ePublishing

ΕΘΝΙΚΟ ΚΕΝΤΡΟ ΤΕΚΜΗΡΙΩΣΗΣ
EKT NATIONAL DOCUMENTATION CENTRE

Συνέδρια της Ελληνικής Επιστημονικής Ένωσης Τεχνολογιών Πληροφορίας & Επικοινωνιών στην Εκπαίδευση

Τόμ. 1 (2011)

2ο Πανελλήνιο Συνέδριο: «Ένταξη και χρήση των ΤΠΕ στην Εκπαιδευτική Διαδικασία»

Χρήση εργαλείων δημιουργίας διαγραμμάτων ροής για την εισαγωγή στην Αλγοριθμική και τον Προγραμματισμό: μια επισκόπηση

Σ. Ξυνόγαλος, Κ. Λαμπροπούλου

Βιβλιογραφική αναφορά:

Ξυνόγαλος Σ., & Λαμπροπούλου Κ. (2023). Χρήση εργαλείων δημιουργίας διαγραμμάτων ροής για την εισαγωγή στην Αλγοριθμική και τον Προγραμματισμό: μια επισκόπηση . *Συνέδρια της Ελληνικής Επιστημονικής Ένωσης Τεχνολογιών Πληροφορίας & Επικοινωνιών στην Εκπαίδευση*, *1*, 0611–0622. ανακτήθηκε από https://eproceedings.epublishing.ekt.gr/index.php/cetpe/article/view/4816

Χρήση εργαλείων δημιουργίας διαγραμμάτων ροής για την εισαγωγή στην Αλγοριθμική και τον Προγραμματισμό: μια επισκόπηση

Σ. Ξυνόγαλος, Κ. Λαμπροπούλου

Τμήμα Διοίκησης Τεχνολογίας, Πανεπιστήμιο Μακεδονίας
{stelios, tm083}@uom.gr

Περίληψη

Στην παρούσα εργασία γίνεται μια επισκόπηση περιβαλλόντων προγραμματισμού που ενσωματώνουν εκδότες, στους οποίους το παραγόμενο έγγραφο είναι κάποιου είδους διαγραμμάτος ροής. Τα εργαλεία που αναλύονται έχουν ως στόχο τη στήριξη των σπουδαστών στην απόκτηση δεξιοτήτων επίλυσης προβλημάτων και μπορούν να χρησιμοποιηθούν στα πλαίσια της διδασκαλίας και εκμάθησης εννοιών τόσο της αλγοριθμικής όσο και του προγραμματισμού.

Λέξεις κλειδιά: *αλγοριθμική, διάγραμμα ροής, απεικονιστική γλώσσα προγραμματισμού.*

1. Εισαγωγή

Η εξοικείωση με βασικές έννοιες της αλγοριθμικής και του προγραμματισμού είναι γνωστό ότι συνοδεύεται από ποικίλες δυσκολίες (Crews & Ziegler, 1998). Στήριξη στην αντιμετώπιση των δυσκολιών αυτών παρέχουν τα περιβάλλοντα προγραμματισμού που ενσωματώνουν εκδότες σύνταξης (syntax editors), εκδότες δηλαδή στους οποίους η ανάπτυξη των προγραμμάτων γίνεται με τη χρήση προτύπων και την επιλογή από μενού συντακτικά σωστών εντολών για κάθε ημιτελές τμήμα ενός προγράμματος (Ξυνόγαλος & Σατρατζέμη, 2002). Το βασικότερο πλεονέκτημα που παρέχουν αυτού του είδους τα περιβάλλοντα είναι η αντιμετώπιση των δυσκολιών που οφείλονται στις συντακτικές λεπτομέρειες μιας γλώσσας προγραμματισμού και η επικέντρωση της προσοχής των σπουδαστών στη διαδικασία επίλυσης προβλημάτων (Calloni & Bagert, 1994; Ziegler & Crews, 1999). Ανάλογα με τον τρόπο αναπαράστασης που χρησιμοποιείται σε ένα εκδότη σύνταξης, γίνεται διάκριση σε εκδότες δομής (structure editor) που βασίζονται σε κειμενικές αναπαραστάσεις και σε απεικονιστικούς εκδότες (iconic editors) ή απεικονιστικές γλώσσες προγραμματισμού (iconic programming language) που βασίζονται σε μια αναπαράσταση των εντολών με τη μορφή εικονιδίων.

Στην παρούσα εργασία γίνεται μια επισκόπηση περιβαλλόντων που ενσωματώνουν απεικονιστικούς εκδότες, στους οποίους το παραγόμενο έγγραφο είναι κάποιου είδους διαγραμμάτος ροής. Τα εργαλεία που παρουσιάζονται μπορούν να χρησιμοποιηθούν τόσο για τη στήριξη της διδασκαλίας και εκμάθησης εννοιών της αλγοριθμικής όσο και του προγραμματισμού.

2. Εργαλεία δημιουργίας διαγραμμάτων ροής

Στις υποενότητες που ακολουθούν αναλύονται οκτώ εργαλεία δημιουργίας διαγραμμάτων ροής. Από τα εργαλεία αυτά διατίθενται ελεύθερα τα εξής: RAPTOR, SFC editor, Δημιουργός Διαγραμμάτων Ροής, Διάγραμμα Ροής.

2.1 RAPTOR (http://raptor.martincarlisle.com/)

Το RAPTOR (Carlisle et al., 2005) είναι ένα οπτικό προγραμματιστικό περιβάλλον, το οποίο έχει ως στόχο τη στήριξη των σπουδαστών στη δημιουργία οπτικών αναπαραστάσεων των αλγορίθμων τους.

Ανάπτυξη αλγορίθμου. Η ανάπτυξη αλγορίθμων στο RAPTOR πραγματοποιείται επιλέγοντας το κατάλληλο σύμβολο από την παλέτα που υπάρχει στο επάνω αριστερό τμήμα του παραθύρου και εισάγοντάς το (με απλό κλικ) στο κατάλληλο σημείο του υπό ανάπτυξη αλγορίθμου. Η βασική παλέτα περιλαμβάνει σύμβολα για την εισαγωγή εντολής εισόδου/εξόδου, εντολής ανάθεσης τιμής, δομής επιλογής, δομής επανάληψης, κλήσης διαδικασίας και εισαγωγή σχολίων. Επίσης, το RAPTOR παρέχει τη δυνατότητα χρήσης μονοδιάστατων και διδιάστατων πινάκων, αρχείων, αλφαριθμητικών και αλληλεπίδρασης με το χρήστη μέσω μιας βιβλιοθήκης γραφικών. Τέλος, υπάρχουν ενσωματωμένες συναρτήσεις και διαδικασίες για την παραγωγή τυχαίων αριθμών, την πραγματοποίηση τριγωνομετρικών υπολογισμών, τη σχεδίαση γραφικών, καθώς επίσης και χρονικές συναρτήσεις. Οι αλγόριθμοι, ή αλλιώς τα προγράμματα όπως τα αποκαλούν οι δημιουργοί του περιβάλλοντος, που αναπτύσσονται παραμένουν πάντα δομημένα και συντακτικά σωστά, αφού ο έλεγχος λαθών πραγματοποιείται κατά την ανάπτυξη.

Εκτέλεση αλγορίθμου. Το RAPTOR παρέχει τη δυνατότητα εκτέλεσης των αλγορίθμων βήμα προς βήμα ή ανίχνευσης της εκτέλεσης τους με ταχύτητα που επιλέγεται από τον χρήστη. Κατά τη βηματική εκτέλεση το περιβάλλον επισημαίνει στο διάγραμμα ροής το σημείο στο οποίο βρίσκεται η εκτέλεση του αλγορίθμου, καθώς επίσης και το περιεχόμενο των μεταβλητών.

Παραγωγή κώδικα. Το RAPTOR παράγει αυτόματα πηγαίο κώδικα σε Ada, C#, C++ και Java από ένα υπάρχον διάγραμμα ροής. Επιπλέον, υπάρχει δυνατότητα δημιουργίας γεννητριών πηγαίου κώδικα και σε άλλες γλώσσες υλοποιώντας μια κλάση σε C# (http://raptor.martincarlisle.com/code_generator.htm).

Διαθέσιμο υλικό. Στην ιστοσελίδα του RAPTOR διατίθενται ελεύθερα σημειώσεις για διάφορα θέματα που σχετίζονται με την αξιοποίηση του εργαλείου, όπως εισαγωγή στην αλγοριθμική σκέψη ή/και στον προγραμματισμό με το RAPTOR, χρήση δομών ελέγχου και μεταβλητών πίνακα, και αποσύνθεση ενός προβλήματος σε συναρτήσεις.

Επιπλέον δυνατότητες. Ο διδάσκοντας έχει τη δυνατότητα να συμπεριλάβει στο περιβάλλον επιπρόσθετες διαδικασίες/συναρτήσεις, τις οποίες οι σπουδαστές μπορούν να χρησιμοποιούν με τον ίδιο τρόπο που χρησιμοποιούν τις ενσωματωμένες συναρτήσεις. Αν και η υλοποίηση της σχετικής βιβλιοθήκης (plugin*.dll) μπορεί να

γίνει θεωρητικά σε οποιαδήποτε .NET γλώσσα, έχει χρησιμοποιηθεί και ελεγχθεί μόνο η γλώσσα C# (http://raptor.martincarlisle.com/creating_plugins.htm).

Αξιολόγηση – στοιχεία χρήσης. Το RAPTOR χρησιμοποιείται, σύμφωνα με τους δημιουργούς του, σε 17 χώρες για την εκπαίδευση σε θέματα της Επιστήμης των Υπολογιστών. Ωστόσο, στη βιβλιογραφία εντοπίσαμε μόνο τις μελέτες που αναφέρονται παρακάτω, στις οποίες συμμετείχε ο δημιουργός του περιβάλλοντος:

- Το RAPTOR χρησιμοποιήθηκε για τρία συνεχόμενα εξάμηνα κατά τα ακαδημαϊκά έτη 2003-2004 στα πλαίσια διδασκαλίας ενός εισαγωγικού μαθήματος Πληροφορικής στο Τμήμα Επιστήμης Υπολογιστών της Στρατιωτικής Ακαδημίας Αεροπορίας των Ηνωμένων Πολιτειών (Carlisle et al., 2004; 2005). Τα συγκριτικά αποτελέσματα από τη διδασκαλία του μαθήματος για τρία συνεχόμενα εξάμηνα έδειξαν ότι το RAPTOR βοηθάει περισσότερο στην ανάπτυξη δεξιοτήτων επίλυσης προβλημάτων σε σχέση με τη διδασκαλία με μη οπτικές γλώσσες. Εξαίρεση αποτελεί η χρήση πινάκων, όπου στη διδασκαλία του μαθήματος για πρώτη φορά με το RAPTOR (φθινόπωρο 2003) οι φοιτητές παρουσίασαν χειρότερη επίδοση σε ένα ερώτημα των εξετάσεων που απαιτούσε τη χρήση πινάκων. Επίσης, στο πρώτο εξάμηνο που υπήρχε δυνατότητα έκφρασης των αλγορίθμων σε διάγραμμα ροής, Ada ή MATLAB, το 95% σχεδόν των φοιτητών προτίμησε τη χρήση διαγραμμάτων ροής.

- Το RAPTOR χρησιμοποιήθηκε το 2005 στη Στρατιωτική Ακαδημία των Ηνωμένων Πολιτειών στα πλαίσια ενός εισαγωγικού μαθήματος που επικεντρώνεται στη σχεδίαση αλγορίθμων και την υλοποίησή τους σε Java (Giordano & Carlisle, 2006). Από τους φοιτητές που παρακολούθησαν το μάθημα, οι 54 χρησιμοποίησαν το RAPTOR και οι 18 ένα εμπορικό εργαλείο δημιουργίας διαγραμμάτων. Οι φοιτητές που χρησιμοποίησαν το RAPTOR είχαν καλύτερη επίδοση στις τελικές εξετάσεις, αν και το δείγμα δεν ήταν επαρκές για να προκύψουν στατιστικά σημαντικά αποτελέσματα. Οι φοιτητές της ομάδας του RAPTOR ανέφεραν ότι προτιμούν το συγκεκριμένο εργαλείο σε σχέση με το εμπορικό εργαλείο που κλήθηκαν να χρησιμοποιήσουν σε κάποια εργασία του μαθήματος, καθώς επίσης και ότι το RAPTOR τους βοήθησε να αναπτύξουν προγράμματα σε Java.

2.2 BACCII – BACCII++

Τα περιβάλλοντα BACII και BACCII++ (Calloni & Bagert, 1994; 1997) αναπτύχθηκαν στο Texas Tech University, έχοντας ως στόχο τη στήριξη της διδασκαλίας του διαδικαστικού και του αντικειμενοστρεφούς προγραμματισμού.

Ανάπτυξη αλγορίθμου. Για την καθοδήγηση των σπουδαστών κατά την ανάπτυξη αλγορίθμων, ή προγραμμάτων όπως τα χαρακτηρίζουν οι δημιουργοί του BACCII++, το περιβάλλον παρέχει ένα συντακτικά καθοδηγούμενο ενδιάμεσο (syntax-directed interface) και μια παλέτα με τα εικονίδια των διαθέσιμων εντολών. Το ενδιάμεσο που χρησιμοποιείται για την ανάπτυξη των διαγραμμάτων, καθοδηγούμενο από τους συντακτικούς κανόνες της γλώσσας επιτρέπει την εισαγωγή μόνο σωστών συνδυασμών εντολών (εικόνων) στο υπό ανάπτυξη διάγραμμα ροής, δημιουργεί τις απαραίτητες συνδέσεις και επανασχεδιάζει το διάγραμμα αυτόματα. Το περιβάλλον

παρέχει τη δυνατότητα δήλωσης μεταβλητών όλων των βασικών τύπων, χρήσης εντολών ανάθεσης, ελέγχου και επανάληψης, τη δήλωση και κλήση υποπρογραμμάτων, τη χρήση αρχείων για διάβασμα ή/και εγγραφή δεδομένων και την είσοδο δεδομένων από το πληκτρολόγιο. Τα χρησιμοποιούμενα εικονίδια έχουν το σχήμα ενός τετραγώνου με μια χαρακτηριστική εικόνα που υποδηλώνει την εντολή που αντιπροσωπεύει και δεν χρησιμοποιούνται τα καθιερωμένα σύμβολα αναπαράστασης των αλγορίθμων με διαγράμματα ροής.

Παραγωγή κώδικα. Μετά το τέλος της υλοποίησης ενός αλγορίθμου το περιβάλλον παρέχει τη δυνατότητα αυτόματης παραγωγής συντακτικά σωστού πηγαίου κώδικα σε Pascal, C, Fortran, Basic και C++.

Αξιολόγηση – στοιχεία χρήσης. Το BACII χρησιμοποιήθηκε στα πλαίσια ενός μαθήματος εισαγωγής στον προγραμματισμό (CS1), το οποίο βασίζονταν στο κατηγορηματικό παράδειγμα προγραμματισμού και την Pascal. Το BACCII++ χρησιμοποιήθηκε στα πλαίσια του παραπάνω μαθήματος χρησιμοποιώντας όμως την C++, καθώς επίσης και στα πλαίσια ενός μαθήματος δομών δεδομένων (CS2), το οποίο βασίζονταν στο αντικειμενοστρεφές παράδειγμα προγραμματισμού και την C++. Τα στατιστικά σημαντικά αποτελέσματα και τα συμπεράσματα συνοψίζονται ως εξής (Calloni & Bagert, 1994; 1997):

- Οι φοιτητές που χρησιμοποίησαν τα περιβάλλοντα BACII και BACCII++ παρουσίασαν υψηλότερη επίδοση τόσο στις εργασίες όσο και στις τελικές εξετάσεις.

- Η χρήση των BACCII και BACCII++ στα μαθήματα CS1 και CS2 έδειξε ότι το περιβάλλον είχε μεγαλύτερη επίδραση στο μάθημα CS2, το οποίο βασίζονταν στο αντικειμενοστρεφές παράδειγμα προγραμματισμού. Επιπλέον, οι βαθμοί των σπουδαστών παρουσίασαν μικρότερη απόκλιση από το μέσο βαθμό σε όλες τις κατηγορίες (προγράμματα, εργαστήρια, εξετάσεις), γεγονός που υποδηλώνει μια πιο ομοιόμορφη μάθηση.

- Το πιο απρόσμενο αποτέλεσμα ήταν η μεγαλύτερη κατανόηση, από την ομάδα των φοιτητών που χρησιμοποίησαν τα BACCII και BACCII++, της σύνταξης της Pascal και της C++ αντίστοιχα σε σχέση με τους φοιτητές που χρησιμοποίησαν συμβατικά περιβάλλοντα. Οι Calloni και Bagert απέδωσαν το αποτέλεσμα αυτό στο γεγονός ότι οι σπουδαστές έβλεπαν συνεχώς συντακτικά σωστά προγράμματα.

2.3 FLINT (FLowchart INTerpreter)

Το FLINT αναπτύχθηκε στο Τμήμα Επιστήμης Υπολογιστών του Πανεπιστημίου Western Kentucky με σκοπό να χρησιμοποιηθεί στο πρώτο μισό του μαθήματος CS1.

Ανάπτυξη αλγορίθμου. Για τη σχεδίαση μιας λύσης για ένα πρόβλημα οι φοιτητές ακολουθούν τη μέθοδο της βήμα-προς βήμα βελτίωσης του σχεδίου τους, το οποίο έχει τη μορφή ενός από πάνω προς τα κάτω δομημένου διαγράμματος. Ο καθορισμός ενός τέτοιου σχεδίου είναι υποχρεωτικός πριν από την ανάπτυξη αλγορίθμου για κάποιο από τα βήματα του σχεδίου. Όταν ο φοιτητής ξεκινήσει την ανάπτυξη του πρώτου αλγορίθμου, το βασικό σχέδιο λύσης του προβλήματος αποθηκεύεται

αυτόματα προκειμένου να αξιολογηθεί από τον καθηγητή. Για την αναπαράσταση των αλγορίθμων χρησιμοποιούνται δομημένα διαγράμματα ροής. Το FLINT επιτρέπει την εισαγωγή, διαγραφή, μεταφορά και αντιγραφή μόνο ολοκληρωμένων εντολών, με αποτέλεσμα το διάγραμμα να παραμένει πάντα δομημένο και συντακτικά σωστό. Οι διαθέσιμες δομές είναι η σειριακή, η επαναληπτική και η δομή επιλογής. Ο σχεδιασμός των διαγραμμάτων γίνεται μέσω ενός καταλόγου επιλογών, ενώ κάθε μεταβλητή πριν να χρησιμοποιηθεί πρέπει να δηλωθεί.

Εκτέλεση αλγορίθμου. Κατά την βηματική εκτέλεση, κάθε εντολή φωτίζεται πριν να εκτελεστεί, ενώ ταυτόχρονα φωτίζονται και οι μεταβλητές που σχετίζονται με τη συγκεκριμένη εντολή. Προκειμένου να ελεγχθεί η ορθότητα ενός προγράμματος ο καθηγητής ορίζει τον αριθμό των δοκιμαστικών εκτελέσεων του. Το FLINT πριν από κάθε δοκιμαστική εκτέλεση ενός προγράμματος "ζητάει" από το φοιτητή την είσοδο και την αναμενόμενη έξοδο, και αποθηκεύει αυτές τις πληροφορίες για χρήση από τον καθηγητή. Μετά το τέλος κάθε εκτέλεσης ο φοιτητής καλείται να εξετάσει αν η έξοδος ήταν η αναμενόμενη, προκειμένου να διαπιστώσει αν υπάρχουν λάθη.

Αξιολόγηση – στοιχεία χρήσης. Οι δημιουργοί του FLINT διεξήγαγαν ένα πείραμα με αντικείμενο τη σύγκριση των δομημένων διαγραμμάτων ροής και προγραμμάτων σε Qbasic (Crews & Ziegler, 1998). Τα συμπεράσματα του πειράματος συνοψίζονται ως εξής: (1) οι φοιτητές χρειάστηκαν λιγότερο χρόνο για την κατανόηση των δομημένων διαγραμμάτων ροής, (2) οι φοιτητές έκαναν λιγότερα λάθη, είχαν μεγαλύτερη αυτοπεποίθηση, είδαν τους αλγόριθμους λιγότερες φορές και χρειάστηκαν λιγότερο χρόνο για να απαντήσουν στις ερωτήσεις όταν χρησιμοποιούσαν διαγράμματα ροής, και (3) η αποτελεσματικότητα των διαγραμμάτων ροής αυξάνεται όσο αυξάνεται η πολυπλοκότητα των αλγορίθμων.

2.4 SFC Editor (http://watts.cs.sonoma.edu/SFC/)

Το Structure Flow Chart Editor (Tia Watts, 2001) είναι ένα εργαλείο για ανάπτυξη αλγορίθμων τόσο για αρχάριους, όσο και για έμπειρους προγραμματιστές.

Ανάπτυξη αλγορίθμου. Στο βασικό παράθυρο του SFC Editor παρουσιάζονται ταυτόχρονα το διάγραμμα ροής και ο αντίστοιχος ψευδοκώδικας που παράγεται αυτόματα. Κάνοντας διπλό κλικ στα σημεία εγκοπής που εμφανίζονται ανάμεσα στο σύμβολο αρχής και τέλους εμφανίζεται ένα πλαίσιο διαλόγου στο οποίο ο χρήστης επιλέγει το είδος της δομής που θέλει να προσθέσει στο συγκεκριμένο σημείο του διαγράμματος. Το περιβάλλον υποστηρίζει τις τρεις βασικές αλγοριθμικές δομές: ακολουθιακή, επιλογής και επανάληψης. Με την επιλογή κάθε δομής εμφανίζεται παράθυρο διαλόγου για την είσοδο των παραμέτρων καθώς και επιλογές μορφοποίησης της εμφάνισης του διαγράμματος. Με κάθε νέα εισαγωγή στο διάγραμμα ροής δημιουργείται αντίστοιχα κώδικας στο δεξιό πλαίσιο που όμως δε μπορεί να τροποποιηθεί. Επιπλέον παρέχονται δυνατότητες αποκοπής και αντιγραφής των στοιχείων του διαγράμματος και επικόλλησης στο ίδιο ή σε άλλο διάγραμμα, καθώς επίσης και τροποποίησης μιας δομής του διαγράμματος.

Παραγωγή κώδικα. Στο βασικό παράθυρο του SFC Editor παρουσιάζεται

ψευδοκώδικας σε μορφή που μοιάζει με C++ ή σε μια γενική Pascal-like μορφή.

Διαθέσιμο υλικό. Στην ιστοσελίδα του SFC Editor διατίθεται ένας αναλυτικός οδηγός χρήσης του εργαλείου με επεξηγήσεις και εικόνες για κάθε δομή.

Επιπλέον δυνατότητες. Στην έκδοση 2.3 προστέθηκε η δυνατότητα εξαγωγής: (1) του ψευδοκώδικα σε αρχείο κειμένου (.txt) προκειμένου να μπορέσει ο χρήστης να κάνει τις απαραίτητες αλλαγές, να μεταγλωττίσει και να εκτελέσει το πρόγραμμα σε κάποιο άλλο περιβάλλον, και (2) του διαγράμματος ροής σε ένα αρχείο Scalable Vector Graphic (.svg), μια μορφοποίηση γραφικών που έχει σχεδιαστεί από την Adobe Acrobat και βασίζεται σε XML. Τα αρχεία SVG μπορούν να ανοιχτούν με το Adobe SVG Viewer και να εισαχθούν σε ιστοσελίδες.

Αξιολόγηση – στοιχεία χρήσης. Η δημιουργός του SFC Editor και καθηγήτρια στο τμήμα Επιστήμης των Υπολογιστών στο Sonoma State University αναφέρει ότι το συγκεκριμένο εργαλείο έχει χρησιμοποιηθεί τόσο σε εισαγωγικά όσο και σε προχωρημένα μαθήματα. Στο (Tia Watts, 2001) αναφέρονται κάποια στοιχεία χρήσης του SFC Editor στα πλαίσια ενός εισαγωγικού μαθήματος με βασικό στόχο την εισαγωγή φοιτητών στη σχεδίαση αλγορίθμων. Μετά την εκπόνηση μιας σειράς εργασιών, στα πλαίσια των οποίων σχεδιάστηκαν διαγράμματα ροής με το Word ή το WordPerfect και το SFC Editor, δόθηκε στους φοιτητές ένα ερωτηματολόγιο βαθμολόγησης των λογισμικών από το 0 (εύκολο στην χρήση) μέχρι το 10 (δύσκολο). Ο μέσος όρος για το SFC Editor ήταν 2.5, ενώ για το MS-Word 4.7. Αναφορά γίνεται επίσης στη θετική στάση απέναντι στο SFC Editor από φοιτητές στο Indiana University of Pennsylvania, μετά από τη χρήση του για την ανάπτυξη διαγραμμάτων ροής και ψευδοκώδικα στα πλαίσια ενός μεγάλου project υλοποίησης τμημάτων ενός λειτουργικού συστήματος στα πλαίσια σχετικού μαθήματος.

2.5 Δημιουργός Διαγραμμάτων Ροής (http://www.ecedu.upatras.gr/flowchart/)

Ανάπτυξη αλγορίθμου. Ο χρήστης αναπτύσσει ένα διάγραμμα ροής επιλέγοντας την κατάλληλη δομή από την εργαλειοθήκη και τοποθετώντας την στο διάγραμμα με την τεχνική σύρε και άφησε. Τα αντικείμενα που υπάρχουν στην εργαλειοθήκη είναι τα σύμβολα αρχή και τέλος, η εντολή ανάθεσης, η εντολή εισόδου/εξόδου και η συνθήκη. Η σύνδεση των αντικειμένων στο διάγραμμα γίνεται με γραμμές που πρέπει να τοποθετηθούν από το χρήστη. Οι επαναληπτικές δομές σχεδιάζονται από τον χρήστη επιλέγοντας τα κατάλληλα αντικείμενα και γραμμές.

Εκτέλεση αλγορίθμου. Ο Δημιουργός Διαγραμμάτων Ροής επιτρέπει την εκτέλεση αλγορίθμων με ταχύτητα που επιλέγεται από τον χρήστη. Κατά τον αργό ρυθμό εκτέλεσης επισημαίνεται το σημείο που εκτελείται, ενώ παρουσιάζεται ταυτόχρονα το περιεχόμενο των μεταβλητών και της στοίβας.

Παραγωγή κώδικα. Ο Δημιουργός Διαγραμμάτων Ροής δεν υποστηρίζει την δημιουργία πηγαίου κώδικα, αλλά δημιουργεί εκτελέσιμο αρχείο για τον αλγόριθμο.

Διαθέσιμο υλικό. Στην ιστοσελίδα του εργαλείου παρέχονται πληροφορίες σχετικά με τις δυνατότητες και τον τρόπο χρήσης του, καθώς επίσης και σύνδεσμοι για διάφορες

πτυχές του έργου "Αλγοριθμική", στα πλαίσια του οποίου δημιουργήθηκε.

Επιπλέον δυνατότητες. Ο Δημιουργός Διαγραμμάτων Ροής παρέχει τη δυνατότητα αποθήκευσης ενός αλγορίθμου και χρήσης του είτε ως αυτόνομο αλγόριθμο είτε ως συνάρτηση, η οποία μπορεί στη συνέχεια να κληθεί από οποιοδήποτε άλλο αλγόριθμο ή ακόμα και από τον εαυτό του (αναδρομικότητα).

Αξιολόγηση – στοιχεία χρήσης. Ο Δημιουργός Διαγραμμάτων Ροής χρησιμοποιήθηκε στα πλαίσια μελέτης της επίδρασης δύο εναλλακτικών αναπαραστάσεων αλγορίθμων, των διαγραμμάτων ροής και του ψευδοκώδικα σε εισαγωγικές έννοιες του προγραμματισμού (Σιώζου, Τσέλιος & Κόμης, 2008). Στην έρευνα συμμετείχαν 38 φοιτητές, οι οποίοι χωρίστηκαν σε 2 ομάδες και μετά από την παρουσίαση των βασικών εννοιών του προγραμματισμού, έλυσαν ένα πρόβλημα με πέντε υπό-ερωτήματα κλιμακούμενης δυσκολίας. Η πρώτη ομάδα χρησιμοποίησε τον Δημιουργό Διαγραμμάτων Ροής, ενώ η δεύτερη ομάδα τον Διερμηνευτής της Γλώσσας, ο οποίος βασίζεται στη χρήση ψευδοκώδικα. Από την ανάλυση των δεδομένων που συγκεντρώθηκαν δεν διαπιστώθηκε στατιστικά σημαντική διαφορά στην επίδοση των φοιτητών σε συνάρτηση με το εργαλείο που χρησιμοποίησαν.

2.6 Διάγραμμα Ροής (http://users.sch.gr/vraa8/index.php?option=com_content&view=article&id=2&Itemid=4)

Το Διάγραμμα Ροής προτείνεται ως ένα διδακτικό εργαλείο για τη διδασκαλία των μαθημάτων «Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον» της Γ' τάξης του Ενιαίου Λυκείου και «Προγραμματισμός Η/Υ» στο ΕΠΑΛ (Βρακόπουλος & Καρτσιώτης, 2008).

Ανάπτυξη αλγορίθμου. Το εργαλείο Διάγραμμα Ροής υποστηρίζει όλες τις αλγοριθμικές δομές του δομημένου προγραμματισμού που χρησιμοποιούνται για τη σχεδίαση διαγραμμάτων ροής. Ο μαθητής επιλέγει μία δομή, τη μεταφέρει με την τεχνική σύρε και άφησε στην επιφάνεια δημιουργίας του διαγράμματος, εισάγει τις απαραίτητες πληροφορίες στα πλαίσια διαλόγου που εμφανίζονται και δημιουργεί τις απαραίτητες συνδέσεις. Το πρόγραμμα υποστηρίζει μεταβλητές, μονοδιάστατους και διδιάστατους πίνακες, μαθηματικές συναρτήσεις, και εισαγωγή σχολίων για τον αλγόριθμο. Επίσης, για λόγους ευχρηστίας παρέχονται έτοιμες δομές, όπως η «Όσο».

Εκτέλεση αλγορίθμου. Υπάρχει δυνατότητα βηματικής εκτέλεσης των αλγορίθμων, με ταυτόχρονη παρακολούθηση των περιεχομένων των μεταβλητών και καταγραφή των οθονών ως αρχεία bmp.

Παραγωγή κώδικα. Το Διάγραμμα Ροής παρέχει τη δυνατότητα αυτόματης παραγωγής κώδικα σε ψευδογλώσσα και στη συνέχεια σε Basic ή Pascal.

Διαθέσιμο υλικό. Στην ιστοσελίδα του εργαλείου υπάρχουν πληροφορίες χρήσης και παραδείγματα.

Αξιολόγηση – στοιχεία χρήσης. Οι Βρακόπουλος και Καρτσιώτης (2008) αναφέρουν ότι «*η αξιοποίηση του προγράμματος Διάγραμμα Ροής βοήθησε τους μαθητές να*

κατανοήσουν καλύτερα τις έννοιες της επανάληψης, της επιλογής και τη συνθήκη τερματισμού μιας επανάληψης, καθώς επίσης και να εμπεδώσουν τη σχέση αλγόριθμου – διαγράμματος ροής – ψευδογλώσσας – κώδικα». Ωστόσο, δεν γίνεται αναφορά στις συνθήκες διδασκαλίας ή/και τα δεδομένα της σχετικής έρευνας.

2.7 Άλλα εργαλεία

SICAS. Το SICAS (Marcelino et al., 2004) σχεδιάστηκε έχοντας ως στόχο τη στήριξη των σπουδαστών στην εκμάθηση βασικών εννοιών του διαδικαστικού προγραμματισμού. Τα διαγράμματα ροής αναπτύσσονται επιλέγοντας την κατάλληλη εντολή από τη γραμμή εργαλείων και στη συνέχεια το σημείο στην περιοχή σχεδίασης όπου θα τοποθετηθεί αυτή. Το περιβάλλον υποστηρίζει τη χρήση εντολών ανάθεσης, εισόδου/εξόδου, επιλογής και επανάληψης, αριθμητικών και αλφαριθμητικών μεταβλητών, πινάκων ακεραίων ή αλφαριθμητικών τιμών και συναρτήσεων. Μετά την ολοκλήρωση ενός αλγορίθμου υπάρχει η δυνατότητα βηματικής εκτέλεσης με ταυτόχρονη παρακολούθηση των μεταβολών τιμής των μεταβλητών, καθώς και αυτόματης παραγωγής ψευδοκώδικα και κώδικα σε C ή Java.

B#. Το B# (Cilliers, Calitz & Greyling, 2005) σχεδιάστηκε στο Nelson Mandela Metropolitan University έχοντας ως στόχο να στηρίξει τα αρχικά στάδια ενός μαθήματος CS1. Το B# ενσωματώνει μια απεικονιστική γλώσσα και ένα ολοκληρωμένο περιβάλλον προγραμματισμού. Στο διάγραμμα ροής χρησιμοποιούνται εικονίδια, όπως στα BACCII/BACCII++, που διαφέρουν από τα καθιερωμένα σύμβολα των διαγραμμάτων ροής. Ταυτόχρονα με την ανάπτυξη ενός διαγράμματος παρουσιάζεται συντακτικά σωστός πηγαίος κώδικας σε Borland Pascal, ενώ υπάρχει δυνατότητα ανίχνευσης της εκτέλεσης του διαγράμματος ροής και του προγράμματος.

3. Συμπεράσματα

Στην παρούσα εργασία παρουσιάστηκαν οκτώ εργαλεία δημιουργίας διαγραμμάτων ροής, τα οποία χρησιμοποιούνται στα πλαίσια διδασκαλίας της αλγοριθμικής και του προγραμματισμού. Από τη συγκριτική ανάλυση των εργαλείων προκύπτουν τα εξής:

- Σε όλα τα εργαλεία ο αλγόριθμος αναπτύσσεται με τη μορφή ενός *διαγράμματος ροής*. Σε ορισμένες περιπτώσεις, ωστόσο, δεν χρησιμοποιούνται τα καθιερωμένα σύμβολα, αλλά σύμβολα με το ίδιο σχήμα και χαρακτηριστικά εικονίδια. Στη δεύτερη περίπτωση, οι δημιουργοί των εργαλείων (BACCII/BACCII++, B#) χαρακτηρίζουν τη χρησιμοποιούμενη σημειογραφία ως *απεικονιστική γλώσσα προγραμματισμού*. Οι απεικονιστικές γλώσσες προγραμματισμού έχουν αυξημένο ρεπερτόριο εντολών, υποστηρίζουν τη δήλωση μεταβλητών, τη διαχείριση αρχείων και γενικότερα δυνατότητες που δεν αποτυπώνονται συνήθως στα καθιερωμένα διαγράμματα ροής.

- Η πλειονότητα των εργαλείων παρέχει τη δυνατότητα εκτέλεσης, ή πιο σωστά τη δυνατότητα προσομοίωσης της εκτέλεσης, των αλγορίθμων που αναπτύσσονται με τη μορφή διαγραμμάτων ροής. Εξαίρεση αποτελούν τα εργαλεία BACCII/BACCII++

και SFC. Η δυνατότητα εκτέλεσης είναι ιδιαίτερα σημαντική, αφού στηρίζει τους σπουδαστές στην κατανόηση της σημασίας των δομών που προκαλούν δυσκολίες και της ροής εκτέλεσης, καθώς και στον εντοπισμό λογικών λαθών.

- Μια επιπλέον δυνατότητα που παρέχουν τα σύγχρονα εργαλεία δημιουργίας διαγραμμάτων ροής είναι η αυτόματη παραγωγή πηγαίου κώδικα σε διάφορες γλώσσες προγραμματισμού, στην πλειονότητά τους διαδικαστικές. Ο κώδικας αυτός σε άλλες περιπτώσεις χρειάζεται εκτεταμένες τροποποιήσεις προκειμένου να μεταγλωττιστεί σε μια γλώσσα προγραμματισμού (για παράδειγμα στο SFC editor) και σε άλλες όχι (για παράδειγμα στα BACCII/BACCII++). Η αυτόματη παραγωγή συντακτικά σωστού πηγαίου κώδικα μπορεί να στηρίξει τους αρχάριους προγραμματιστές στην κατανόηση της σχέσης μεταξύ των ενεργειών που έχουν περιγράψει με οπτικό τρόπο σε ένα διάγραμμα ροής και της υλοποίησής τους σε μια γλώσσα προγραμματισμού. Οι αρχάριοι προγραμματιστές έχουν τη δυνατότητα να μελετήσουν και να εκτελέσουν το πρόγραμμα που αναπαριστά τον αλγόριθμο που ανέπτυξαν, χωρίς να χρειάζεται από την αρχή να αναλωθούν στη δύσκολη διαδικασία της αποσφαλάτωσης. Επίσης, η δυνατότητα αυτόματης παραγωγής πηγαίου κώδικα σε διάφορες γλώσσες παρουσιάζει με τον καλύτερο τρόπο το γεγονός ότι *ένας καλά σχεδιασμένος αλγόριθμος μπορεί εύκολα να μεταφραστεί σε αρκετές γλώσσες προγραμματισμού – ο πιο σημαντικός παράγοντας στη σχεδίαση ενός αλγορίθμου είναι η σύλληψή του και όχι η γλώσσα προγραμματισμού στην οποία θα υλοποιηθεί*.

- Τα στοιχεία αξιολόγησης που παρουσιάζονται προέρχονται από μελέτες που διεξήγαγαν, ή στις οποίες συμμετείχαν, οι δημιουργοί των εργαλείων. Δυστυχώς, δεν καταφέραμε να εντοπίσουμε μελέτες ανεξάρτητων ερευνητών, ακόμα και για το RAPTOR που σύμφωνα με τους δημιουργούς του χρησιμοποιείται σε 17 χώρες. Αυτό βέβαια μπορεί να οφείλεται στο γεγονός ότι τα εργαλεία που διατίθενται ελεύθερα δημιουργήθηκαν τα τελευταία χρόνια, με εξαίρεση το SFC. Επιπλέον, στις μελέτες που διεξήχθησαν καταγράφονται αντικρουόμενα αποτελέσματα. Σε κάποιες διαπιστώνεται μια υπεροχή των διαγραμμάτων ροής έναντι άλλων μορφών αναπαράστασης αλγορίθμων (Carlisle et al., 2004; 2005), ενώ σε άλλες όχι (Σιώζου, Τσέλιος & Κόμης, 2008). Είναι προφανές ότι απαιτείται περαιτέρω έρευνα προκειμένου να διερευνηθεί η εκπαιδευτική αξία των περιβαλλόντων δημιουργίας διαγραμμάτων ροής ως εργαλείων για την εισαγωγή στον προγραμματισμό.

Τα εργαλεία που παρουσιάστηκαν έχουν αρκετές ομοιότητες, αλλά σε ορισμένες περιπτώσεις και ιδιαίτερα χαρακτηριστικά που τα διακρίνουν. Ο συνδυασμός τους θεωρούμε ότι θα είχε ως αποτέλεσμα ένα ακόμα πιο ισχυρό εργαλείο. Ιδανικά, ένα περιβάλλον για την εισαγωγή σε βασικές έννοιες της αλγοριθμικής και του προγραμματισμού θα μπορούσε να ενσωματώνει: (1) τη δυνατότητα *σχεδίασης της λύσης ενός προβλήματος με τη μορφή ενός από πάνω προς τα κάτω δομημένου διαγράμματος*, η οποία όμως - σε αντίθεση με το FLINT - θα είναι προαιρετική, (2) τη *χρήση πολλαπλών αναπαραστάσεων - ψευδοκώδικα, διαγράμματος ροής, προγράμματος* - για την έκφραση της λύσης σε ένα πρόβλημα, (3) τη χρήση ενός *συντακτικά καθοδηγούμενου ενδιάμεσου για την ανάπτυξη του αλγορίθμου* ανεξάρτητα

από τη μορφή αναπαράστασής του, (4) τη δυνατότητα *αυτόματης παραγωγής ενός αλγορίθμου σε άλλη μορφή αναπαράστασης από εκείνη στην οποία σχεδιάστηκε*, (5) τη δυνατότητα της *βήμα προς βήμα εκτέλεσης* κάθε μορφής αναπαράστασης ενός αλγορίθμου με τον ίδιο ακριβώς τρόπο.

Βιβλιογραφία

Calloni, B. A. & Bagert, D. J. (1997). Iconic Programming Proves Effective for Teaching the First Year Programming Sequence, *ACM, SIGSCE '97,* pp.262-266.

Calloni, B. & Bagert, D. (1994). Iconic Programming in BACCII vs. Textual Programming: which is a better learning environment?, *ACM, SIGSCE* '94, pp. 188-192

Carlisle, M., Wilson, T., Humphries, J. & Hadfield, S. (2004). RAPTOR: Introducing programming to non-majors with flowcharts, *Journal of Computing Sciences in Colleges* , Volume 19, Issue 4, pp. 52-60.

Carlisle, M., Wilson, T., Humphries, J. & Hadfield, S. (2005). RAPTOR: A Visual Programming Environment for Teaching Algorithmic Problem Solving. *ACM SIGCSE '05*, pp. 176-180.

Cilliers A. & Calitz A. & Greyling J. (2004). The effect of integrating an Iconic Programming Notation into CS1. *ACM ITiCSE 2005,* 108-112.

Crews, T. & Ziegler, U. (1998). The Flowchart Interpreter for Introductory Programming Courses, *In Proceedings of FIE '98 Conference*, pp. 307-312.

Giordano, J. & Carlisle, M. (2006). Toward a more effective visualization tool to teach novice programmers. *Proceedings of the 7th Conference on Information Technology Education*, pp. 115-122.

Marcelino, M., Gomes, A., Dimitrov, N. & Mendes, A. (2004). Using a computer-based interactive system for the development of basic algorithmic and programming skills. *In Proceedings of the 5th International Conference on Computer Systems and Technologies.* ACM, New York, NY, USA, pp. 1-6.

Watts, T. (2004). The SFC Editor a graphical tool for algorithm development. *Journal of Computing Sciences in Colleges*, vol. 20, Issue 2, pp. 74-85.

Ziegler, U. & Crews, T. (1999). An Integrated Program Development Tool for Teaching and Learning How to Program, *ACM, SIGSCE '99,* pp. 276-280.

Βρακόπουλος Α. & Καρτσιώτης Θ. (2008). Το πρόγραμμα «Διάγραμμα Ροής» και η διδακτική του αξιοποίηση στην Διδασκαλία του προγραμματισμού. *Πρακτικά 4ου Πανελλήνιου Συνεδρίου Διδακτική της Πληροφορικής*, 55-60.

Ξυνόγαλος, Σ. & Σατρατζέμη, Μ. (2002). Εκπαιδευτικά Προγραμματιστικά Περιβάλλοντα βασισμένα στους Εκδότες Σύνταξης: Ανάλυση Εργαλείων και Επισκόπηση Αποτελεσμάτων Αξιολόγησης, *Πρακτικά 3ου Πανελλήνιου Συνεδρίου*

με Διεθνή Συμμετοχή «Οι ΤΠΕ στην Εκπαίδευση», 87-96 Β' τόμος.

Σιώζου, Σ. & Τσέλιος, Ν. & Κόμης, Β. (2008). Επίδραση της χρήσης πολλαπλών αναπαραστάσεων σε εκπαιδευτικό περιβάλλον προγραμματισμού. *Πρακτικά 4ου Πανελλήνιου Συνεδρίου Διδακτική της Πληροφορικής*, 161-170.