«Αβάκιο», ένα μαθησιακό περιβάλλον βασισμένο σε ψηφίδες λογισμικού

Μ. Κουτλής, Χ. Κυνηγός, Γ. Τσιρώνης, Κ. Κυρίμης, Μ. Δεκόλη, Γ. Βασιλείου

{koutlis, kynigos, tsironis, kyrimis, dekoli, vasiliou}@cti.gr

Περίληψη

Στο άρθρο αυτό παρουσιάζεται η ερευνητική εργασία γύρω από το σχεδιασμό και την ανάπτυξη του «Αβακίου», ενός μαθησιακού περιβάλλοντος που προσφέρεται για την κατασκευή διερευνητικού εκπαιδευτικού λογισμικού υψηλής ποιότητας από μή προγραμματιστές. Το Αβάκιο υιοθετεί μια ψηφιδο-κεντρική προσέγγιση σχεδιασμού και ανάπτυξης λογισμικού (component software), εισάγοντας μια πολυεπίπεδη μεθοδολογία ανάπτυξης εκπαιδευτικού λογισμικού με κύριο χαρακτηριστικό την ισοβαρή εκπροσώπηση της εκπαιδευτικής κοινότητας έναντι των τεχνολόγων στη διαδικασία ανάπτυξης. Οι Κατασκευαστές Ψηφίδων (τεχνολόγοι) σχεδιάζουν και αναπτύσουν επαναχρησιμοποιήσιμες και αλληλοσυνεργάσιμες Ψηφίδες, γενικού σκοπού ή προσανατολισμένες προς συγκεκριμένους γνωστικούς χώρους, οι οποίες προορίζονται να παίξουν το ρόλο δομικών λίθων στα χέρια των συγγραφέων εκπαιδευτικών Μικρόκοσμων. Οι Σχεδιαστές Εκπαιδευτικών Δραστηριοτήτων, συνήθως ερευνητές εκπαιδευτικοί, ειδικοί στην ανάπτυξη εκπαιδευτικού υλικού για τις γνωστικές περιοχές της ειδικότητάς τους, σχεδιάζουν δραστηριότητες με βάση συγκεκριμένους εκπαιδευτικούς στόχους και χρησιμοποιούν τα εργαλεία σύνθεσης Μικρόκοσμων για ν'αναπτύξουν το εκπαιδευτικό λογισμικό. Το Αβάκιο έχει αναπτυχθεί σε Java και φιλοξενείται στο web-site: http://E-Slate.cti.gr.

Abstract

This paper reports on research work on the design and development of E-Slate, an end-user environment for creating high-quality educational software of exploratory nature. Component developers produce domain-specific, reusable, interoperable and programmable software components that will play the role of primitive -albeit rather complex!- building blocks in the hands of activity authors. Individual component specs stem from the nature of the envisaged educational activities. Activity authors are discipline experts-researchers who design learning activities with pedagogical principles and educational targets in mind -eventually identifying concrete component requirements. They use authoring tools to assemble components into generic activity templates representing "families" of many individual activities. These templates are customized and enriched with specific content either by content providers or by teachers and learners according to individual needs. E-Slate is currently based on the Java platform and related technologies, and hosted at *http://E-Slate.cti.gr*

Index terms: component software architectures, educational software development, authoring systems, exploratory software, Internet based tools.

1. Εισαγωγή

Είναι ευρέως παραδεκτό ότι η υιοθέτηση νέων τεχνολογιών στην εκπαίδευση δυνητικά είναι ο καταλύτης μιας βαθειάς αλλαγής στον τρόπο μάθησης και διδασκαλίας. Αν και η τεχνολογία είναι πολύ πιθανό να βρει χρήσεις σε πιο «κλασσικούς» ρόλους διδακτικής πρακτικής, διαφαίνεται πιθανότερο ότι μια μαθητο-κεντρική, κοστρακτιβιστική προσέγγιση έχει να προσφέρει γονιμώτερο έδαφος για χρήση των νέων τεχνολογιών, υποστηρίζοντας έναν **διερευνητικό τρόπο μάθησης** [0, 0, 0, 0].

Διερευνητικό είναι το λογισμικό που παρέχει δυνατότητες για τη δυναμική σύνδεση διαφορετικών αναπαραστάσεων οντοτήτων και εννοιών του μελετώμενου γνωστικού χώρου, την εξομοίωση φυσικών φαινομένων, τη διενέργεια πειραμάτων και συγκρίσεων μεταξύ διαφορετικών σεναρίων με αλλαγή συνθηκών και παραμέτρων, την υποστήριξη ομαδικής εργασίας και ανταλλαγής δεδομένων, εργασιών και εμπειριών, την υποστήριξη συμβολικής έκφρασης μέσα από υψηλού επιπέδου γλώσσες ή άλλα συμβολοσυστήματα, την αποθήκευση και διαχείρηση πληροφορίας κατά βούληση, την αποθήκευση των ενεργειών του μαθητή για μετέπειτα ανάλυση ή επανεξέταση, τη χρήση πλούσιου εποπτικού υλικού (χάρτες, φωτογραφίες, animations, γραφήματα, κλπ).

Ερωτήματα που εύλογα έπονται, αφορούν το «πώς» μπορεί να παραχθεί τέτοιου είδους λογισμικό, από «ποιόν», με τί κόστος, σε ποιά ποιότητα και ποσότητα. Και τα ερωτήματα αυτά αποκτούν ιδιαίτερο ενδιαφέρον σε συνδυασμό με δυο επιπλέον διαπιστώσεις:
α) Μια ματιά στη διεθνή αγορά εκπαιδευτικού λογισμικού αρκεί για να αναδείξει ότι δεν υπάρχουν παρά ελάχιστα δείγματα λογισμικού του είδους που προδιαγράφτηκε (της τάξης των είκοσι) [0].
β) Άριστα δείγματα εκπαιδευτικού λογισμικού που έχουν παραχθεί σε ερευνητικά πλαίσια δεν κατάφεραν να έχουν τον αναμενόμενο αντίκτυπο στην εκπαιδευτική διαδικασία κλίμακας [0].
Γιατί συμβαίνει αυτό; Ψήγματα της απάντησης βρίσκονται στα ακόλουθα:
1. Η αγορά του εκπαιδευτικού λογισμικού είναι μικρή (σχετικά με άλλες, όπως για παράδειγμα αυτής του "edutainment"), με αποτέλεσμα την αποθάρρυνση των απαραίτητων επενδύσεων για R&D [0]. Βασικά αίτια αυτής της πραγματικότητας είναι:

- Η έλλειψη προϋπολογισμού από τα σχολεία για αγορά εκπαιδευτικού λογισμικού, που με τη σειρά της οφείλεται στην έλλειψη μοντέρνου υπολογιστικού εξοπλισμού στα σχολεία, και –πιο σημαντικό- την έλλειψη διαδικαστικού πλαισίου χρήσης των νέων τεχνολογιών (π.χ. πρόβλεψη από το πρόγραμμα σπουδών, επιμόρφωση των εκπαιδευτικών στους νέους τρόπους διδασκαλίας με χρήση νέων τεχνλογιών).
- Η «δυστροπία» του χώρου: πολλές και διαφορετικές προδιαγραφές, εκπαιδευτικά συστήματα, παιδαγωγικές προσεγγίσεις, διαδικαστικές δυσκολίες (π.χ. έγκριση από Υπουργείο Παιδείας και αρμόδιους φορείς).

2. Η παραγωγή ποιοτικού εκπαιδευτικού λογισμικού (του είδους που προδιαγράφτηκε στο προηγούμενο κεφάλαιο) είναι πολύ ακριβή. Ενδεικτικά αναφέρεται ότι το κόστος παραγωγής εκτιμάται σε 80 μέχρι και 600 εκ. δρχ για ένα «ανταγωνιστικό» προϊόν, γεγονός που είτε αποθαρρύνει τους παραγωγούς, είτε οδηγεί σε προϊόντα χαμηλής ποιότητας ως προς την εκπαιδευτική τους διάσταση. Το τελευταίο, οδηγεί την εκπαιδευτική κοινότητα σε σκεπτικισμό ως προς τη σκοπιμότητα της χρήσης του [0, 0], επιτείνοντας έτσι την κατάσταση που περιγράφτηκε στο (1).
3. Η παραγωγή εκπαιδευτικού λογισμικού γίνεται αποσπασματικά και με ad-hoc μεθόδους με αποτέλεσμα τη δημιουργία κατακερματισμένης αγοράς. Πρωτότυπα ή προϊόντα που προέρχονται είτε από το χώρο της έρευνας είτε της αγοράς, καταλήγουν σε συλλογές από ασύμβατα προϊόντα λογισμικού, καθένα εστιασμένο σε συγκεκριμένη γνωστική περιοχή με ιδιαίτερη φιλοσοφία χρήσης, χωρίς πρόβλεψη συνέργιας [0].
Θέματα του είδους που αναφέρονται στο πρώτο σημείο είναι βέβαια πολύ ευρύτερου βεληνεκούς και χρήζουν αντιμετώπιση αντίστοιχης κλίμακας, αλλά τα θέματα των δυο επόμενων σημείων είναι σίγουρα περισσότερο προσιτά. Εδώ το πρόβλημα εντοπίζεται στην ανεπάρκεια των κλασσικών μεθόδων ανάπτυξης εκπαιδευτικού λογισμικού, στην ανάπτυξη δηλαδή μεμονωμένων «εφαρμογών» είτε με χρήση εργαλείων ανάπτυξης γενικού σκοπού (development tools) είτε με χρήση «συγγραφικών συστημάτων» (authoring tools). Οι προσεγγίσεις αυτές πέραν του παράγοντα κόστους (ανάπτυξης αλλά και συντήρησης), μειονεκτούν και ως προς το παραγόμενο αποτέλεσμα των «κλειστών» εφαρμογών που προσφέρουν ελάχιστες δυνατότητες αλληλοσυνέργιας, ελευθερίας δομικών επεμβάσεων και σύνθεσης από τους ενδιάμεσους ή τελικούς χρήστες. Το γεγονός αυτό απομονώνει από την παραγωγική διαδικασία τους εκπαιδευτικούς, που είναι λόγω ιδιότητας οι πλέον κατάλληλοι σχεδιαστές λογισμικού για εκπαιδευτική χρήση.
Ας δούμε όμως λεπτομερέστερα τη διαδικασία παραγωγής εκπαιδευτικού λογισμικού. Διακρίνουμε τρια σχήματα.

1.1. Ανάπτυξη «εφαρμογών»

Η ανάπτυξη του λογισμικού γίνεται σε συνεργασία μιας ομάδας από εκπαιδευτικούς και τεχνολόγους (με την αναλογία να βαρύνει συνήθως προς τη μια κατεύθυνση) όπου οι πρώτοι προσδιορίζουν το αντικείμενο, το περιεχόμενο και τις λειτουργικές απαιτήσεις του λογισμικού και οι δεύτεροι αναλαμβάνουν την υλοποίησή του χρησιμοποιώντας μεθόδους της αρεσκίας

τους. Ο κύκλος επαναλαμβάνεται για κάθε καινούργια απαίτηση ή ιδέα. Η διαδικασία αυτή μπορεί μεν να οδηγήσει σε αξιόλογα προϊόντα, τα οποία όμως σπάνια είναι επαναχρησιμοποιήσιμα για περιστάσεις πέρα από το πεδίο εφαρμογών για το οποίο σχεδιάστηκαν με αποτέλεσμα να μένουν συνήθως σε περιορισμένη χρήση. Μ'άλλα λόγια είναι «κομμένα και ραμμένα» στα μέτρα των συγκεκριμένων απαιτήσεων και σχεδιασμού (εκπαιδευτικού και τεχνολογικού) και για έστω μικρές αλλαγές ή διαφοροποιήσεις απαιτείται επανάληψη του κύκλου ανάπτυξης κάνοντας την όλη διαδικασία απαγορευτικά δαπανηρή.
Το πρόβλημα σ'αυτή την περίπτωση εστιάζεται στην ανεπάρκεια του μοντέλου ανάπτυξης «εφαρμογών» (applications), οι οποίες λειτουργώντας αυτόνομα, έχουν ελάχιστες δυνατότητες συνεργασιμότητας με άλλες εφαρμογές, μικρές πιθανότητες επαναχρησιμοποίησης, και μεγάλο κόστος συντήρησης, βελτιώσεων ή αλλαγών.

1.2 Χρήση «συγγραφικών συστημάτων»

Στο σχήμα αυτό οι εκπαιδευτικοί υιοθετούν κάποιο «σύστημα συγγραφής» προκειμένου να υλοποιήσουν οι ίδιοι κάποιες ιδέες εκπαιδευτικών σεναρίων. Η εμφάνιση συστημάτων όπως για παράδειγμα το Apple Hypercard, το Asymetrix Toolbook, και τα MacroMind Director και Authorware, όπως στο παρελθόν και η γλώσσα Basic είχε στόχο την κάλυψη τέτοιου είδους αναγκών όπου μη ειδικοί χρειάζονταν κάποιο εργαλείο που θα τους έδινε «εύκολη» πρόσβαση στον υπολογιστή για την υλοποίηση των ιδεών τους. Αργά ή γρήγορα όμως οι εκπαιδευτικοί βρίσκονται στη θέση να ξοδεύουν πολύ περισσότερο χρόνο για ανάπτυξη του λογισμικού απ'ότι για τη χρήση του προς επίτευξη του επιδιωκόμενου μαθησιακού αποτελέσματος: σύμφωνα με στατιστικές χρησιμοποιώντας ένα σύστημα συγγραφής όπως το Asymetrix Toolbook απαιτούνται 200 περίπου ώρες εργασίας για την ανάπτυξη μίας ώρας εκπαιδευτικού λογισμικού* [0] και μάλιστα από ειδικευμένο χρήστη. Επιπρόσθετα, τα αποτελέσματα ως προς το τελικό αποτέλεσμα είναι πολύ μέτρια: φτωχή λειτουργικότητα, δυσκίνητη εκτέλεση, μη επαναχρησιμοποιήσιμες ή συντηρήσιμες κατασκευές. Κι αυτό γιατί η αποτελεσματική χρησιμοποίηση συγγραφικών συστημάτων απαιτεί επένδυση από την πλευρά των εκπαιδευτικών για την οποία στην πλειοψηφία των περιπτώσεων είτε δεν είναι διατεθειμένοι, είτε δεν έχουν το χρόνο να κάνουν. Συνήθης κατάληξη αυτών των προσπαθειών είναι η αναζήτηση βοήθειας από κάποιον «προγραμματιστή» ο οποίος καλείται να φέρει σε πέρας το στοχευόμενο λογισμικό, υποχρεωμένος να χρησιμοποιήσει το ίδιο περιβάλλον συγγραφής, το οποίο φυσικά δεν παρέχει τα εργαλεία ή τις δυνατότητες που θα ήθελε από ένα «επαγγελματικό» περιβάλλον ανάπτυξης.
Στο δεύτερο αυτό λοιπόν σχήμα ανάπτυξης, το πρόβλημα εντοπίζεται:
1. Στον πολύ γενικού τύπου χαρακτήρα των συγγραφικών συστημάτων πράγμα που α) επιβάλλει το ξεκίνημα από μηδενική κάθε φορά βάση για την ανάπτυξη εκπαιδευτικών εφαρμογών και β) δεν αποφεύγει την εμπλοκή του χρήστη σε συλλογιστική κύκλου ανάπτυξης εφαρμογών, δηλαδή τη μοντελοποίηση οντοτήτων με βάση προκαθορισμένες δομές δεδομένων, ανάπτυξη αλγορίθμων και κωδικοποίησή τους σε κάποια γλώσσα προγραμματισμού, βήματα αναπόφευκτα έστω και για απλοϊκές κατασκευές λογισμικού.
2. Στην αδυναμία των συγγραφικών συστημάτων να αντεπεξέλθουν στις απαιτήσεις χρήσης από τεχνολόγους για την αποτελεσματική ανάπτυξη εφαρμογών που ξεφεύγουν από «απλές» λειτουργικές απαιτήσεις.

1.3 Χρήση εξειδικευμένων εργαλείων

Στο τρίτο κατά σειρά σχήμα η ανάπτυξη του λογισμικού γίνεται με εξειδικευμένα περιβάλλοντα, φτιαγμένα αποκλειστικά για εκπαιδευτικές χρήσεις. Το στοχευόμενο κοινό είναι η εκπαιδευτική κοινότητα, ερευνητές, εκπαιδευτικοί και μαθητές, οι οποίοι μόνοι τους κατασκευάζουν «Μικρόκοσμους». Βασική υποκείμενη σχεδιαστική αρχή σ'αυτά τα περιβάλλοντα είναι η πεποίθηση:
α) Ότι θα πρέπει να παρέχεται στον «τελικό χρήστη» η δυνατότητα πλήρους πρόσβασης στα εργαλεία που επιτρέπουν την (ανα)κατασκευή Μικρόκοσμων κατά βούληση.

β) Ότι η «κατασκευή» του προς μελέτη αντικειμένου είναι εξίσου σημαντική διαδικασία όσο και η καθαυτό μελέτη του.

Με μια ματιά σε μερικά από τα πλέον διαδεδομένα τέτοια περιβάλλοντα, διακρίνουμε δυο βασικές κατηγορίες: γενικού και ειδικού σκοπού. Στην πρώτη κατηρορία ανήκουν περιβάλλοντα όπως η Logo και το Boxer, ενώ εργαλεία όπως το Rehearsal-world [0], το Playground [0], το Geometre's SketchPad [] και το Agentsheets [0], τα οποία παρέχουν στο χρήστη οντότητες υψηλού επιπέδου με προκαθορισμένα χαρακτηριστικά και συμπεριφορά (συνήθως εστιασμένα σε συγκεκριμένους γνωστικούς χώρους), καθώς επίσης και τη δυνατότητα δόμησης με βάση τα αντικείμενα αυτά, κατατάσσονται στη δεύτερη κατηγορία.

Τα περιβάλλοντα της πρώτης κατηγορίας, όντας γενικού σκοπού παρουσιάζουν τα ίδια προβλήματα που αναφέρθηκαν παραπάνω για τα συστήματα συγγραφής: οι χρήστες ξεκινάνε την «ανάπτυξη» λογισμικού πάντα από μηδενική βάση, χρησιμοποιώντας τις χαμηλού επιπέδου δομές της γλώσσας προγραμματισμού για να κωδικοποιήσουν τη δομή και συμπεριφορά του στοχευόμενου λογισμικού, εργασία επίπονη της οποίας η πολυπλοκότητα αυξάνει εκθετικά ως προς τις απαιτήσεις.

Τα εργαλεία της δεύτερης κατηγορίας απαλλάσουν μεν το χρήστη από την ενασχόληση με τεχνικές λεπτομέρειες, παρέχοντάς του την άνεση να σκεφτεί σε υψηλότερο νοηματικό επίπεδο και να εστιάσει αποκλειστικά στη μαθησιακή διαδικασία, αλλά μπορούν να πραγματώσουν μικρό μόνο εύρος εκπαιδευτικών εφαρμογών λόγω ακριβώς της εξειδίκευσης των δομών, αντικειμένων και τρόπων έκφρασης που παρέχουν (π.χ. γεωμετρικές κατασκευές με το Sketchpad, animations). Επιπρόσθετα, οι κατασκευές (λογισμικό) που παράγονται δεν είναι χρησιμοποιήσιμες παρά αυστηρά μέσα στα όρια του κάθε συστήματος και τα αντικείμενα-δομικοί λίθοι δεν είναι ανταλλάξιμοι από το ένα στο άλλο. Μια πυραμίδα π.χ. που κατασκευάζεται προς μελέτη με το Sketchpad ως προς τη γεωμετρική του υφή, δεν μπορεί να χρησιμοποιηθεί σ'έναν Μικρόκοσμο Γεωγραφίας ή/και Ιστορίας ως αντικείμενο μελέτης στο χάρτη της Αιγύπτου.

Ποιός θα ήταν ο τρόπος να συνδυαστούν τα πλεονεκτήματα και των δυο αυτών κόσμων απαλείφοντας παράλληλα τις εγγενής τους αδυναμίες; Με ποιόν τρόπο θα μπορούσε να στοιχειοθετηθεί κάποιο περιβάλλον που θα παράσχει δομικούς λίθους υψηλού επιπέδου αλλά γενικού σκοπού μέσα σ'ένα προγραμματιστικό περιβάλλον που θα επιτρέπει τη διαχείριση των χαρακτηριστικών και της συμπεριφοράς τους στο κατάλληλο επίπεδο έκφρασης; Πώς θα εξασφαλίζονταν οι βασικές σχεδιαστικές αρχές διερευνητικού λογισμικού που μεταξύ άλλων μιλούν για δυνατότητα διαμόρφωσης των χαρακτηριστικών και της λειτουργικότητας των Μικρόκοσμων κατά βούληση από τους τελικούς αποδέκτες ως μέρος της μαθησιακής διαδικασίας; Πώς θα μπορούσε η όποια καινούργια αυτή μέθοδος να επιφέρει ένα συνεργατικό κλίμα ανάμεσα σε εκπαιδευτικούς και τεχνολόγους εξισορροπώντας το βαθμό ανάμειξης των δυο πλευρών στη διαδικασία ανάπτυξης λογισμικού και παράλληλα αναθέτοντας ρόλους αρμοστούς στις ικανότητες και απαιτήσεις του καθενός;

Ερωτήματα σαν κι αυτά οδήγησαν στο σχεδιασμό του περιβάλλοντος «Αβάκιο» το οποίο παρέχει προκατασκευασμένα αντικείμενα υψηλού επιπέδου, τις «Ψηφίδες», προσανατολισμένες στο γνωστικό αντικείμενο ενδιαφέροντος, σχεδιασμένες ώστε να είναι αλληλοσυνεργάσιμες μεταξύ τους σε διατάξεις που συντίθενται από τους «χρήστες» και συνολικά αποδίδουν την επιθυμητή λειτουργικότητα στους Μικρόκοσμους.

2. Το Αβάκιο

Τα βασικά χαρακτηριστικά του Αβακίου διαμορφώθηκαν μέσα από συνδυασμό των γενικών αρχών για το χαρακτήρα του διερευνητικού λογισμικού και των ειδικώτερων απαιτήσεων των γεωγραφικών Μικρόκοσμων κατά την αντιστοιχία που φαίνεται στον παρακάτω πίνακα.

Κύριες «απαιτήσεις» (οπτική γωνία «χρήστη»)	**Βασικές σχεδιαστικές αποφάσεις** (οπτική γωνία τεχνικού)
Δυνατότητα πρόσβασης και χειρισμών σε παράλληλες και διασυνδεόμενες εναλλακτικές αναπαραστάσεις των φαινομένων, οντοτήτων και εννοιών μελέτης.	Σχεδιασμός και ανάπτυξη «Ψηφίδων» λογισμικού (software components) που είναι προσανατολισμέ-νες σε συγκεκριμένους γνωστικούς χώρους ενδιαφέροντος μοντελοποιώντας οντότητες, έννοιες σχέσεις και φαινόμενα κάθε χώρου, ταιριαστές στο νοητικό επίπεδο των χρηστών.
Δυνατότητα πρόσβασης σε εργαλεία γενικού τύπου όπως βάσεις δεδομένων, γραφήματα, χάρτες, εξομοιωτές, κλπ.	Σχεδιασμός και ανάπτυξη των εν λόγω εργαλείων στη μορφή αλληλοσυνεργάσιμων και επαναχρησιμοποιήσιμων Ψηφίδων λογισμικού.
Δυνατότητα επιλεκτικής χρήσης αναπαραστάσεων και εργαλείων σε κατά βούληση διατάξεις ανάλογα με τις απαιτήσεις κάθε εκπαιδευτικού σεναρίου χρήσης.	Ανάπτυξη εργαλείου δημιουργίας και διαχείρισης «Μικρόκοσμων» το οποίο μεταξύ άλλων θα επιτρέπει την ομαδοποίηση και χωροθέτηση Ψηφίδων στο interface ενός Μικρόκοσμου την αποθήκευση και ανάκτησή τους, την αναδιάταξη και ανασύνθεσή τους κλπ.
Δυνατότητα συγχρονισμού και γενικώτερα λειτουργικού συσχετισμού των εναλλακτικών αναπαραστάσεων κατά βούληση, ανάλογα με τις απαιτήσεις των οπτικοποιήσεων ή/και εξομοιώσεων του υλοποιούμενου εκπαιδευτικού σεναρίου.	Παροχή μηχανισμού που επιτρέπει τον ορισμό αλληλοσυσχετίσεων μεταξύ Ψηφίδων (με οπτικό ή συμβολικό τρόπο) εξασφαλίζοντας την ανταλλαγή δεδομένων και συνδυασμένη λειτουργικότητα Ψηφίδων μέσω κατάλληλων επικοινωνιακών καναλιών και πρωτοκόλλων.
Δυνατότητα ενσωμάτωσης δεδομένων και πληροφοριών κατασκευασμένων από τους χρήστες, όπως π.χ. multimedia assets ή ακόμα και πιο πολύπλοκες οντότητες.	Παροχή εργαλείων για τη δημιουργία αντικειμένων διαφόρων –προκαθορισμένων-τύπων με ή χωρίς συμπεριφορά (π.χ. agents).
Δυνατότητα «προγραμματισμού» της συμπεριφοράς των αναπαραστάσεων ή/και της λειτουργικότητας των εργαλείων με αλγοριθμικό τρόπο, μέσα από κάποια «άμεση» γλώσσα προγραμματισμού.	Παροχή μηχανισμών scripting (άμεσου προγραμματισμού) που επιτρέπουν τη διαχείριση των Ψηφίδων μέσα από κατάλληλα γραμμένα scripts (σύνολα εντολών που εκτελούνται σειριακά) ως προς τις ιδιότητες και τα λειτουργικά τους χαρακτηριστικά.
Δυνατότητα δημοσιοποίησης και ανταλλαγής των «κατασκευών» (διατάξεων, αντικειμένων, δεδομένων) μεταξύ χρηστών.	Παροχή δυνατότητας «πακεταρίσματος» των Ψηφίδων και Μικρόκοσμων με τρόπο που να είναι εφικτή η ελεύθερη διακίνησή τους μέσω δικτύου.
Υποστήριξη συνεργατικής εργασίας μεταξύ ομάδων παιδιών.	Υποστήριξη μηχανισμών που εξασφαλίζουν συνέργια μεταξύ Ψηφίδων και Μικρόκοσμων που βρίσκονται σε απομακρυσμένους σταθμούς εργασίας.

2.1 Η μεθοδολογία ανάπτυξης λογισμικού «Αβάκιο»

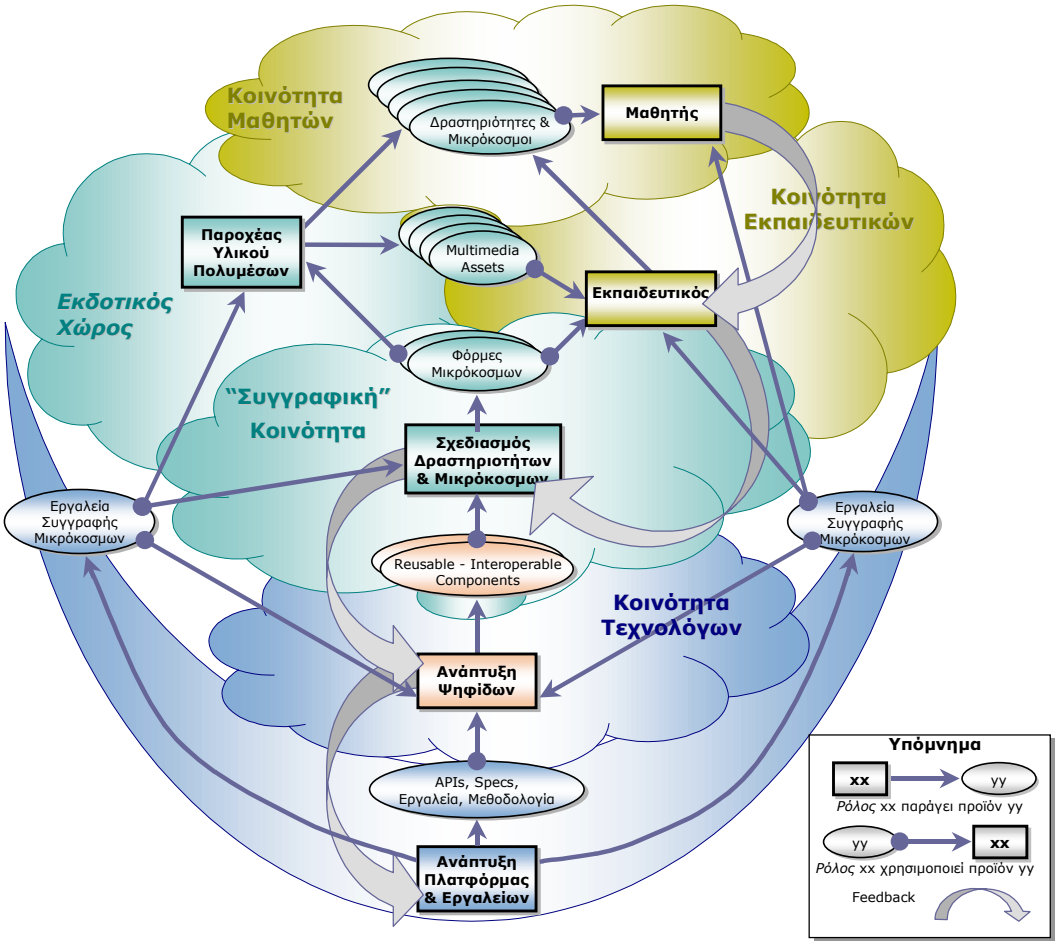
Το Αβάκιο εισάγει μια πολυεπίπεδη μεθοδολογία ανάπτυξης εκπαιδευτικού λογισμικού, διακρίνοντας σαφώς τις αρμοδιότητες των διαφορετικών σταδίων και αντιστοίχως εμπλεκόμενων ρόλων, όπως εικονίζεται στο Σχήμα 1. Το πλεονέκτημα της μεθόδου έγκειται στο ότι οι επιμέρους ρόλοι, και κατά κύριο λόγο των εκπαιδευτικών από τη μια -ως των πλέον κατάλληλων για το σχεδιασμό εκπαιδευτικών δραστηριοτήτων-, και των τεχνολόγων από την άλλη -ως των πλέον αρμόδιων για την ανάπτυξη αποδοτικού λογισμικού- διακρίνονται σαφώς με τρόπο που καθεμία να εξυπηρετείται βέλτιστα: οι μεν πρώτοι μπορούν να σχεδιάσουν στο νοητικό επίπεδο του γνωστικού πεδίου ενδιαφέροντος και όχι σ'αυτό που επιβάλλεται από μια γλώσσα προγραμματισμού, οι δε τελευταίοι αποκτούν την ελευθερία επιλογής του κατά την κρίση τους κατάλληλου τεχνολογικού υπόβαθρου και εργαλείων για την ανάπτυξη των Ψηφίδων. Μ' άλλα λόγια η εκπαιδευτική κοινότητα έχοντας πρόσβαση σε υπολογιστικές οντότητες με τρόπο που να εξασφαλίζει την ελευθερία της εύκολης σύνθεσης και επανασύνθεσης προσωπικών κατασκευών, αποκτά δυνητικά πρωταγωνιστικό ρόλο στην παραγωγή εκπαιδευτικού λογισμικού.

Αναλυτικά, η φύση και αρμοδιότητες των εμπλεκόμενων ρόλων έχουν ως εξής:

1. Οι Κατασκευαστές Εργαλείων, σχεδιάζουν και αναπτύσουν την τεχνολογική βάση (πλατφόρμα) η οποία στηρίζει όλα τα υπεκείμενα επίπεδα και συγκεκριμένα: α) τα απαραίτητα APIs (application programming interfaces) και βιβλιοθήκες για χρήση από τους κατασκευαστές Ψηφίδων, β) τα εργαλεία συγγραφής για τη σύνθεση Μικρόκοσμων από τους συγγραφείς εκπαιδευτικών δραστηριοτήτων και γ) τα εργαλεία χρήσης και διαχείρισης των Μικρόκοσμων από τους τελικούς αποδέκτες.
2. Οι Κατασκευαστές Ψηφίδων (τεχνολόγοι) σχεδιάζουν και αναπτύσουν επαναχρησιμοποιήσιμες και αλληλοσυνεργάσιμες Ψηφίδες, γενικού σκοπού ή προσανατολισμένες προς συγκεκριμένους γνωστικούς χώρους, οι οποίες προορίζονται να παίξουν το ρόλο δομικών λίθων στα χέρια των συγγραφέων Μικρόκοσμων. Χρησιμοποιούν τις σχεδιαστικές κατευθύνσεις ή/και εργαλεία που παρέχονται από τους Κατασκευαστές Εργαλείων και χρησιμοποιούν επαγγελματικού τύπου περιβάλλοντα και γλώσσες ανάπτυξης (Java, C++ κλπ).
3. Οι Σχεδιαστές Εκπαιδευτικών Δραστηριοτήτων είναι συνήθως ερευνητές εκπαιδευτικοί, ειδικοί στην ανάπτυξη εκπαιδευτικού υλικού για τις γνωστικές περιοχές της ειδικότητάς τους. Σχεδιάζοντας τις δραστηριότητες με βάση συγκεκριμένους εκπαιδευτικούς στόχους, εντέλει καταλήγουν στη διατύπωση συγκεκριμένων απαιτήσεων προς τους Κατασκευαστές Ψηφίδων για καινούργια χαρακτηριστικά και λειτουργίες (όπου αυτά δεν καλύπτονται από υπάρχουσες Ψηφίδες). Χρησιμοποιούν τα εργαλεία σύνθεσης Μικρόκοσμων που παρέχουν οι Κατασκευαστές Εργαλείων για να φτιάξουν «φόρμες» (καλούπια, templates) υποδειγματικών δραστηριοτήτων οι οποίες εκπροσωπούν ολόκληρες «οικογένειες» Μικρόκοσμων. Μ'αυτό τον τρόπο, δίνεται η δυνατότητα στους ειδικούς ν' αποτυπώνουν τις σχεδιαστικές τους ιδέες και τις επιθυμητές εκπαιδευτικές προσεγγίσεις σε φόρμες-γεννήτριες που μπορούν να χρησιμοποιηθούν από τρίτους (εκπαιδευτικούς, εκδότες, κλπ) για την κατασκευή πλειάδας επιμέρους Μικρόκοσμων (οι οποίοι όμως ενσωματώνουν τις σχεδιαστικές αρχές και εκπαιδευτικές απόψεις των ειδικών). Οι Μικρόκοσμοι αυτοί υλοποιούνται είτε με απλή προσθήκη πρωτογεννούς multimedia υλικού στις φόρμες, είτε με μικρο-αλλαγές και προσαρμογή των χαρακτηριστικών και της λειτουργικότητας που προδιαγράφεται από μια φόρμα.
4. Οι Εκδότες ή/και Παροχείς «Υλικού» συνισφέρουν multimedia πρωτογεννές υλικό που μπορεί είτε α) να προσαρμοστεί σε κάποια προκατασκευασμένη φόρμα παράγοντας έτσι μια σειρά από συναφείς σε υφή και χαρακτήρα Μικρόκοσμους (καθένας όμως με εξειδικευμένο περιεχόμενο), είτε β) να πλαισιώσει ειδικά σχεδιασμένους Μικρόκοσμους εξ'αρχής (δηλαδή όχι αναγκαστικά βασισμένους σε φόρμες).
5. Οι Εκπαιδευτικοί μπορούν α) να χρησιμοποιήσουν προκατασκευασμένους Μικρόκοσμους β) να προσαρμόσουν ήδη υπάρχοντες ή/και φόρμες Μικρόκοσμων (που αναπτύσσονται από

τους ειδικούς Σχεδιαστές Δραστηριοτήτων) με προσθήκη προσωπικού υλικού, λειτουργικών ή δομικών παρεμβάσεων, κλπ και γ) να σχεδιάσουν προσωπικής τους έμπνευσης Μικρόκοσμους εξ'αρχής. Σε όλες τις περιπτώσεις, έχουν στη διάθεσή τους τα ίδια εργαλεία σύνθεσης και διαχείρισης Μικρόκοσμων που χρησιμοποιούν και οι δυο προηγούμενες κατηγορίες «χρηστών».

6. Οι Μαθητές, όπως και οι εκπαιδευτικοί, είναι αποδέκτες Μικρόκοσμων, φορμών, πρωτογενούς υλικού και εργαλείων σύνθεσης, έχοντας έτσι τη δυνατότητα επιλογής ρόλου από αυτόν του απλώς χρηστών έτοιμου υλικού, έως το ρόλο του συνθέτη ιδιοκατασκευών.



Σχήμα 1: Η μεθοδολογία ανάπτυξης λογισμικού με το Αβάκιο

2.2 Το Αβάκιο ως μαθησιακό περιβάλλον

Το Αβάκιο προσφέρει στην εκπαιδευτική κοινότητα -ερευνητές, εκπαιδευτικούς, μαθητές, συγγραφείς εκπαιδευτικών δραστηριοτήτων, εκδότες- εργαλεία υψηλού επιπέδου για τη σύνθεση (παρά «ανάπτυξη») Μικρόκοσμων για πειραματισμό και διερεύνηση φαινομένων, εννοιών, υποθέσεων και συχετισμών.

Δομήσιμο λογισμικό

Σ'ένα περιβάλλον όπως το Αβάκιο, ιδέες για εκπαιδευτικές δραστηριότητες μπορούν εύκολα και γρήγορα να μετατραπούν σε λογισμικό με την απαιτούμενη λειτουργικότητα. Με απλούς χειρισμούς, οι χρήστες μπορούν να συνδυασουν Ψηφίδες και να δομήσουν Μικρόκοσμους

όπως αυτόν που εικονίζεται παρακάτω. Το συγκεκριμένο παράδειγμα, στοχεύοντας στην εξοικοίωση των μαθητών με έννοιες διανυσμάτων, στρέφεται γύρω από την ιδέα-παιχνίδι της υλοποίησης πτήσεων μ'ένα αεροπλάνο πάνω από την Ευρώπη. Οι μαθητές μπορούν να κατευθύνουν χειριζόμενοι το πηδάλιο (διάνυσμα) της ώθησης και κατεύθυνσης, ενώ ταυτόχρονα μπορούν να ορίσουν και τις συνθήκες ανέμου (ταχύτητα – κατεύθυνση) που επηρρεάζει την πτήση του αεροπλάνου το οποίο κινείται τελικά σύμφωνα με τη συνισταμένη των δυο διανυσμάτων. Οι Ψηφίδες Χάρτης, Διάνυσμα (επί τρία), Αεροπλάνο, Ρολόι και Χρονιστής τοποθετούνται στην οθόνη του Μικρόκοσμου και «διασυνδέονται» κατάλληλα (βλέπε παρακάτω) έτσι ώστε να λειτουργούν «εν χορώ»:

- Το Διάνυσμα πάνω δεξιά συνδέεται στι Αεροπλάνο έτσι ώστε να καθορίζει την ώθηση των μηχανών και κατεύθυνση του αεροπλάνου (δρώντας ως πηδάλιο πλοήγησης).
- Το αμέσως πιο κάτω Διάνυσμα ορίζεται να αναπαριστά την κατεύθυνση και δύναμη του ανέμου, και συνδέεται επίσης στο Αεροπλάνο έτσι ώστε να το τροφοδοτεί με την κατάσταση του ανέμου.
- Το τελευταίο στη σειρά Διάνυσμα ορίζεται να αναπαριστά τη συνισταμένη των δυο προηγούμενων, δείχνοντας έτσι σε κάθε στιγμή την ταχύτητα του αεροπλάνου ως προς το έδαφος (μέτρο και κατεύθυνση).

Οι παραπάνω ρόλοι των διανυσμάτων, μπορούν βέβαια ν'ανατεθούν και διαφορετικά, ανάλογα με τις προθέσεις του σχεδιαστή του Μικρόκοσμου, απλά αλλάζοντας τη συνδεσμολογία και πιθανώς τη χωροθέτησή τους.

- Το Αεροπλάνο συνδέεται στο Χάρτη, έτσι ώστε ο τελευταίος να μπορεί να δείχνει το στίγμα και την πορεία που διανύει.
- Ο Χρονιστής, ο οποίος δίνει ρυθμό στην όλη προσομοίωση, συνδέεται στο Αεροπλάνο και στο Ρολόι, παρέχοντας χρονικούς «παλμούς» που αξιοποιούνται από τα τελευταία για να υπολογίσουν τις ενέργειές τους σε σχέση με τον χρόνο που περνά.

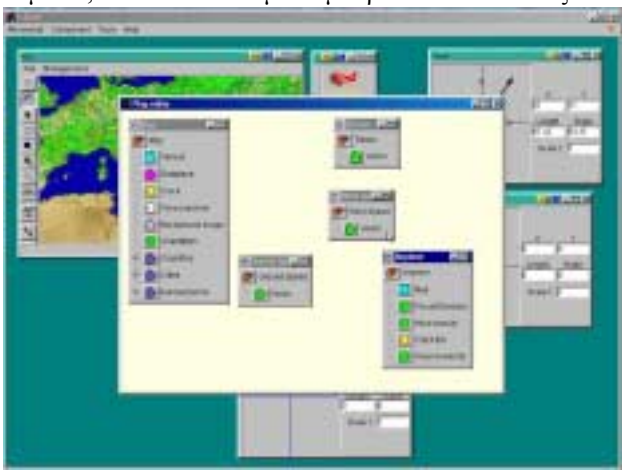
Μετά από λίγο, μια νέα ιδέα έρχεται στο προσκήνιο: καθώς το αεροπλάνο περνάει πάνω από διάφορες χώρες της Ευρώπης, θα μπορούσε να ανασύρει διάφορα στοιχεία για την κάθε χώρα (π.χ. όνομα, πρωτεύουσα, πληθυσμό, έκταση, δείκτες ανάπτυξης, κλπ). Για να υλοποιηθεί κάτι τέτοιο, απλά ενσωματώνεται μια νέα Ψηφίδα στο Μικρόκοσμο, η οποία εικονίζει τα εν λόγω στοιχεία στη μορφή «καρτέλας». Τα στοιχεία αυτά για κάθε χώρα αποτελούν αναπόσπαστο μέρος του χάρτη ο οποίος εκτός από τη γεωγραφική πληροφορία για κάθε περιοχή, διατηρεί και περιγραφικά στοιχεία σε μορφή Βάσης Δεδομένων. Η κίνηση του αεροπλάνου πάνω στο χάρτη, προκαλεί την επιλογή των περιγραφικών αυτών στοιχείων για τη χώρα πάνω από την οποία βρίσκεται, τα οποία στη συνέχεια τροφοδοτούνται στην Ψηφίδα Εξερευνητής Πίνακα (καρτέλα) για απεικόνιση.

Όπως είναι φανερό, θα μπορούσε με τον ίδιο τρόπο να υλοποιηθούν μια σειρά συναφών Μικρόκοσμων στη βάση του ίδιου σεναρίου-ομπρέλα (της πλοήγησης δηλαδή με αεροπλάνο), όπως για παράδειγμα η δημιουργία γραφημάτων ή/και κατανομών για χαρακτηριστικά μεγέθη των διαφόρων χωρών με στόχο τη σύγκριση και εξαγωγή συμπερασμάτων, κλπ.

Έτσι, ο αρχικός Μικρόκοσμος που περιγράφτηκε, θα μπορούσε να χαρακτηριστεί ως γεννήτορας (φόρμα) μιας οικογένειας άλλων Μικρόκοσμων που στρέφονται γύρω από την ίδια βασική ιδέα. Στη φόρμα αυτή θα μπορούσαν να ενσωματωθούν μια σειρά από χάρτες και πληροφορίες για τις περιοχές που εκπροσωπούν (από παροχείς υλικού, ή τους ίδιους τους χρήστες), να διαμορφωθούν διαφορετικοί τρόποι χρήσης με παρεμβάσεις στην εργονομία, όψη ή και λειτουργικότητα των Μικρόκοσμων.

Διασυνδέσιμες Ψηφίδες

Πώς όμως ορίζεται η λειτουργία ενός Μικρόκοσμου; Κάθε Ψηφίδα διαθέτει ένα σύνολο από «Συνδέσμους», καθένας με συγκεκριμένο όνομα που υποδηλώνει το είδος, ρόλο του (π.χ. «χρώμα», «φωτογραφία», «ώρα», κλπ) και συγκεκριμένο χρώμα και σχήμα (σαν κομμάτια από puzzle) που υποδηλώνουν τις δυνατότητες διασύνδεσής του με άλλους συνδέσμους. Οι σύνδεσμοι μιας Ψηφίδας εμφανίζονται επιλέγοντας το κουμπί που βρίσκεται πάνω δεξιά στην μπάρα της και μπορούν να συνδεθούν μεταξύ τους με απλούς και άμεσους χειρισμούς (direct manipulation). Όντας συνδεμένοι, δυο σύνδεσμοι μπορούν ν'ανταλλάξουν πληροφορίες (για λογαριασμό των αντίστοιχων Ψηφίδων στις οποίες ανήκουν) και να συγχρονίσουν έτσι τη λειτουργία τους.



Ο κανόνας διασύνδεσης, είναι απλός: μόνο Σύνδεσμοι που έχουν το ίδιο χρώμα και συμπληρωματικό σχήμα (αρσενικό – θηλυκό) μπορούν να διασυνδεθούν. Η δε διαδικασία διασύνδεσης γίνεται παράλληλα με τη χρήση του Μικρόκοσμου (χωρίς δηλαδή να απαιτείται κάποιου είδους αλλαγή τρόπου λειτουργίας) και οι επιπτώσεις κάθε σύνδεσης στη συμπεριφορά του Μικρόκοσμου είναι άμεσα ορατές. Στο διπλανό στιγμιότυπο εικονίζεται ένα στάδιο της διαδικασίας διασύνδεσης Ψηφίδων.

Προγραμματισιμότητα

Οι Ψηφίδες, επιπρόσθετα της ικανότητάς τους να συνδέονται με άλλες Ψηφίδες για να διαμοιράζονται πληροφορίες και να συγχρονίζουν τη συμπεριφορά τους, παρέχουν τη

δυνατότητα «προγραμματισμού» των χαρακτηριστικών ή/και της συμπεριφορά τους μέσα από εντολές που μπορούν να συνταχθούν σε μια γλώσσα βασισμένη στη Logo.
Αυτή η δυνατότητα μπορεί να ειδωθεί με δυο τρόπους:
α) Η Logo γίνεται η scripting γλώσσα των Ψηφίδων.
β) Οι Ψηφίδες προσφέρονται ως υψηλού επιπέδου αντικείμενα για διαχείριση στο περιβάλλον της Logo επαυξάνοντας έτσι το βελτηνεκές χρησιμοποίησής της [0].
Σε κάθε περίπτωση οι χρήστες έχουν στη διάθεσή τους ένα προσιτό περιβάλλον προγραμματισμού όπως αποδειγμένα προσφέρει η γλώσσα Logo (η οποία επιλέχτηκε γι'αυτόν ακριβώς το λόγο) με κατάλληλα ορισμένες εντολές για τη διαχείριση των Ψηφίδων. Κάθε Ψηφίδα ορίζει συγκεκριμένες εντολές (γλωσσικά "primitives") τις οποίες δηλώνει στο περιβάλλον της Logo (μηχανισμό scripting) κάθε φορά που χρησιμοποιείται, επαυξάνοντας έτσι το ρεπερτόριο εντολών που μπορούν να χρησιμοποιηθούν στη μορφή προγραμμάτων (scripts). Συγκεκριμένες εντολές μπορούν να αποτανθούν σε συγκεκριμένες Ψηφίδες μέσα από τους προσδιορισμούς «Πες» ("Tell"), «Ζήτα» ("Ask") και «Για κάθε» ("Each") συνοδευόμενους από τα ονόματα των Ψηφίδων-παραληπτών.

Ακολουθεί ένα παράδειγμα προγραμματισμού λειτουργιών των Ψηφίδων με χρήση του μηχανισμού scripting, με αναφορά στο Μικρόκοσμο που εικονίζεται στο διπλανό στιγμιότυπο.
Σ'αυτό το παράδειγμα, μια Ψηφίδα Βάση Δεδομένων (πάνω δεξιά) διαχειρίζεται τα στοιχεία (πίνακες δεδομένων) των χωρών της Ευρώπης τα οποία παραλαμβάνει από την Ψηφίδα Χάρτη ως είσοδο μέσω κατάλληλων Συνδέσμων. Ο χρήστης μπορεί να επιλέξει για συγκριτική μελέτη ένα σύνολο από χώρες είτε κατευθείαν απ'το Χάρτη, είτε από τη Βάση Δεδομένων, είτε μέσω εντολής αναζήτησης με βάση κριτήρια από την Ψηφίδα Logo (πάνω αριστερά). Σαν επόμενο βήμα της δραστηριότητας, ο χρήστης μπορεί εκτελώντας μια εντολή (από την Ψηφίδα Logo) να πάρει τη γραφική κατανομή του πληθυσμού των επιλεγμένων χωρών σε μορφή ραβδογράμματος στην Ψηφίδα Καμβάς (κάτω αριστερά). Η αλληλουχία διαδικασιών που επιφέρει το τελικό αυτό αποτέλεσμα έχει ως εξής:

- Ο Χάρτης ενημερώνει τη Βάση Δεδομένων για το ποιές χώρες έχει επιλέξει ο χρήστης. Ως αποτέλεσμα, η τελευταία επιλέγει τις αντίστοιχες εγγραφές (records) του πίνακα «Χώρες».
- Ένα μικρό πρόγραμμα (script) στη Logo ζητά από τη Βάση Δεδομένων (μέσα από εντολές που η τελευταία υποστηρίζει) τις τιμές των πεδίων «Όνομα» και «Πληθυσμός» των επιλεγμένων εγγραφών (χωρών).
- Για κάθε ζευγος τιμών (όνομα χώρας και πληθυσμός) το script δίνει εντολές σχηματισμού μιας ράβδου στον Καμβά (μέσα από το

ρεπερτόριο γραφικών εντολών που ο τελευταίος υποστηρίζει).
Το script εντολών που επιτυγχάνουν αυτό το αποτέλεσμα είναι της τάξης των 20 γραμμών και μπορεί να στοιχειοθετηθεί από το συγγραφέα της δραστηριότητας (μη τεχνικό) με μικρή προσπάθεια.
Σ'ένα άλλο παράδειγμα που εικονίζεται παραπλεύρως, εντολές Logo χρησιμοποιούνται για την πλοήγηση ενός «Ταξιδιώτη» σ'έναν χάρτη του ιστορικού κέντρου της Αθήνας. Μ'αυτό τον τρόπο ο χρήστης έχει προγραμματιστικό έλεγχο της κίνησης και συμπεριφοράς του Ταξιδιώτη (επιπρόσθετα του άμεσου χειρισμού του από το User Interface του Χάρτη). Μπορεί για παράδειγμα να «ρωτήσει» τον Ταξιδιώτη για αντικείμενα τα οποία ανακαλύπτει στην πορεία του, για το τί βλέπει στο οπτικό του πεδίο, κλπ.

2.3 Πλεονεκτήματα

Πώς συγκρίνεται η προσέγγιση Αβάκιο από την οπτική γωνία του χρήστη σε σχέση με τις άλλες προγενέστερες μεθόδους που αναφέρθηκαν στην εισαγωγή;
1. Δίνεται στους εκπαιδευτικούς η δυνατότητα να συνθέσουν Μικρόκοσμους δικής τους έμπνευσης ή/και να τροποποιήσουν ήδη υπάρχοντες (χρησιμοποιώντας τα ίδια εργαλεία με τους αρχικούς κατασκευαστές). Κατά τη διαδικασία σύνθεσης Μικρόκοσμων οι σχεδιαστές μπορούν να σκέφτονται για την κατασκευή τους με βάση δομικά υλικά υψηλού επιπέδου, τις Ψηφίδες, οι οποίες αναπαριστούν γνώριμες οντότητες του γνωστικού τους χώρου, αντί να ξεκινούν από πρωτόγονα δομικά συστατικά (π.χ. αριθμούς και δομές δεδομένων κάποιας γλώσσας προγραμματισμού) στην προσπάθεια να κωδικοποιήσουν τις απαραίτητες έννοιες και αλγορίθμους. Οι Ψηφίδες προσφέρουν προ-πακεταρισμένη την απαιτούμενη συμπεριφορά, εκμετελλευόμενες τα χαρακτηριστικά του πεδίου εφαρμογών που αναπαριστούν.
2. Η ευελιξία σύνθεσης λειτουργικών διατάξεων (Μικρόκοσμων) που γίνεται δυνατή με τη χρήση Ψηφίδων, ανοίγει καινούργιες προοπτικές για την υλοποίηση συνδυαστικών λειτουργιών που δεν ήταν εφικτές με προηγούμενες μεθόδους.
3. Γίνεται δυνατή η σε μεγάλη κλίμακα επαναχρησιμοποίηση δομών και κατασκευών (Ψηφίδων και Μικρόκοσμων) με προφανή ωφέλη: μικρό χρόνο ανάπτυξης πρωτοτύπων (tapid prototyping), ελαχιστοποίηση προσπάθειας και χρόνου δηλαδή κόστους.
4. Ο ρόλος των τεχνικών «περιορίζεται» στο χώρο ειδίκευσής τους, στην κατασκευή δηλαδή υπολογιστικά άρτιων και αποτελεσματικών Ψηφίδων, δίνοντας την ελευθερία των σχεδιαστικών αποφάσεων για το εκπαιδευτικό λογισμικό στους ειδικούς του χώρου: οι εκπαιδευτικοί μπορούν να επιλέξουν τώρα μόνοι τους το περιεχόμενο, την παιδαγωγική προσέγγιση, τον τρόπο παρουσίασης, πλοκής και συμπεριφοράς ενός Μικρόκοσμου.
5. Οι Ψηφίδες μπορούν να σχεδιαστούν με τρόπο που να είναι αλληλοσυνεργάσιμες και ανταλλάξιμες ακόμα κι αν προέρχονται από διαφορετικούς κατασκευαστές, γεφυρώνοντας έτσι το κενό μεταξύ των μέχρι τώρα απομονωμένων «εφαρμογών» που δεν μπορούσαν παρά να χρησιμοποιηθούν μόνο για τις περιστάσεις για τις οποίες σχεδιάστηκαν. Γίνεται δηλαδή εφικτή η σύνθεση παραδοσιακά μεταξύ τους «ξένων» περιβαλλόντων σε ενιαία λειτουργικά σύνολα. Η δυνατότητα αυτή δίνει επιπλέον στους χρήστες των Ψηφίδων την ελευθερία επιλογής ανάμεσα σε διαφορετικούς κατασκευαστές (όπως την περίπτωση επιλογής συσκευών για τη σύνθεση ενός ηχοσυνόλου από συσκευές διαφορετικών κατασκευαστών).
Τα πλεονεκτήματα όμως αυτά δεν έρχονται ανέξοδα. Η Ψηφιδο-κεντρική προσέγγιση αναδεικνύει μια σειρά από θέματα που εντέλει ανάγονται σε τεχνικές απαιτήσεις ή/και αρχές ανάπτυξης Ψηφίδων.

2.4 Σχεδιαστικά θέματα

Γενικά ο σχεδιασμός επαναχρησιμοποιήσιμων Ψηφίδων ενέχει βαθμό δυσκολίας μιας τάξης μεγέθους μεγαλύτερο από το σχεδιασμό «μονολιθικών» εφαρμογών [0]. Ο σχεδιασμός μιας Ψηφίδας συνίσταται στη σύνθεση των χαρακτηριστικών (λειτουργικών, δομικών) που είναι κοινά μεταξύ των πολλαπλών εμφανίσεων της Ψηφίδας σε διαφορετικά σενάρια Μικρόκοσμων που προβλέπεται να χρησιμοποιηθεί. Πρόκειται μ'άλλα λόγια για μια

αφαιρετική διαδικασία, όπου από συγκεκριμένες εμφανίσεις (instances) συνάγεται μια γενικευμένη (generic) περιγραφή (class).

Granularity Ψηφίδων

Η ανθρώπινη αντίληψη είναι πολύ ευέλικτη στο να ορίζει «αντικείμενα», ανάλογα με τις επιδιώξεις και τα ενδιαφέροντα της «στιγμής» [0]. Άρα, πώς είναι δυνατό να προσδιοριστούν προκατασκευασμένα αντικείμενα τα οποία θα συνάδουν με τις κάθε φορά ανάγκες και το επίπεδο συλλογισμών των σχεδιαστών εκπαιδευτικών δραστηριοτήτων; Το κλειδί φαίνεται να είναι η ισορροπία μεταξύ δυο άκρων: από τη μια μεριά οι Ψηφίδες μπορεί να καταλήξουν να είναι πολύ γενικού σκοπού και χαμηλού νοηματικού επιπέδου ως προς την οντολογία της γνωστικής περιοχής στην οποία θα χρησιμοποιηθούν, πράγμα που συμβαίνει π.χ. με τα «κουμπιά» και άλλα αντικείμενα γενικού τύπου που συνήθως παρέχονται από τα συγγραφικά εργαλεία για τη δόμηση του user interface εφαρμογών. Από την άλλη μεριά, οι Ψηφίδες μπορεί να καταλήξουν να είναι «παραφορτωμένες» με λειτουργίες, και άρα σύνθετες και μη ευέλικτες.

Η απάντηση σ'αυτό το δίλημα δίνεται από τους σχεδιαστές Ψηφίδων με στάθμιση των απαιτήσεων του τελικού χρήστη των Ψηφίδων. Κι εδώ ακριβώς η εμπειρία δείχνει ότι η παραδοσιακή «ακολουθιακή» προσέγγιση στο σχεδιασμό λογισμικού (η ακολουθία δηλαδή απαιτήσεων-σχεδιασμού-υλοποίησης) είναι ανεπαρκής για την ανάδειξη των σωστών «συμβιβασμών» και την εξεύρευση της χρυσής τομής ως προς την οριοθέτηση της υφής και υπόστασης μιας Ψηφίδας.

Οι «απαιτήσεις» δεν είναι γνωστές παρά μόνον αφού οι εκπαιδευτικοί έχουν χρησιμοποιήσει κάποιο χονδροειδές πρωτότυπο και είναι σε θέση να σκεφτούν τις δυνατότητες και το πώς «θα μπορούσε» να είναι το ζητούμενο. Συνήθως λογισμικό και εκπαιδευτική πρακτική συν-εξελίσσονται: με την αλλαγή της τεχνολογίας και των δυνατοτήτων που καθίστανται εφικτές, οι ιδέες των εκπαιδευτικών για το «τί» και το «πώς», επίσης αλλάζουν.

Μια επαναληπτική διαδικασία σχεδιασμού που περιλαμβάνει διαδοχικά ενδιάμεσα πρωτότυπα και που υλοποιείται σε στενή συνεργασία μεταξύ σχεδιαστών Ψηφίδων (τεχνολόγων) και σχεδιαστών Μικρόκοσμων (εκπαιδευτικών), φαίνεται να έχει τα καλύτερα αποτελέσματα [0].

Συγκεκριμένα, η ακόλουθη διαδικασία έχει εκ πείρας αναδειχτεί ως η πλέον ενδεδειγμένη για τον προσδιορισμό της καταλληλότερης υφής και granularity Ψηφίδων:

1. Σ'ενα πρώτο στάδιο, σχεδιάζεται (στο χαρτί) ένα πρωτότυπο του λογισμικού σε συνεργασία με εκπαιδευτικούς, εστιάζοντας στα γενικά λειτουργικά χαρακτηριστικά.
2. Ακολουθεί το στάδιο της «Ψηφιδο-ποίησης», της αποσύνθεσης δηλαδή του επιθυμητού λογισμικού σε Ψηφίδες με το δυνατόν γενικευμένη μορφή, έτσι ώστε να είναι δυνατό να μπορούν να ικανοποιήσουν μια «οικογένεια» συναφών χαρακτηριστικών λογισμικού όπως αυτό που αρχικά προδιαγράφηκε. Τα κριτήρια για μια τέτοια αποσύνθεση περιλαμβάνουν τα ακόλουθα:
   I. Οι Ψηφίδες πρέπει να βρίσκονται εννοιολογικά στο επίπεδο των «ατόμων» -με την έννοια της μη παραπέρα αποσύνθεσης- του γνωστικού χώρου των χρηστών, θα πρέπει δηλαδή να μοντελοποιούν οντότητες, έννοιες, φαινόμενα, σχέσεις και συμπεριφορές του στοχευόμενου γνωστικού χώρου. Στην περίπτωση της γεωγραφίας για παράδειγμα, ο «χάρτης» είναι μια Ψηφίδα η οποία χρησιμοποιείται ως έχει, χωρίς να ενδιαφέρει η παραπέρα ανάλυσή της σε δομές και αλγόριθμους.
   II. Η επιλογή τέτοιων «ατόμων» θα πρέπει να στοχεύει στις το δυνατόν «ευρύτερες» οντότητες οι οποίες έχουν μεγαλύτερες πιθανότητες να παίξουν κυρίαρχο ρόλο στις κατασκευές των χρηστών τους.
   III. Η διασύνδεση μεταξύ των όποιων Ψηφίδων αποφασιστούν, θα πρέπει να μην επιφέρει κατασκευαστικό βάρος ή πολυπλοκότητα χρήσης στο επίπεδο της σύνθεσης Μικρόκοσμων.
3. Με βάση τις Ψηφίδες που διακρίθηκαν (και αφού αυτές αναπτυχθούν), (ανα)συντίθεται το αρχικά στοχευόμενο λογισμικό (Μικρόκοσμος) και δίνεται στους εκπαιδευτικούς για χρήση

και σχολιασμό. Παράλληλα, επιδεικνύεται η επιπρόσθετη δυνατότητα της σύνθεσης μιας σειράς από παρεμφερείς Μικρόκοσμους με πολύ μικρές παρεμβάσεις. Αν η διαδικασία του βήματος (2) ήταν σωστή, και παρήγαγε Ψηφίδες στο κατάλληλο νοηματικό επίπεδο, τότε οι εκπαιδευτικοί πολύ γρήγορα θα καταφέρουν να είναι σε θέση να σκέφτονται με βάση τα δομικά αυτά υλικά που αναπτύχθηκαν, και να φτιάχουν μόνοι τους ιδιο-κατασκευές, οι οποίες δεν είχαν κάν προβλεφτεί προτύτερα.

4. Ένας τέτοιος όμως πρώτος κύκλος είναι αμφίβολο ότι θα καταφέρει να παράξει το σύνολο των επιθυμητών Ψηφίδων, κι έτσι συνήθως ακολουθεί ένας δεύτερος γύρος (επανα)σχεδιασμού Ψηφίδων, αυτή τη φορά με βάση και τα αναλυτικά σχόλια που προκύτπουν από τη χρήση τους σε πραγματικές συνθήκες παρατήρησης σε σχολείο.

Συγκεκριμένα, από τη χρήση των Ψηφίδων για τη σύνθεση διαφόρων πρωτοτύπων, αναδεικνύονται οι επιπρόσθετοι βαθμοί ελευθερίας που απαιτούνται, τα σημεία όπου επαναληπτικές διαδικασίες ρουτίνας θα μπορούσαν να αυτοματοποιηθούν, όπως και ανάγκες που δεν είχαν καν αναδειχτεί στο παρελθόν. Αυτή η διαδικασία, συχνά οδηγεί στο «σπάσιμο» μιας Ψηφίδας σε περισσότερες Ψηφίδες μικρότερου granularity, ή σχεδιασμό επιπρόσθετων (νέων) Ψηφίδων, ή αλλαγή των δυνατοτήτων διασύνδεσης και προγραμματισμού των Ψηφίδων.

Διασυνδεσιμότητα

Οι Ψηφίδες πρέπει να κάνουν σαφείς τις δυνατότητες διασύνδεσής τους με άλλες, καθώς και τον τρόπο υλοποίησης των συνδέσεων. Περιβάλλοντα που επιτρέπουν σχεδόν οποιαδήποτε Ψηφίδα να συνδεθεί με οποιαδήποτε άλλη μέσω πλειάδας διαφορετικών τρόπων, καταλήγουν να είναι μη χρησιμοποιήσιμα από τους εκπαιδευτικούς λόγω της πολυπλοκότητας που επιφέρει το μεγάλο εύρος των ανοιχτών δυνατοτήτων.

Εδώ απαιτούνται τεχνικές για την απάλλειψη της νοητικής επιβάρυνσης. Τέτοιες είναι:

- Η ρητή «διαφήμιση» των δυνατοτήτων διασύνδεσης μέσω ενός απλού και συνεκτικά χρησιμοποιούμενου συμβολισμού, που υιοθετεί μεταφορές που είναι εύκολα αντιληπτές από τους χρήστες του.
- Η αυτοματοποίηση της συνδεσμολογίας Ψηφίδων όπου κάτι τέτοιο είναι εφικτό χωρίς να δημιουργεί διφορούμενες κατατάσεις (π.χ. η αυτόματη σύνδεση ενός χρονιστή σε όλα τα ρολόγια ενός Μικρόκσομου).
- Παροχή καθοδήγησης του χρήστη προς την κατεύθυνση του ορισμού συνδέσεων με βάση κάποια πρότυπα ή/και ευριστικές μεθόδους (π.χ. την καθοδήγηση του σχεδιαστή προς τη σύνδεση μιας οντότητας σε όλους τις Ψηφίδες που μπορούν να διαχειριστούν ή να οπτικοποιήσουν αυτή την οντότητα).

Υποστήριξη Προγραμματισιμότητας

Η δυνατότητα χειρισμού των Ψηφίδων με προγραμματιστικό τρόπο είναι πολύ σημαντικό χαρακτηριστικό και από μαθησιακής πλευράς όπως έχει ήδη αναφερθεί, αλλά και από πλευράς ανάπτυξης εκπαιδευτικού λογισμικού από τους Σχεδιαστές Δραστηριοτήτων. Ανικανότητα των Ψηφίδων να προσαρμοστούν σε καινούργιες απαιτήσεις μπορεί εύκολα να τις καταστήσει άχρηστες.

Η υποστήριξη προγραμματισιμότητας πρέπει να προσφέρεται σε διάφορα επίπεδα:

- Ανάπτυξη (από τον τελικό χρήστη) μικρών Ψηφίδων για κάλυψη «κενών» σε λειτουργίες που δεν υποστηρίζονται από τη βιβλιοθήκη υπαρχόντων Ψηφίδων.
- Διαμόρφωση ή και αλλαγή της συμπεριφοράς μιας Ψηφίδας για να ικανοποιήσει κάποια διάταξη που δεν είχε προβλεφθεί.
- Διασύνδεση Ψηφίδων με προγραμματιστικό τρόπο σε περιπτώσεις όπου η διασύνδεση δεν είναι εφικτή με άλλον τρόπο (π.χ. έλλειψη συμβατών μηχανισμών επικοινωνίας).

Ευχαριστίες

Η ανάπτυξη του Αβακίου χρηματοδοτείται στην παρούσα φάση από τα έργα: «Θρανίο» (ΕΠΕΤ ΙΙ , 1998-2000), "C3" (Esprit LTR, Experimental School Environments, 1998-2000), και «Έργα Επίδειξης» (ΕΠΕΑΕΚ 1996 –2000, Ενέργεια «Οδύσσεια»).

Αναφορές

Couclelis H.: 1992, "People Manipulate Objects (but Cultivate Fields): Beyond the Raster- Vector Debate in GIS", Theories and Models of Spatio-Temporal Reasoning in Geographic Space. Ed. A. U. Frank, I. Campari and U. Formentini. Pisa: Springer-Verlag.

Eisemberg M.: 1995, "Programmable Applications", SIGCHI Bulletin, Vol 27, No 2, April 1995

Fenton J., Beck K.: 1989, "Playground: An Object Oriented Simulation System with Agent Rules for Children of All Ages", OOPSLA '89 Proceedings, October 1-6, 1989.

Finzer W., Gould L.: 1984, "Programming by rehearsal", Byte, Vol. 9, No 6, June 1984.

Kaput, J., Roschelle, J.: 1996, "SimCalc 2nd Year Project Report", SimCalc project

Koutlis M., Kynigos Ch., Oikonomou A., Tsironis, G.: 1997, "Empowering Logo through a Component-Oriented Approach", προς δημοσίευση στο EuroLogo '97 conference, Αύγουστος '97.

Kynigos Ch., Koutlis M., Hadzilacos Th.: 1998, "Mathematics with component-oriented exploratory software", International Journal of Computers for Mathematical Learning (Seymour Papart (ed), Celia Hoyles, Richard Noss (co-ed)).

Nierstrasz O., Gibbs S., Tsichritzis D.: 1992, "Component-Oriented Software Development", Communications of the ACM, Sept. 1992, Vol. 35, No 9.

Reinhardt A.: 1995, "New Ways to Learn", Byte Magazine, pp 50-71

Report to the President on the Use of Technology to Strengthen K-12 Education in the United States (March 1997) http://www.whitehouse.gov/WH/EOP/OSTP/NSTC/PCAST/k-12ed.html

Reppening A.: 1995 "Agentsheets: A Medium for Creating Domain-Oriented Visual-Languages", IEEE Computer, March 1995.

Roschelle J. and Kaput J., 1996: "Educational software architecture and systemic impact: The promise of component software", Journal of Educational Computing Research, 14(3), 1996, pp. 217-228

Roschelle J., Koutlis M., Reppening A., etal, 1999: "Developing Educational Software Components", IEEE Computer, September 1999 Special Issue on Web based learning and collaboration. Vol 32 No 9, pp 50-58.

Solloway E., 1998: "No One Is Making Money in Educational Software", Communcations of the ACM, February 1998.

Solloway E.: 1991, "Quick: Where do the Computers Go?", Communications of the ACM, February 1991, Vol. 32, No 2.

Solloway E.: 1993, "Reading and writing in the 21st Century", Communications of the ACM, May 1993, Vol. 36, No 5.