόσταση και Προσαρμοστική Υποστήριξη της Συνεργασίας

Δ. Τσομπανούδη, Μ. Σατρατζέμη

Τμήμα Εφαρμοσμένης Πληροφορικής, Πανεπιστήμιο Μακεδονίας
{despinats, maya}@uom.gr

Περίληψη

Στοχεύοντας να αποκομίσει τα οφέλη της συνεργατικής μάθησης και ταυτοχρόνως να μυήσει τους εκπαιδευόμενους σε πραγματικές συνθήκες ανάπτυξης λογισμικού, ο συνεργατικός προγραμματισμός αποτελεί μία τεχνική στην οποία δύο άτομα διαμοιράζονται έναν υπολογιστή για να σχεδιάσουν και να γράψουν από κοινού ένα πρόγραμμα. Αν και η εφαρμογή του ξεκίνησε στον επαγγελματικό χώρο με στόχο την ανάπτυξη ποιοτικότερου λογισμικού, σύντομα δοκιμάστηκε και στο χώρο της εκπαίδευσης με πολύ ενθαρρυντικά αποτελέσματα. Ταυτόχρονα, αναπτύχθηκαν αρκετά συστήματα για να καλύψουν την ανάγκη εφαρμογής του συνεργατικού προγραμματισμού από απόσταση. Στην παρούσα εργασία γίνεται μία ανασκόπηση του συνεργατικού προγραμματισμού και παρουσιάζονται τα βασικά χαρακτηριστικά των συστημάτων που υποστηρίζουν το συνεργατικό προγραμματισμό από απόσταση. Επιπλέον, προτείνονται στρατηγικές για την προσαρμοστική υποστήριξη της συνεργατικής ανάπτυξης κώδικα, οι οποίες αναμένεται να περιορίσουν τα μειονεκτήματα που διαπιστώθηκαν στα υπάρχοντα συστήματα συνεργατικού προγραμματισμού, και να υποστηρίξουν τους αρχάριους προγραμματιστές με στοχευμένες παρεμβάσεις στην εκπαιδευτική τους διαδικασία.

Λέξεις κλειδιά: *συνεργατικός προγραμματισμός, προσαρμοστική υποστήριξη της συνεργασίας.*

Abstract

With the aim to gain the benefits of collaborative learning and also to introduce students in professional software development conditions, pair programming is a computer programming technique where two programmers share one computer to design and write computer programs. While pair programming has its origins in the software industry, soon it was applied in computer science education with very promising results. Several systems have been developed to cover the need for pair programming over distance. This paper presents an overview of pair programming and the features of distributed pair programming systems. Moreover, adaptation strategies for supporting collaborative programming are proposed, which aim to eliminate the weaknesses of current systems of distributed pair programming, and to support novice programmers by making appropriate interventions during the learning process.

Keywords: *pair programming, adaptive collaboration support.*

1. Εισαγωγή

Στην επιστήμη της Πληροφορικής ένα από τα δυσκολότερα μαθήματα που καλούνται να αντιμετωπίσουν οι φοιτητές κατά τη διάρκεια των σπουδών τους είναι ο προγραμματισμός Η/Υ. Κυρίως οι αρχάριοι προγραμματιστές είναι αυτοί που αντιμετωπίζουν τα περισσότερα προβλήματα καθώς δυσκολεύονται να αποκτήσουν τις δεξιότητες που απαιτούνται για την κατανόηση και επίλυση προβλημάτων (Lister et al., 2004). Το γεγονός αυτό έχει αρνητικό αντίκτυπο στην απόδοσή τους τόσο κατά τη διάρκεια του εξαμήνου όσο και στις τελικές εξετάσεις. Η χαμηλή απόδοση των φοιτητών οδήγησε στην αναζήτηση νέων προσεγγίσεων στη διδασκαλία του προγραμματισμού που θα βοηθούσαν τους αρχάριους προγραμματιστές να αντιμετωπίσουν τις δυσκολίες τους. Μία από αυτές τις προσεγγίσεις ήταν ο συνεργατικός προγραμματισμός (ΣΠ), ο οποίος είχε εφαρμοστεί με μεγάλη επιτυχία από επαγγελματίες προγραμματιστές. Η θετική αξιολόγησή του τόσο στον επαγγελματικό χώρο όσο και στο χώρο της εκπαίδευσης είχε ως αποτέλεσμα να αναπτυχθούν συστήματα για να μπορεί να εφαρμοστεί ο συνεργατικός προγραμματισμός και από απόσταση. Ένα σύστημα που υποστηρίζει τον συνεργατικό προγραμματισμό από απόσταση (ΣΠαΑ) διευρύνει τις δυνατότητές του, καθώς μπορεί να εμπλουτιστεί με λειτουργίες που ανταποκρίνονται στις ανάγκες των προγραμματιστών και ενισχύουν τη συνεργασία τους. Πάνω σε αυτό τον άξονα βασίστηκε η έρευνά μας, με στόχο τη βελτίωση της εκπαιδευτικής διαδικασίας σε περιβάλλοντα που εφαρμόζουν το ΣΠαΑ.

Στην παρούσα εργασία παρουσιάζονται αρχικά τα χαρακτηριστικά του ΣΠ και τα αποτελέσματα ερευνών από την εφαρμογή του στην εκπαίδευση. Στη συνέχεια παρουσιάζεται ο ΣΠαΑ και τα συστήματα που μπορούν να χρησιμοποιηθούν για την εφαρμογή του. Τέλος, περιγράφονται τα χαρακτηριστικά που θα μπορούσαν να ενσωματωθούν σε ένα σύστημα ΣΠαΑ για να λειτουργήσουν ως μηχανισμοί υποστήριξης της συνεργατικής ανάπτυξης λογισμικού.

2. Ο Συνεργατικός Προγραμματισμός

Μία εναλλακτική μέθοδος διδασκαλίας του προγραμματισμού αποτελεί η τεχνική του Συνεργατικού Προγραμματισμού (pair programming) η οποία μπορεί να εφαρμοστεί και από αρχάριους προγραμματιστές. Στο ΣΠ η ανάπτυξη ενός προγράμματος λογισμικού γίνεται από δύο άτομα τα οποία διαμοιράζονται έναν υπολογιστή για το σχεδιασμό και την ανάπτυξη του κώδικα. Ό ένας από τους δύο αναλαμβάνει το ρόλο του «οδηγού» έχοντας τον έλεγχο του πληκτρολογίου και γράφει τον κώδικα του προγράμματος, ενώ ο δεύτερος, ο οποίος καλείται «παρατηρητής» ελέγχει τον κώδικα για συντακτικά ή λογικά λάθη και προτείνει εναλλακτικές λύσεις. Οι ρόλοι του «οδηγού» και του «παρατηρητή» θα πρέπει να εναλλάσσονται ανά τακτά χρονικά διαστήματα για να επωφελούνται και οι δύο προγραμματιστές από τα πλεονεκτήματα που προσφέρει ο κάθε ρόλος ξεχωριστά, και για να παραμένει σταθερή η απόδοση της ομάδας. Η ιδέα του ΣΠ προήλθε από επαγγελματίες προγραμματιστές και βασίζεται στη συνεργασία των προγραμματιστών κατά τη διάρκεια ανάπτυξης ενός

πακέτου λογισμικού (Williams et al., 2000). Στόχος της συνεργατικής ανάπτυξης λογισμικού είναι η παραγωγή ποιοτικότερων προγραμμάτων και η ταχύτερη παράδοση του λογισμικού στον πελάτη, καθώς με τη συνεργασία παρατηρείται ότι επιτυγχάνεται σημαντική βελτίωση στην απόδοση των προγραμματιστών. Επιπλέον, η ομαδική εργασία βοηθά στο διαμοιρασμό γνώσεων και δεξιοτήτων και ενισχύει τη μάθηση μέσω της αλληλεπίδρασης και της επικοινωνίας με τα υπόλοιπα μέλη της ομάδας.

Οι διαπιστώσεις αυτές οδήγησαν στη δοκιμή του ΣΠ και στο χώρο της εκπαίδευσης με στόχο να επωφεληθούν οι αρχάριοι προγραμματιστές από τα πλεονεκτήματα που προσφέρει ο ΣΠ και να εξοικειωθούν οι φοιτητές με έναν τρόπο εργασίας που θα αντιμετωπίσουν και μελλοντικά ως επαγγελματίες. Ακολούθησαν αρκετές έρευνες που παρουσίαζαν τα αποτελέσματα από την εφαρμογή του ΣΠ στην εκπαίδευση. Ένα από τα βασικότερα πλεονεκτήματα που αναφέρεται σε αυτές αποτελεί το γεγονός ότι τα προγράμματα που παράγονται έχουν ποιοτικότερο κώδικα και εμφανίζουν λιγότερα λάθη σε σύγκριση με προγράμματα που αναπτύσσονται από ένα μόνο άτομο (Cockburn & Williams, 2001; Duque & Bravo, 2008; McDowell et al., 2003a; Sanjay & Vanshi, 2010; Zacharis, 2009). Επιπλέον, οι έρευνες αναφέρουν ότι οι φοιτητές που εργάζονται σε ομάδες παρουσιάζουν μεγαλύτερη αυτοπεποίθηση για τις ασκήσεις που παραδίδουν, αντιμετωπίζουν λιγότερο άγχος στο μάθημα και δηλώνουν ότι απολαμβάνουν περισσότερο την ενασχόλησή τους με τον προγραμματισμό (Boyer et al., 2008; Cockburn & Williams, 2001; Zacharis, 2009). Έχει παρατηρηθεί επίσης, ότι οι φοιτητές που χρησιμοποιούν την τεχνική του ΣΠ παρουσιάζουν καλύτερη απόδοση στις εργαστηριακές ασκήσεις και στις τελικές εξετάσεις (Jun et al., 2007; McDowell et al., 2003b; Williams et al., 2002), και έχουν περισσότερες πιθανότητες να παρακολουθήσουν μέχρι το τέλος και να περάσουν ένα μάθημα προγραμματισμού (McDowell et al., 2003b; Williams et al., 2002). Ο ΣΠ αποτελεί και μία μορφή συνεργατικής μάθησης (Preston, 2005), στην οποία οι εμπλεκόμενοι έχουν τη δυνατότητα να συζητούν και να διαπραγματεύονται τις ενδεχόμενες λύσεις σε ένα πρόβλημα, μπορούν να διαμοιράζονται τις γνώσεις τους κι επιπλέον έχουν τη δυνατότητα να διδαχθούν από το συνεργάτη τους καλές πρακτικές στον προγραμματισμό (Sanjay & Vanshi, 2010; Williams et al., 2002). Ακόμη και έμπειροι προγραμματιστές που έχουν εφαρμόσει το ΣΠ στην εργασία τους, αναφέρουν ότι μέσω της συζήτησης με το συνεργάτη τους και της προσπάθειάς τους να εξηγήσουν τη λογική πίσω από κάθε ενέργειά τους μπόρεσαν να αποφύγουν κάποια λάθη στον κώδικά τους και να βελτιώσουν τις προγραμματιστικές τους δεξιότητες (Williams et al., 2000). Οφέλη από την υιοθέτηση του ΣΠ σε εργαστηριακά μαθήματα λαμβάνουν και οι καθηγητές, καθώς μειώνεται για αυτούς ο φόρτος εργασίας επειδή οι φοιτητές μπορούν πλέον να συζητούν και να λύνουν μόνοι τους τις απορίες τους συζητώντας μεταξύ τους (Williams et al., 2002; Hanks, 2007). Έτσι, ο καθηγητής έχει την ευχέρεια να αφιερώνει περισσότερο χρόνο σε ερωτήματα των φοιτητών που είναι πιο ουσιαστικά και όχι σε τετριμμένες απορίες.

Αν και ο ΣΠ φαίνεται να προσφέρει πολλά πλεονεκτήματα στους προγραμματιστές, ωστόσο έχουν αναφερθεί και μειονεκτήματα. Πολλά από αυτά τα μειονεκτήματα απαντώνται στις περισσότερες μορφές συνεργατικής μάθησης και αποτελούν απόρροια της ομαδικής εργασίας. Βασικό παράγοντα στην επιτυχή έκβαση μιας συνεργασίας αποτελεί η σύνθεση της ομάδας. Έχουν δοκιμαστεί διάφορες στρατηγικές για την επιλογή των μελών της ομάδας, άλλες βασιζόμενες στο επίπεδο γνώσεων των φοιτητών, άλλες σε ελεύθερη επιλογή συνεργάτη από τους φοιτητές, ωστόσο καμία από αυτές δεν έχει αποδειχτεί ότι είναι πιο αποδοτική από τις υπόλοιπες. Ένα άλλο βασικό πρόβλημα στις ομαδικές εργασίες είναι η μη ισορροπημένη συμμετοχή των μελών στο κοινό έργο και οι μεγάλες διαφορές στο βαθμό συνεισφοράς τους. Πολλές φορές δηλαδή, οι μαθητές που έχουν περισσότερες γνώσεις ή που είναι πιο δυναμικοί ως χαρακτήρες αναλαμβάνουν περισσότερες αρμοδιότητες από αυτές που τους αναλογούν με αποτέλεσμα να παραγκωνίζονται τα υπόλοιπα μέλη της ομάδας. Ένα ακόμη μειονέκτημα που έχει αναφερθεί σε μερικές έρευνες είναι ότι οι φοιτητές που εργάζονται σε ομάδες χρειάζονται περισσότερο χρόνο για την ολοκλήρωση της εργασίας τους σε σύγκριση με αυτούς που εργάζονται μόνοι τους, διότι δαπανούν αρκετό χρόνο για τον μεταξύ τους συντονισμό (Cockburn & Williams, 2001; Duque & Bravo, 2008). Σχετικά, αναφέρεται ακόμη, ότι ο ΣΠ προϋποθέτει τα άτομα που συνεργάζονται να βρίσκονται στον ίδιο χώρο, κάτι που μπορεί να οδηγήσει σε προβλήματα συνεννόησης και στη δυσκολία συγχρονισμού.

Τα μειονεκτήματα του ΣΠ που αναφέρθηκαν, μπορούν να αντιμετωπιστούν με διάφορους τρόπους οι οποίοι αναλύονται στη συνέχεια, και με εξειδικευμένα λογισμικά τα οποία περιγράφονται αναλυτικότερα σε επόμενη ενότητα.

3. Συνεργατικός Προγραμματισμός από Απόσταση

Για να αντιμετωπισθεί το πρόβλημα της υποχρεωτικής παρουσίας των μελών της ομάδας στον ίδιο χώρο και το οποίο οδηγεί σε προβλήματα συγχρονισμού, αναπτύχθηκαν διάφορα συστήματα για να μπορεί να εφαρμοστεί ο Συνεργατικός Προγραμματισμός από Απόσταση (distributed pair programming). Σε αυτή την παραλλαγή του ΣΠ, τα μέλη της ομάδας βρίσκονται σε διαφορετικές τοποθεσίες και μπορούν να συνεργαστούν είτε χρησιμοποιώντας λογισμικά που επιτρέπουν την κοινή χρήση της επιφάνειας εργασίας ενός υπολογιστή είτε μέσω ενός κοινόχρηστου συντάκτη κώδικα (Sanjay & Vanshi, 2010). Έχοντας αυτή τη δυνατότητα, η φυσική παρουσία στον ίδιο χώρο δεν είναι πλέον απαραίτητη, κι επιπλέον μπορούν να συνεργάζονται άτομα που μένουν σε απομακρυσμένες περιοχές. Ένα τέτοιο σύστημα μπορεί να έχει εφαρμογή στα ανοικτά πανεπιστήμια ή στην εκπαίδευση από απόσταση όπου οι εκπαιδευόμενοι εργάζονται κυρίως από το σπίτι τους και καλούνται πολλές φορές να συνεργαστούν με τους συμφοιτητές τους μέσω του διαδικτύου και των υπηρεσιών που αυτό προσφέρει. Αρχικά, με την εμφάνιση των συστημάτων αυτών υπήρχε μία επιφύλαξη για το αν η συνεργασία θα αποφέρει τα ίδια οφέλη με τον δια ζώσης ΣΠ, και αν θα επηρεάσει αρνητικά την απόδοση των

προγραμματιστών. Έρευνες που ακολούθησαν έδειξαν ότι η εφαρμογή του ΣΠαΑ δεν φαίνεται να έχει τελικά κάποιο αρνητικό αντίκτυπο στην απόδοση των φοιτητών (Hanks, 2008; Jun et al., 2007). Επομένως, ο ΣΠαΑ, διατηρεί όλα τα πλεονεκτήματα του ΣΠ κι επιπλέον περιορίζει το πρόβλημα της αναγκαστικής φυσικής παρουσίας της ομάδας στον ίδιο χώρο.

4. Συστήματα για το Συνεργατικό Προγραμματισμό από Απόσταση

Όπως προαναφέρθηκε, για να μπορεί να εφαρμοστεί ο ΣΠαΑ, θα πρέπει να χρησιμοποιηθεί είτε ένα λογισμικό που επιτρέπει την κοινή χρήση της επιφάνειας εργασίας ενός υπολογιστή είτε να χρησιμοποιηθεί ένα εξειδικευμένο λογισμικό που θα περιλαμβάνει έναν κοινόχρηστο συντάκτη κώδικα στον οποίο θα προγραμματίζουν από κοινού και σε πραγματικό χρόνο οι δύο συνεργάτες. Ωστόσο, έπειτα από δοκιμή και των δύο μεθόδων επικράτησαν να χρησιμοποιούνται τα εξειδικευμένα λογισμικά καθώς στα συστήματα διαμοιρασμού της επιφάνειας εργασίας υπήρχαν αρκετά προβλήματα, κυρίως στην ταχύτητα μετάδοσης των δεδομένων από τον έναν υπολογιστή στον άλλο.

Αρχικά, τα συστήματα που αναπτύχθηκαν για τον ΣΠαΑ ήταν αυτόνομα, αλλά στη συνέχεια άρχισαν να ενσωματώνονται σε ολοκληρωμένα περιβάλλοντα ανάπτυξης λογισμικού έτσι ώστε οι προγραμματιστές να έχουν τη δυνατότητα να χρησιμοποιούν περιβάλλοντα που τους είναι οικεία και που τους προσφέρουν πλήρη λειτουργικότητα για την ανάπτυξη ενός προγράμματος. Για να υποστηρίξει ένα σύστημα το ΣΠαΑ θα πρέπει να ικανοποιεί κάποιες βασικές απαιτήσεις οι οποίες κρίνονται απαραίτητες για την επικοινωνία και τον συντονισμό μεταξύ των χρηστών. Τα βασικότερα χαρακτηριστικά που πρέπει να έχει ένα τέτοιο σύστημα είναι ένας κοινός χώρος εργασίας, ένα κανάλι επικοινωνίας, πολιτικές ελέγχου του κοινόχρηστου χώρου, υποστήριξη χειρονομιών/επισημάνσεων και βασικούς μηχανισμούς ενημέρωσης (Winkler et al., 2010). Τα χαρακτηριστικά αυτά υποστηρίζονται από τα περισσότερα συστήματα ΣΠαΑ που υπάρχουν σήμερα. Στον κοινό χώρο εργασίας οι χρήστες διαμοιράζονται έναν συντάκτη πηγαίου κώδικα και μπορούν να πληκτρολογούν σε πραγματικό χρόνο τις εντολές τους. Το σύστημα θα πρέπει να υποστηρίζει τους ρόλους του «οδηγού» και του «παρατηρητή» και να επιτρέπει την εναλλαγή ρόλων. Αυτό σημαίνει ότι θα πρέπει να αποτρέπεται η ταυτόχρονη πληκτρολόγηση στον κοινό συντάκτη για να μην απομακρύνονται οι μαθητευόμενοι από το ρόλο τους και τον στόχο που έχει αυτός. Για την επικοινωνία μεταξύ των χρηστών το σύστημα θα πρέπει να παρέχει τουλάχιστον ένα κανάλι επικοινωνίας. Όλα τα υπάρχοντα συστήματα ΣΠαΑ παρέχουν επικοινωνία μέσω ανταλλαγής γραπτών μηνυμάτων, ενώ σε μερικά από αυτά χρησιμοποιήθηκαν εξωτερικά λογισμικά φωνητικής συνομιλίας, κάτι το οποίο φαίνεται να είναι περισσότερο βολικό για έναν προγραμματιστή ο οποίος θα πρέπει αφενός να

επικοινωνεί γραπτώς με το συνεργάτη του και αφετέρου να πληκτρολογεί κώδικα στον κοινόχρηστο χώρο. Άλλη μία βασική λειτουργία που πρέπει να παρέχουν τα συστήματα ΣΠαΑ είναι οι ενέργειες κατάδειξης και επισήμανσης στον κώδικα από τους προγραμματιστές. Καθώς ο ρόλος του «παρατηρητή» είναι να ελέγχει τον κώδικα που γράφει ο «οδηγός» για λογικά ή συντακτικά λάθη, όταν η συνεργασία γίνεται από απόσταση θα πρέπει να έχει ο «παρατηρητής» τη δυνατότητα να ενημερώνει τον «οδηγό» για το σημείο που έκανε ένα λάθος επισημαίνοντας την ακριβή θέση με το ποντίκι του. Έτσι η συνεννόηση μεταξύ του «οδηγού» και του «παρατηρητή» γίνεται πιο αποτελεσματική και πιο γρήγορη. Τέλος, είναι σημαντικό να υπάρχουν στο σύστημα λειτουργίες που θα ενημερώνουν τους χρήστες για την κατάσταση και τις ενέργειες των συνεργατών τους μέσω κατάλληλων ενδείξεων.

Κάποια συστήματα που εφαρμόζουν το ΣΠαΑ και ενσωματώνουν τα παραπάνω χαρακτηριστικά είναι το RIPPLE, το XPairtise και το GrewpEdit. Τα δύο πρώτα συστήματα αποτελούν plugins του Eclipse, το οποίο είναι ένα δημοφιλές ολοκληρωμένο περιβάλλον ανάπτυξης λογισμικού, και το τρίτο σύστημα αποτελεί μία αυτόνομη εφαρμογή.

Το RIPPLE είναι ένα λογισμικό ανοικτού κώδικα και αναπτύχθηκε με σκοπό την εφαρμογή του ΣΠαΑ σε Java, την απομακρυσμένη διδασκαλία προγραμματισμού και τη συλλογή δεδομένων για ερευνητικούς σκοπούς (Boyer et al., 2008). Βασίζεται σε ένα παλαιότερο plugin, το Sangam, το οποίο εμπλουτίστηκε με την αυτόματη καταγραφή των ενεργειών των χρηστών με σκοπό τη χρήση τους σε ερευνητικές εργασίες. Στο RIPPLE, ο ΣΠ υλοποιείται υιοθετώντας τους ρόλους του «οδηγού» και του «παρατηρητή» και η επικοινωνία μεταξύ τους επιτυγχάνεται μέσω ενός παραθύρου chat. Μόνο ο «οδηγός» έχει το δικαίωμα να πληκτρολογεί κώδικα στον κοινόχρηστο συντάκτη ενώ αγνοείται και δεν γίνεται ορατό στον «οδηγό» οτιδήποτε πληκτρολογήσει ο «παρατηρητής». Το RIPPLE αξιολογήθηκε σε ένα εισαγωγικό μάθημα προγραμματισμού κατά τη διάρκεια μιας εργαστηριακής δραστηριότητας. Τα αποτελέσματα της αξιολόγησης έδειξαν ότι οι φοιτητές το βρήκαν εύκολο στη χρήση, ότι τους φάνηκε ευχάριστη η διαδικασία της συνεργατικής επίλυσης μιας δραστηριότητας και ότι θα το χρησιμοποιούσαν ξανά αν τους δινόταν η ευκαιρία (Boyer et al., 2008).

Το XPairtise, άλλο ένα plugin ανοικτού κώδικα, υποστηρίζει κι αυτό το ΣΠ υιοθετώντας τους ρόλους του «οδηγού» και του «παρατηρητή» κι επιπλέον εισάγει το ρόλο του «θεατή». Οι «θεατές» μπορούν να συμμετέχουν σε συζητήσεις και να παρακολουθούν την ανάπτυξη ενός προγράμματος αλλά δεν μπορούν να παρεμβαίνουν και να κάνουν επισημάνσεις στον κώδικα. Επιπλέον, το XPairtise υποστηρίζει τη διαχείριση λογαριασμών των χρηστών, την απομακρυσμένη επισήμανση κώδικα, ένα κοινόχρηστο καμβά για τη δημιουργία σχεδίων και γραπτή επικοινωνία. Το XPairtise αξιολογήθηκε σε διάστημα 18 εβδομάδων και τα αποτελέσματα έδειξαν ότι υπήρξε άνιση συμμετοχή στον προγραμματισμό και ότι η

γραπτή επικοινωνία και η απομακρυσμένη επισήμανση κώδικα χρησιμοποιήθηκαν λιγότερο απ' ότι αναμενόταν (Schümmer & Lukosch, 2009), κάτι που πιθανότατα να οφείλεται στη χρήση του Skype για τη συνομιλία των χρηστών μεταξύ τους.

Το GrewpEdit ήταν ένα από τα πρώτα συστήματα για την εφαρμογή του ΣΠαΑ γι' αυτό και είναι ένα από τα λίγα συστήματα που αποτελεί αυτόνομη εφαρμογή. Υποστηρίζει κοινή χρήση ενός κειμενογράφου, ενός σχεδιαστικού καμβά, κι ενός συντάκτη κώδικα σε Java, C, HTML και Scheme (Granville & Hickey, 2005). Το GrewpEdit δεν υποστηρίζει ωστόσο τους ρόλους «οδηγού» και «παρατηρητή», κάτι που σημαίνει ότι οι συμμετέχοντες μπορούν να πληκτρολογούν ταυτόχρονα στον κοινό συντάκτη. Ένα αυτόνομο σύστημα έχει το μειονέκτημα ότι δεν μπορεί να παρέχει τις λειτουργίες ενός ολοκληρωμένου περιβάλλοντος ανάπτυξης λογισμικού γι' αυτό και τα περισσότερα συστήματα ΣΠ ενσωματώθηκαν σε εφαρμογές όπως είναι το Eclipse.

Τα περισσότερα συστήματα που μπορούν να χρησιμοποιηθούν για τον ΣΠαΑ έχουν περίπου τα ίδια χαρακτηριστικά με τα συστήματα που προαναφέρθηκαν και καλύπτουν τις βασικές προϋποθέσεις για την εφαρμογή του ΣΠ. Όταν η τεχνική του ΣΠ χρησιμοποιείται στα πλαίσια διδασκαλίας του προγραμματισμού, τότε τα συστήματα αυτά δεν μπορούν να υποστηρίξουν σε ικανοποιητικό βαθμό τους αρχάριους προγραμματιστές και τις ανάγκες τους. Όπως φαίνεται από την αξιολόγηση του XPairtise, τα υπάρχοντα συστήματα δεν μπορούν να εξαλείψουν τα μειονεκτήματα του ΣΠ όπως είναι η μη ισορροπημένη συνεισφορά στη συνεργατική εργασία. Επιπλέον, δεν διατηρούν στοιχεία για τις προγραμματιστικές συνεδρίες, τα οποία θα μπορούσαν να αξιοποιηθούν για εξαγωγή συμπερασμάτων σχετικά με τη συνεισφορά του ΣΠ, όπως και για τη δοκιμή διάφορων στρατηγικών στη σύνθεση της ομάδας ή για την καταγραφή του επιπέδου γνώσεων των χρηστών. Λαμβάνοντας υπόψη τους περιορισμούς που παρατηρήθηκαν στα υπάρχοντα συστήματα ΣΠ, και ταυτόχρονα τα προβλήματα που αντιμετωπίζουν αρχάριοι προγραμματιστές, ξεκινήσαμε μία μελέτη για τα χαρακτηριστικά που θα μπορούσαν να ενσωματωθούν σε ένα τέτοιο σύστημα για να το κάνουν πιο εποικοδομητικό για τον αρχάριο μαθητή. Οι πρόσθετες λειτουργίες που προτείνουμε, εντάσσονται σε ένα γενικότερο πλαίσιο προσαρμοστικής υποστήριξης της ομάδας που θα ενισχύουν τη συνεργατικότητα και τη διδασκαλία του προγραμματισμού και περιγράφονται στη συνέχεια.

5. Προσαρμοστική Υποστήριξη του Συνεργατικού Προγραμματισμού

Καθώς ένα από τα βασικότερα μειονεκτήματα του ΣΠ και γενικότερα της ομαδικής εργασίας είναι η μη ισορροπημένη συνεισφορά της ομάδας στο κοινό έργο, το σύστημα θα πρέπει να εξασφαλίζει ότι όλα τα μέλη της ομάδας θα έχουν ίση συμμετοχή (Diziol et al., 2010). Η συνεισφορά ενός ατόμου σε ένα συλλογικό έργο φαίνεται από το χρόνο που έχει δαπανήσει αυτό στο σύνολο ή από το έργο που έχει

παραγάγει (Gardner, 2003). Αν θεωρήσουμε ότι ο χρόνος που αφιερώνει κάποιος σε μία δραστηριότητα προγραμματισμού δεν είναι απολύτως αντικειμενική και έγκυρη παράμετρος για να εκτιμηθεί η συνεισφορά του, τότε θα μπορούσε να χρησιμοποιηθεί ως κριτήριο σύγκρισης οι γραμμές κώδικα που έχει πληκτρολογήσει ο καθένας ώστε να υπολογίσουμε το βαθμό συμμετοχής του. Έτσι, ένα σύστημα θα μπορούσε να προτείνει εναλλαγές ρόλων για να εξασφαλίζεται ισορροπημένη συμμετοχή. Σε αυτή την περίπτωση θα επιτυγχάναμε ίση συμμετοχή, αλλά δεν θα γνωρίζαμε αν οι εκπαιδευόμενοι είχαν αποκτήσει και τις ίδιες γνώσεις/δεξιότητες. Ένα σύστημα λοιπόν θα πρέπει να ελέγχει εκτός από την ποσότητα αλλά και το είδος της συνεισφοράς έτσι ώστε να αναθέτει κατάλληλους ρόλους στους προγραμματιστές με στόχο να αποκτούν ίδιες δεξιότητες και να αποφεύγεται η επαναλαμβανόμενη χρήση των ίδιων εννοιών από το ίδιο άτομο.

Η ανάθεση ρόλων από το ίδιο το σύστημα δεν θα πρέπει να γίνεται ωστόσο σε κάθε δραστηριότητα που λύνουν οι μαθητευόμενοι αλλά θα πρέπει να τους δίνεται και η ευκαιρία να αποφασίζουν οι ίδιοι τους πότε θα εναλλάσσουν τους ρόλους «οδηγού» και «παρατηρητή». Αυτό μπορεί να επιτευχθεί μέσω δραστηριοτήτων διαφορετικού επιπέδου, όπου στο πρώτο επίπεδο θα επιβάλλονται εναλλαγές ρόλων κατά τη διάρκεια επίλυσης μιας δραστηριότητας έτσι ώστε να επιτυγχάνεται η ισορροπημένη απόκτηση γνώσης, ενώ στο δεύτερο επίπεδο δραστηριοτήτων οι εναλλαγές ρόλων δεν θα είναι επιβεβλημένες. Για παράδειγμα, η εμφάνιση του ποσοστού συμμετοχής κάθε μέλους της ομάδας θα στόχευε στην αυτόβουλη κινητοποίηση της συμμετοχής του. Έτσι, θα επιδεικνύεται αρχικά ο ενδεδειγμένος τρόπος συνεργασίας των εκπαιδευόμενων και στη συνέχεια θα τους δίνεται η δυνατότητα να επιλέγουν οι ίδιοι πώς θα συνεργαστούν. Η αξιολόγηση αυτής της προσέγγισης θα δείξει αν οι προσαρμοσμένες αναθέσεις ρόλων μπορούν να βοηθήσουν στον περιορισμό αυτού του προβλήματος του ΣΠ και να εξασφαλιστεί η ισορροπημένη γνώση και συμμετοχή.

Εφόσον το σύστημα θα καταγράφει στοιχεία για τις γνώσεις που έχει αποκτήσει κάθε μέλος της ομάδας, και θα γνωρίζει την απόδοσή του σε μία δραστηριότητα από το βαθμό που θα καταχωρεί ο καθηγητής, θα υπάρχει δυνατότητα η δημιουργία μιας ομάδας να γίνεται λαμβάνοντας υπόψη αυτά τα στοιχεία εφαρμόζοντας διάφορες στρατηγικές. Μία πετυχημένη συνεργασία εξαρτάται σε μεγάλο βαθμό από τη σύνθεση της ομάδας και αν τα μέλη θα ταιριάξουν μεταξύ τους. Συνήθως, η δημιουργία μιας ομάδας γίνεται είτε με βάση το επίπεδο γνώσεων των εκπαιδευόμενων, είτε με τυχαίους συνδυασμούς ή αφήνοντας τους εκπαιδευόμενους να επιλέξουν μόνοι τους συνεργάτες τους. Καθώς οι σχετικές έρευνες δεν έχουν καταλήξει ακόμη στον ιδανικότερο συνδυασμό των μελών σε μία ομάδα, με τα στοιχεία που θα καταγράφει το σύστημα θα μπορεί να γίνεται μία αποτίμηση της συνεργασίας και να εξάγονται χρήσιμα συμπεράσματα για τη σύνθεση της ομάδας. Ο καθηγητής θα έχει λοιπόν τη δυνατότητα να δοκιμάζει διάφορες στρατηγικές και να παρακολουθεί την απόδοση κάθε ομάδας.

Συμπεράσματα

Στην παρούσα εργασία παρουσιάσαμε τα οφέλη που μπορούν να αποκομίσουν αρχάριοι προγραμματιστές αν εντάξουν τη συνεργατική μάθηση στην ανάπτυξη λογισμικού. Τα συστήματα που αναπτύχθηκαν γι' αυτό το σκοπό υποστηρίζουν αφενός την εφαρμογή του συνεργατικού προγραμματισμού από απόσταση κι αφετέρου παρέχουν ένα μέσο το οποίο θα μπορούσε να αξιοποιηθεί για την ανάλυση και την ενίσχυση της συνεργασίας μεταξύ των προγραμματιστών. Οι στρατηγικές που προτείνονται στην τελευταία ενότητα της εργασίας στοχεύουν στη βελτίωση των συνθηκών της ομαδοσυνεργατικής μάθησης δίνοντας στους μαθητές την ευκαιρία για μία εποικοδομητική συνεργασία. Τέλος, με την εφαρμογή προσαρμοσμένων παρεμβάσεων στη συνεργατική ανάπτυξη κώδικα μπορεί να επιτευχθεί αποτελεσματικότερη εφαρμογή του συνεργατικού προγραμματισμού εξασφαλίζοντας ισορροπημένη μάθηση και ισορροπημένη συμμετοχή στους αρχάριους προγραμματιστές.

Βιβλιογραφία

Boyer, K. E., Dwight, A. A., Fondren, R. T., Vouk, M. A., Lester, J. C. (2008). A Development Environment for Distributed Synchronous Collaborative Programming. In *Proceedings of the 13th annual conference on Innovation and technology in computer science education,* ACM, New York, NY, USA, 158-162.

Cockburn, A., Williams, L. (2001). The costs and benefits of pair programming. In *Extreme programming examined*, Giancarlo Succi and Michele Marchesi (Eds.). Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA 223-243.

Diziol, D., Walker, E., Rummel, N., Koedinger, K. R. (2010). Using intelligent tutor technology to implement adaptive support for student collaboration. *Educational Psychology Review.*

Duque, R., Bravo, C. (2008). Analyzing Work Productivity and Program Quality in Collaborative Programming. In *Proceedings of the 2008 The Third International Conference on Software Engineering Advances (ICSEA '08)*. IEEE Computer Society, Washington, DC, USA, 270-276.

Gardner, W. (2003). Assessing Individual Contributions to Group Software Projects. *8th Western Canadian Conference on Computing Education (WCCCE '03),* Courtenay, BC, Canada, May 2003, pp. 33-50.

Granville, K., Hickey, T. J. (2005). The design, implementation, and application of the grewpEdit tool. *Proceedings of the 2005 conference on Diversity in computing*, Albuquerque, New Mexico, USA.

Hanks, B. (2008). Empirical evaluation of distributed pair programming. *Int. J. Hum.-Comput. Stud*. 66, 7 (July 2008), 530-544.

Hanks, B. (2007). Problems encountered by novice pair programmers. In *Proceedings of the third international workshop on Computing education research (ICER '07)*. ACM, New York, NY, USA, 159-164.

Jun, S., Kim, S., Lee, W. (2007). Online Pair-Programming for Learning Programming of Novices. *WSEAS TRANSACTIONS on ADVANCES in ENGINEERING EDUCATION*, Issue 9, Volume 4, September 2007.

Lister, R., Adams, E. S., Fitzgerald, S., Fone, W., Hamer, J., Lindholm, M., McCartney, R., Moström, J. E., Sanders, K., Seppälä, O., Simon, B., Thomas, L. (2004). A multi-national study of reading and tracing skills in novice programmers. In *Working group reports from ITiCSE on Innovation and technology in computer science education (ITiCSE-WGR '04)*. ACM, New York, NY, USA, 119-150.

McDowell, C., Hanks, B., Werner, L. (2003a). Experimenting with pair programming in the classroom. In *Proceedings of the 8th annual conference on Innovation and technology in computer science education (ITiCSE '03)*, David Finkel (Ed.). ACM, New York, NY, USA, 60-64.

McDowell, C., Werner, L., Bullock, H. E., Fernald, J. (2003b). The impact of pair programming on student performance, perception and persistence. In *Proceedings of the 25th International Conference on Software Engineering (ICSE '03)*. IEEE Computer Society, Washington, DC, USA, 602-607.

Preston, D. (2005). Pair programming as a model of collaborative learning: A review of the research. *Consortium for Computing Sciences in Colleges*, 39-45.

Sanjay, G., Vanshi, K. (2010). A Novel Approach for Collaborative Pair Programming. *Journal of Information Technology Education*, USA, Vol. 9, 183-196.

Schümmer, T., Lukosch, S. (2009). Understanding Tools and Practices for Distributed Pair Programming. *Journal of Universal Computer Science*, vol. 15, no. 16, 3101-3125.

Williams, L., Kessler, R., Cunningham, W., Jeffries, R. (2000). Strengthening the Case for Pair Programming. IEEE Software, pp. 19-25.

Williams, L., Yang, K., Wiebe, E., Ferzli, M., Miller, C. (2002). Pair Programming in an Introductory Computer Science Course: Initial Results and Recommendations. *OOPSLA Educator's Symposium*, pages 20-26.

Winkler, D., Biffl, S., Kaltenbach, A. (2010). Evaluating Tools that Support Pair Programming in a Distributed Engineering Environment. *Conference on Evaluation and Assessment in Software Engineering (EASE)*, Keele, Great Britain.

Zacharis, N. (2009). Evaluating the Effects of Virtual Pair Programming on Students' Achievement and Satisfaction. *International Journal Of Emerging Technologies In Learning (IJET)*, 4(3).