ePublishing
ΕΘΝΙΚΟ ΚΕΝΤΡΟ ΤΕΚΜΗΡΙΩΣΗΣ
NATIONAL DOCUMENTATION CENTRE

Θέματα Επιστημών και Τεχνολογίας στην Εκπαίδευση

Τόμ. 12, Αρ. 2 (2019)

Συστήματα αξιολόγησης της Υπολογιστικής Σκέψης στην εκπαίδευση: Βιβλιογραφική επισκόπηση

Εμμανουήλ Πουλάκης, Παναγιώτης Πολίτης

doi: 10.12681/thete.39993

Βιβλιογραφική αναφορά:

Πουλάκης Ε., & Πολίτης Π. (2019). Συστήματα αξιολόγησης της Υπολογιστικής Σκέψης στην εκπαίδευση: Βιβλιογραφική επισκόπηση. *Θέματα Επιστημών και Τεχνολογίας στην Εκπαίδευση*, *12*(2), 99–119. https://doi.org/10.12681/thete.39993

Συστήματα αξιολόγησης της Υπολογιστικής Σκέψης στην εκπαίδευση: Βιβλιογραφική επισκόπηση

Εμμανουήλ Πουλάκης, Παναγιώτης Πολίτης
epoulakis@uth.gr, ppol@uth.gr

Παιδαγωγικό Τμήμα Δημοτικής Εκπαίδευσης, Πανεπιστήμιο Θεσσαλίας

Περίληψη. Το ενδιαφέρον για την Υπολογιστική Σκέψη (ΥΣ) αυξάνεται παγκοσμίως τα τελευταία χρόνια. Παρότι δημιουργείται υλικό για την ανάπτυξή της και προτείνονται προσεγγίσεις για την αξιολόγησή της, δεν υπάρχει ακόμα κοινά αποδεκτός ορισμός. Η αξιολόγηση της ΥΣ αποτελεί απαραίτητη προϋπόθεση για την επιτυχή εισαγωγή της σε προγράμματα σπουδών. Το άρθρο αυτό αποτελεί μια επισκόπηση των ερευνών που έχουν γίνει τα τελευταία χρόνια για την αξιολόγηση της ΥΣ. Οι έρευνες που μελετήθηκαν, ανάλογα με την προσέγγιση που χρησιμοποιούν, κατηγοριοποιήθηκαν σε τρεις ομάδες: α) αξιολόγηση της ΥΣ σε συγκεκριμένα προγραμματιστικά περιβάλλοντα, β) με κριτήρια αξιολόγησης ΥΣ και/ή ψυχομετρικά εργαλεία, και γ) με πολλαπλές μεθόδους αξιολόγησης της ΥΣ. Τα κυριότερα προβλήματα στην αξιολόγηση της ΥΣ είναι η αδυναμία κάλυψης όλων των εννοιών κι ηλικιών των μαθητών και η σχετική ανεπάρκεια των εργαλείων να αξιολογήσουν αυτόνομα κι επαρκώς την ΥΣ. Από τη βιβλιογραφία προκύπτουν προτάσεις για αξιολόγηση της ΥΣ, ενώ η χρήση πολλαπλών μεθόδων αξιολόγησης συνεχίζει να είναι η πλέον ενδεδειγμένη και έγκυρη προσέγγιση.

Λέξεις κλειδιά: υπολογιστική σκέψη, αξιολόγηση, προγραμματισμός

Εισαγωγή

Η έννοια της Υπολογιστικής Σκέψης (ΥΣ) ήρθε στο προσκήνιο το 2006 μέσω της Wing (2006), η οποία προχώρησε στην πρώτη σύγχρονη προσπάθεια ορισμού της. Η ΥΣ «*περιλαμβάνει την επίλυση προβλημάτων, τον σχεδιασμό συστημάτων και την κατανόηση της ανθρώπινης συμπεριφοράς, βασιζόμενη σε έννοιες που είναι θεμελιώδεις για την Επιστήμη των Υπολογιστών*» (Wing, 2006, σ. 33). Ως όρος έχει αναφερθεί από τον Papert το 1980 (Papert, 1993), περιγράφοντας την αλλαγή στις διεργασίες σκέψης που θα επέφερε στη μαθηματική εκπαίδευση η χρήση των υπολογιστών. Οι Grover & Pea (2013) σχολιάζουν την αρχική προέλευση της ιδέας της ΥΣ, κι αναφέρονται στον Alan Perlis στη δεκαετία του 1960, ο οποίος υποστήριζε ότι οι φοιτητές όλων των επιστημών θα έπρεπε να μάθουν να προγραμματίζουν, καθώς και στον Seymour Papert και την διαδικαστική σκέψη του (procedural thinking) μέσω του προγραμματισμού στη LOGO.

Οι Cuny, Snyder & Wing (2010, αναφορά στο Wing, 2011, σ. 1) προχώρησαν σε νέο ορισμό της ΥΣ ως: «*οι διεργασίες σκέψης που σχετίζονται με τη διαμόρφωση προβλημάτων και των λύσεών τους, έτσι ώστε οι λύσεις να αναπαρίστανται σε μορφή που μπορεί να εκτελεστεί αποτελεσματικά από έναν πράκτορα επεξεργασίας της πληροφορίας*». Ο ορισμός αυτός είναι παρόμοιος με αυτόν του Aho (2012, σ. 832), ο οποίος συγκρίνοντας τις έννοιες του υπολογισμού και της ΥΣ, αναφέρει: «*Θεωρούμε την υπολογιστική σκέψη ως τις γνωστικές διεργασίες που σχετίζονται με το μετασχηματισμό προβλημάτων έτσι ώστε οι λύσεις τους να μπορούν να αναπαρασταθούν με υπολογιστικά βήματα και αλγόριθμους*».

Θεωρητικό πλαίσιο

Ορισμός υπολογιστικής σκέψης

Ο ορισμός της ΥΣ έχει απασχολήσει επίσης οργανισμούς, φορείς κι ενώσεις σε παγκόσμιο επίπεδο. Έτσι, οι ISTE και CSTA (2011) δίνουν ένα λειτουργικό ορισμό της ΥΣ για τη γενική εκπαίδευση, στον οποίο αναφέρουν και στάσεις, ενώ το 2012 η Royal Society προχωρά σε έναν ακόμα ορισμό της ΥΣ -σε μια έκθεσή της που έδωσε ιδιαίτερη ώθηση στην αναμόρφωση των προγραμμάτων σπουδών στο Ηνωμένο Βασίλειο και την ενσωμάτωση του υπολογισμού και της ΥΣ σε αυτά- σύμφωνα με τον οποίο η ΥΣ «*είναι η διαδικασία αναγνώρισης υπολογιστικών πτυχών στον κόσμο που μας περιβάλλει και η εφαρμογή εργαλείων και τεχνικών από την Επιστήμη των Υπολογιστών για την κατανόηση και την ερμηνεία τόσο των φυσικών, όσο και των τεχνητών συστημάτων και διεργασιών*» (Royal Society, 2012, σ. 29). Σύμφωνα με ένα μοντέλο της ΥΣ, οι συνιστώσες της μπορούν να διακριθούν σε: έννοιες (concepts), πρακτικές (practices) κι οπτικές (perspectives) (Brennan & Resnick, 2012· Computational Thinking with Scratch, n.d.). Το ίδιο προτείνουν κι οι Lye & Koh (2014) στην ανασκόπησή τους για τη διδασκαλία και την ανάπτυξη ΥΣ μέσω προγραμματισμού.

Οι Barr & Stephenson (2011), μελετώντας τη δημιουργία ορισμού της ΥΣ, καταλήγουν σε ένα σύνολο βασικών εννοιών, αλλά και δεξιοτήτων για την ΥΣ. Οι Selby & Woolard (2013) μελετώντας και συγκρίνοντας υπάρχουσες προτάσεις, βάσει του κοινού τους τόπου, επίσης προχωρούν σε έναν ακόμα ορισμό, ο οποίος περιλαμβάνει τη διεργασία σκέψης και τις έννοιες: αφαίρεση, αποσύνθεση, αλγοριθμική σκέψη, αξιολόγηση και γενίκευση. Μέχρι σήμερα δεν υπάρχει καθολικά αποδεκτός ορισμός της ΥΣ, ενώ σημαντικές διαστάσεις της ΥΣ, όπως η συστημική σκέψη κι η ψηφιακή πολιτειότητα χρειάζεται να μελετηθούν (Fessakis, Mavroudi & Prantsoudi, 2017). Η ποικιλία ορισμών οδηγεί την Ευρωπαϊκή Επιτροπή (Bocconi et al., 2016· European Commission, 2016), στο να προσδιορίσει ένα σύνολο κοινά εμφανιζόμενων κι αποδεκτών εννοιών για την ΥΣ, και οι οποίες είναι: αφαίρεση, αλγοριθμική σκέψη, αυτοματοποίηση, αποσύνθεση, αποσφαλμάτωση και γενίκευση.

Πρωτοβουλίες με παγκόσμια απήχηση, όπως είναι το CS Unplugged (University of Canterbury, n.d.), το Computer Science Education Research Group στο Πανεπιστήμιο Canterbury της Νέας Ζηλανδίας ή το Computing At School (CAS) (Computing At School, n.d.), το οποίο ξεκίνησε στο Ηνωμένο Βασίλειο για την υποστήριξη της διδασκαλίας της επιστήμης υπολογιστών στα σχολεία, χρησιμοποιούν συγκεκριμένες έννοιες για την ΥΣ. Στο CS Unplugged οι δραστηριότητες πλέον συνδέονται με την ΥΣ, όπου χρησιμοποιούνται έξι βασικές συνιστώσες (Bell & Lodi, 2019· University of Canterbury, n.d.): αλγοριθμική σκέψη, αφαίρεση, αποσύνθεση, γενίκευση και πρότυπα, αξιολόγηση και λογική. Οι Bell & Lodi (2019) αναφέρουν τις έξι αυτές διεργασίες ΥΣ βασιζόμενοι στις πέντε πρώτες που προτείνουν ήδη οι Selby & Woollard (2013), και εξηγώντας την προσθήκη της έκτης, της λογικής. Φαίνεται να υπάρχει σύμπνοια απόψεων για τις έξι αυτές έννοιες με το CAS, καθώς το τελευταίο δημοσίευσε έναν οδηγό για εκπαιδευτικούς για την ΥΣ (Computing At School, 2015), όπου αναφέρονται οι έξι αυτές έννοιες ως συνιστώσες της ΥΣ: λογική, αλγοριθμική σκέψη, αποσύνθεση, γενίκευση και πρότυπα, αφαίρεση και αναπαραστάσεις, και τέλος, αξιολόγηση. Το CAS αναφέρεται επίσης περαιτέρω σε τεχνικές και προσεγγίσεις. Ο ιστότοπος Teaching London Computing (n.d.), ο οποίος αποτελεί πρωτοβουλία των Queen Mary University of London και King's College London, αναφέρεται επίσης στον οδηγό της ΥΣ του CAS (Computing at School, 2015). Ο διαγωνισμός Bebras τέλος, εναρμονίζεται με τις προτεινόμενες έννοιες των Selby & Woollard (Dagienė & Sentance, 2016).

Υπάρχουν λοιπόν αρκετές προσπάθειες ορισμού της ΥΣ, οι οποίες προσεγγίζουν πολύπλευρα το ζήτημα, χωρίς να υπάρχει ακόμα κοινά αποδεκτός ορισμός, ενώ από την

άλλη πολλές χώρες, έχοντας αναγνωρίσει τη σπουδαιότητα της ΥΣ, έχουν ήδη εδώ και χρόνια προχωρήσει στην αναμόρφωση των προγραμμάτων σπουδών τους στην τυπική εκπαίδευση, εισάγοντας την ΥΣ σε αυτά (Heintz, Mannila & Färnqvist, 2016). Μια ενδιαφέρουσα ανάλυση σχετικά με τη διαμόρφωση εννοιών της ΥΣ, καθώς και το εύρος που καλύπτουν σε προγράμματα σπουδών, γίνεται από τους Fessakis et al. (2018).

Αξιολόγηση Υπολογιστικής σκέψης

Όπως όμως αναφέρουν οι Grover & Pea (2013), για την ανάπτυξη της ΥΣ μέσα από τα προγράμματα σπουδών της πρωτοβάθμιας και δευτεροβάθμιας εκπαίδευσης, απαραίτητη προϋπόθεση είναι η αξιολόγησή της. Επίσης, για να κριθεί η αποτελεσματικότητα οποιουδήποτε προγράμματος σπουδών που ενδεχομένως στοχεύει στην ανάπτυξη της ΥΣ, οι ίδιοι συγγραφείς αναφέρουν ότι χρειάζεται να υπάρξουν εργαλεία που αξιολογούν έγκυρα την αποκτώμενη γνώση των μαθητών. Τα τελευταία χρόνια μελετήθηκαν τρόποι αξιολόγησης της ΥΣ, ενώ ερευνητές αναφέρουν ότι η αξιολόγηση της ΥΣ βρίσκεται σε πρώιμο στάδιο και δεν καλύπτει όλο το φάσμα της, ούτε και όλες τις ηλικιακές ομάδες (Bocconi et al., 2016· Brennan & Resnick, 2012). Οι Cutumisu, Adams & Lu (2019) αναφέρουν την έλλειψη έγκυρων εργαλείων αξιολόγησης της ΥΣ, αλλά και την έλλειψη συστηματικής οργάνωσης των αξιολογήσεων ΥΣ. Αντίστοιχα, οι Mueller et al. (2017) αναφέρουν την έλλειψη αξιόπιστων εργαλείων αξιολόγησης ΥΣ και τονίζουν την δυσκολία που αυτή η έλλειψη συνιστά για την ανάπτυξη της ΥΣ μέσα από τα προγράμματα σπουδών της τυπικής εκπαίδευσης, σύμφωνα με διεθνείς οργανισμούς όπως ο ACM και ο CSTA.

Όπως αναφέρουν οι Lye & Koh (2014), η ΥΣ περιλαμβάνει τη χρήση εννοιών επιστήμης υπολογιστών κι η ενασχόληση με τον προγραμματισμό έχει ως αποτέλεσμα την εντρύφηση των μαθητών με την ΥΣ. Έτσι αρχικά, λόγω της σύνδεσης της ΥΣ με την επιστήμη των υπολογιστών, για την αξιολόγησή της έχουν χρησιμοποιηθεί προϋπάρχουσες μέθοδοι αξιολόγησης προγραμμάτων σπουδών της επιστήμης των υπολογιστών, ειδικά για τον προγραμματισμό και την αλγοριθμική σκέψη. Στο παρελθόν όμως εργαλεία αξιολόγησης της ΥΣ απέτυχαν (Ambrósio, Georges, & Xavier, 2014), ενώ σε ό,τι αφορά στη δευτεροβάθμια εκπαίδευση, φαίνεται ότι υπάρχει έλλειψη εργαλείων αξιολόγησης ΥΣ και συχνά εκπαιδευτικοί επιχειρούν την αξιολόγησή της με ίδια εργαλεία, χωρίς όμως πάντα έγκυρα αποτελέσματα (Yadav et al., 2015). Προφανώς, η κατασκευή εργαλείων αξιολόγησης της ΥΣ είναι μια απαιτητική εργασία που δεν μπορεί να γίνεται εμπειρικά, αλλά ούτε να υιοθετεί εργαλεία αξιολόγησης που σχεδιάστηκαν για άλλες διεργασίες, χωρίς την αναγκαία αναπροσαρμογή και έλεγχο.

Η έρευνα

Σκοπός & στόχοι

Σκοπός της παρούσας έρευνας είναι η διερεύνηση των προσεγγίσεων για την αξιολόγηση της ΥΣ. Η παρούσα έρευνα συνιστά μια βιβλιογραφική επισκόπηση στο θέμα της αξιολόγησης της ΥΣ και βασίζεται στις αρχές της ποιοτικής ανάλυσης περιεχομένου (Bryman, 2008). Για την ανάλυση περιεχομένου και τον ορισμό των κατηγοριών που θα προκύψουν ακολουθήθηκε η προσέγγιση των έξι βημάτων που προτείνει ο Altheide (2004, αναφορά στο Bryman, 2008), ώστε σταδιακά να προκύψουν οι κατηγορίες. Σκοπός είναι αφενός η διερεύνηση της εξέλιξης της αξιολόγησης της ΥΣ, ο έλεγχος αν βρίσκεται σε πρώιμο στάδιο ή αν έχει ωριμάσει, αν καλύπτει πλέον όλες τις ηλικιακές ομάδες, κι αφετέρου η αποτύπωση των σύγχρονων ερευνητικών τάσεων, όπως διαμορφώνονται την τελευταία δεκαετία.

Πίνακας 1. Δείγμα έρευνας ανά αναζήτηση σε βάση δεδομένων

Βάση δεδομένων	Αποτελέσματα	Αποκλεισμός από το δείγμα	Ενσωμάτωση στο δείγμα
ACM Digital Library	179	138	41
IEEE Xplore Digital Library	53	40	13
Elsevier Science Direct	175	162	13
Springer Link	540	525	15
Σύνολο	**947**	**865**	**82**

Συγκεκριμένα οι στόχοι της βιβλιογραφικής μας επισκόπησης αφορούν στη διερεύνηση των προσεγγίσεων για την αξιολόγηση της ΥΣ:

- ως προς τις παραμέτρους που απαιτούνται για την εφαρμογή των εν λόγω προσεγγίσεων (π.χ. ηλικία, συγκεκριμένο προγραμματιστικό περιβάλλον, κ.ά.)
- ως προς τον τρόπο με τον οποίο υλοποιείται η αξιολόγηση (π.χ. αυτοματοποιημένη, κ.ά.).

Μέθοδος

Για την έρευνα και τη συλλογή των κατάλληλων άρθρων, έγινε αναζήτηση σε τέσσερις βάσεις δεδομένων για τις οποίες υπήρχε τεχνική ευχέρεια πρόσβασης: α) ACM Digital Library, β) Elsevier Science Direct, γ) IEEE Xplore Digital Library και δ) Springer Link, μέσω της ακαδημαϊκής πρόσβασης που παρέχεται από το Πανεπιστήμιο Θεσσαλίας. Για την αναζήτηση χρησιμοποιήθηκαν οι αγγλικές λέξεις-κλειδιά "computational thinking" και "assessment" ή "assessing" χρησιμοποιώντας το λογικό σχήμα «"COMPUTATIONAL THINKING" AND ("ASSESSMENT" OR "ASSESSING")», καθώς είχε παρατηρηθεί ότι σε κάποιους τίτλους άρθρων χρησιμοποιείται το "assessing". Τα άρθρα ελέγχθηκαν αρχικά από τους τίτλους, τις λέξεις-κλειδιά και τις περιλήψεις τους. Ο μεγάλος αρχικός αριθμός αποτελεσμάτων ήταν αναμενόμενος λόγω του μεγάλου χρονικού εύρους (τελευταία 10ετία). Από τη βάση Springer Link, σε σχέση με τις άλλες βάσεις, αναδύθηκε πολύ μεγαλύτερος αριθμός αποτελεσμάτων, τα οποία τελικά δεν ήταν συναφή με την αναζήτηση, εμφανίστηκαν για παράδειγμα αποτελέσματα που είχαν να κάνουν με τους όρους "digital literacy", "ICT literacy" κι όχι με την ΥΣ. Για το λόγο αυτό παρατηρείται το γεγονός να έχουν συμπεριληφθεί τόσα λίγα τεκμήρια στο δείγμα συγκριτικά με το μεγάλο αρχικό αριθμό αποτελεσμάτων της συγκεκριμένης αναζήτησης.

Η αρχική αναζήτηση έγινε για την τρέχουσα δεκαετία, δηλαδή τα έτη 2010-2019. Σε κάποιες περιπτώσεις, για λόγους εμβάθυνσης κι ορθότερης παρουσίασης των αποτελεσμάτων, μέσω των αναφορών των άρθρων που επιλέχθηκαν στην αρχική φάση αναζητήθηκαν επιπλέον άρθρα. Ο αριθμός των τεκμηρίων που αποτέλεσαν το αρχικό δείγμα της έρευνας τελικά ανήλθε σε ογδόντα δύο (82), όπως φαίνεται και στον Πίνακα 1. Ως μονάδα ανάλυσης θεωρείται το εκάστοτε άρθρο.

Ανάλυση Δεδομένων

Η ανάλυση των άρθρων βάσει της ανάλυσης περιεχομένου έγινε λαμβάνοντας υπόψη συγκεκριμένες προκαθορισμένες παραμέτρους, αλλά και παραμέτρους που προέκυψαν κατά τη διάρκεια της έρευνας, σύμφωνα με το σκοπό αυτής.

Τα άρθρα αρχικά ελέγχθηκαν στη βάση της ηλικίας των μαθητών τους οποίους στοχεύουν τα σχήματα αξιολόγησης. Παρατηρήθηκε έντονο το φαινόμενο να μην γίνεται σαφής

αναφορά στις προτεινόμενες ηλικιακές ομάδες, αλλά να υπάρχει συνήθως εφαρμογή σε ένα μικρό ερευνητικό δείγμα επιλογής των ερευνητών, που παρέπεμπε σε βολική/συμπτωματική δειγματοληψία και δείγμα ευχέρειας. Από την βιβλιογραφική ανασκόπηση φαίνεται πως ελάχιστα εργαλεία έχουν επικυρωθεί κι έχουν περάσει ελέγχους εγκυρότητας και αξιοπιστίας. Στα αποτελέσματα της παρούσας έρευνας, παρουσιάζονται σύντομα οι ερευνητικές προτάσεις, ενώ γίνεται αναφορά σε ηλικιακές ομάδες κι έννοιες αξιολόγησης.

Η επόμενη εξέταση των άρθρων έγινε με βάση την προσέγγιση που ακολουθούν. Προέκυψε ένας μεγάλος αριθμός προσεγγίσεων που είναι σχεδιασμένες πάνω σε συγκεκριμένα προγραμματιστικά περιβάλλοντα, αρκετές προσεγγίσεις που χρησιμοποιούν κλίμακες ή ψυχομετρικά εργαλεία, και κάποιες προσεγγίσεις που ακολουθούν μικτή μεθοδολογία με πολλαπλές μεθόδους αξιολόγησης, οι οποίες δεν βασίζονται σε ένα μόνο εργαλείο. Στα αποτελέσματα παρουσιάζονται αναλυτικά οι κατηγορίες που προέκυψαν.

Τέλος, μια ακόμα παράμετρος που ελέγχθηκε είναι αν υπάρχει σύνδεση της προτεινόμενης μεθοδολογίας αξιολόγησης με τη διδασκαλία, αν δηλαδή λαμβάνονται υπόψη οι δραστηριότητες κατά τη διάρκεια διδασκαλίας ή η μεθοδολογία αποτελεί αυτόνομο εργαλείο αξιολόγησης της ΥΣ. Βρέθηκαν ελάχιστα εργαλεία που αυτόνομα αξιολογούν την ΥΣ, κυρίως στην κατηγορία των κριτηρίων/ψυχομετρικών εργαλείων, ενώ σε επόμενο επίπεδο ελέγχθηκε η επάρκεια των αποσυνδεμένων από συγκεκριμένες διδασκαλίες εργαλείων να αξιολογήσουν την ΥΣ.

Μετά την ανίχνευση των προαναφερόμενων παραμέτρων (ηλικία, προσέγγιση, σύνδεση με διδασκαλία) και την συνεπαγόμενη δημιουργία κατηγοριών, αποφασίστηκε η σύντομη παρουσίαση των εργαλείων/μεθοδολογιών αξιολόγησης για τις κατηγορίες που προέκυψαν.

Αποτελέσματα

Οι Román-González, Moreno-León & Robles (2017) κατατάσσουν τις υπάρχουσες προσεγγίσεις στην αξιολόγηση της ΥΣ σε πέντε κατηγορίες: α) αθροιστική, β) διαμορφωτική-επαναλαμβανόμενη, η οποία παρέχει ανατροφοδότηση, γ) μεταφοράς δεξιοτήτων, δ) κλιμάκων πεποιθήσεων-στάσεων και ε) λεξιλογίου. Ακολούθως προσθέτουν δύο ακόμα κατηγορίες (Román-González et al., 2019): στ) διαγνωστικά εργαλεία και ζ) εργαλεία εξόρυξης δεδομένων ΥΣ. Στην παρούσα έρευνα, σύμφωνα με τα ερωτήματα που έχουν τεθεί, οι κατηγορίες αξιολόγησης της ΥΣ που προκύπτουν από την ανάλυση περιεχομένου είναι:

α) *Σε συγκεκριμένα προγραμματιστικά περιβάλλοντα*, η οποία συνδέεται άμεσα με το περιβάλλον υλοποίησης του εκάστοτε έργου, που μπορεί να είναι πρόγραμμα ή εκπαιδευτικό παιχνίδι, ενώ σε κάποιες προσεγγίσεις αυτοματοποιείται πλήρως. Παρατηρούνται διαφοροποιήσεις όταν ο στόχος είναι η *κατασκευή παιχνιδιών*, η οποία εμφανίζεται ως υποκατηγορία, καθώς οι έννοιες αξιολόγησης ΥΣ σε μερικές περιπτώσεις κατασκευής παιχνιδιών διαφοροποιούνται από τους ερευνητές, κι εισάγονται νέοι τομείς αξιολόγησης που δεν τους συναντάμε στο υπόλοιπο φάσμα της ΥΣ, όπως η μηχανική παιγνιδιών.

β) *Με κριτήρια αξιολόγησης ΥΣ ή/και ψυχομετρικά εργαλεία*, στα οποία εντάσσονται οι κλίμακες πεποιθήσεων-στάσεων, που αναφέρουν οι Román-González et al. (2017), και οι προσπάθειες σύνδεσης της ΥΣ με μεταγνωστικούς παράγοντες ή με χαρακτηριστικά της προσωπικότητας, στην προσπάθεια να μελετηθεί η ΥΣ από τη σκοπιά της ψυχολογίας.

γ) *Με πολλαπλές μεθόδους αξιολόγησης*, οι οποίες απαρτίζονται από φάκελους με ολοκληρωμένα έργα, παρατήρηση πεδίου, συνεντεύξεις με τους μαθητές, κ.ά. και διαφοροποιούνται ανάλογα με το προτεινόμενο ερευνητικό πλαίσιο. Στις προσεγγίσεις της ΥΣ μέσω STEM παρατηρείται επίσης διαφοροποίηση στις αξιολογούμενες έννοιες και

πρακτικές, εμφανίζονται π.χ. οι προσομοιώσεις (simulations). Για το λόγο αυτό θα μπορούσε να θεωρηθεί ως χωριστή υποκατηγορία, όμως στην παρούσα έρευνα περιλαμβάνεται στις πολλαπλές μεθόδους αξιολόγησης.

Οι κατηγορίες δεν είναι πάντα ανεξάρτητες μεταξύ τους, π.χ. στην κατηγοριοποίηση των Román-González et al. (2017) μπορεί ένα τυποποιημένο εργαλείο σε προγραμματιστικό περιβάλλον να χρησιμοποιηθεί τόσο για διαμορφωτική όσο και για τελική αξιολόγηση, ενώ οι πολλαπλές μέθοδοι αξιολόγησης, που προτείνονται ως κατηγορία στην παρούσα έρευνα, μπορεί να αναφέρονται και σε συγκεκριμένα προγραμματιστικά περιβάλλοντα.

Αξιολόγηση ΥΣ σε συγκεκριμένα προγραμματιστικά περιβάλλοντα

Ο προγραμματισμός συναντάται συνεχώς στις προσεγγίσεις οικοδόμησης ΥΣ, έτσι έχει γίνει αρκετή δουλειά στην αξιολόγηση εννοιών ΥΣ μέσω προγραμματισμού, ενασχόλησης με παιχνίδια κ.ά. Προσπάθειες γίνονται για την αυτοματοποίηση διαδικασιών, ενώ η ειδική υποκατηγορία των παιχνιδιών έχει οδηγήσει σε αναγνώριση προτύπων κι αναφορά σε έννοιες που ίσως ξενίζουν, καθώς δεν συμπίπτουν με τις συχνά χρησιμοποιούμενες έννοιες της ΥΣ. Ερευνητές έχουν αναγνωρίσει προγραμματιστικά πρότυπα σε παιχνίδια, και βάσει αυτών έχουν προχωρήσει σε συστήματα αξιολόγησης ΥΣ.

Στο περιβάλλον AgentSheets/AgentCubes η ΥΣ αξιολογείται αυτόματα χρησιμοποιώντας γράφους (Computational Thinking Pattern Graph - CTPG), που εκφράζουν οπτικά το βαθμό επίτευξης προγραμματιστικών στόχων-προτύπων (Koh et al., 2010), τεχνικών δηλαδή που εμφανίζονται στη δημιουργία παιχνιδιού (Πίνακας 2) κι ονομάζονται Σπείρα Προτύπων ΥΣ (Basawapatna, Koh & Repenning, 2010· Koh et al., 2014). Μέσω των γράφων αξιολογείται κι η μεταφορά γνώσης. Στην ίδια πλατφόρμα (SGDA), το σύστημα Real Time Evaluation and Assessment of Computational Thinking (REACT) παράγει αυτοματοποιημένα γράφους CTPG (Basawapatna, Repenning & Koh, 2015· Koh et al., 2014), δίνοντας άμεση εικόνα για την πορεία της κατασκευής και βοηθώντας στην ενδιάμεση αξιολόγηση.

Το εργαλείο Fairy Performance Assessment (Werner et al., 2012) αξιολογεί σε περιβάλλον Alice την αλγοριθμική σκέψη, την αφαίρεση και τη μοντελοποίηση (Carnegie Mellon, n.d.), όχι όμως την κλιμάκωση (Πίνακας 2). Χρησιμοποιεί σενάρια αξιολόγησης στο AgentCubes (Webb, 2010), ενώ η αξιολόγηση γίνεται με διορθώσεις/προσθήκες σε δοσμένο κώδικα.

Για τον προγραμματισμό παιχνιδιών ερευνητές αναφέρουν την έννοια της Υπολογιστικής Μάθησης - ΥΜ (Computational Learning-CL), η οποία συνδέεται με την ΥΣ (Werner, Denner & Campe, 2015). Για την αξιολόγηση της ΥΜ προτείνουν το Game Computational Sophistication (GCS), το οποίο χρησιμοποιεί προγραμματιστικές κατασκευές, πρότυπα και μηχανική παιχνιδιών στο Alice (Πίνακας 2).

Το μοντέλο Progression of Early Computational Thinking (PECT) στοχεύει σε τάξεις δημοτικού σχολείου κι αξιολογεί προγράμματα σε Scratch (Seiter & Foreman, 2013). Οι έννοιες ΥΣ (Πίνακας 2) που αξιολογεί είναι υποσύνολο αυτών που προτείνει η CSTA (Barr & Stephenson, 2011), ενώ χρησιμοποιεί και ρουμπρίκες αξιολόγησης, όπως αυτή των Wilson, Hainey & Connolly (2012).

Το Dr. Scratch είναι περιβάλλον αυτοματοποιημένης αξιολόγησης της ΥΣ (Moreno & Robles, 2014· Moreno-León & Robles, 2015a· 2015b). Αξιολογεί αυτόματα ένα πρόγραμμα Scratch αποδίδοντας ένα βαθμό ΥΣ βάσει προγραμματιστικών εννοιών (Πίνακας 2), αλλά κι αναλύει κακές προγραμματιστικές τακτικές. Προτείνεται η υιοθέτηση φακέλου εργασιών για κάθε χρήστη, καθώς μια μόνο εργασία δεν δίνει σίγουρα αποτελέσματα αλλά ούτε πληροφορίες σχετικά με την εξέλιξη της ΥΣ του χρήστη. Δεν μπορεί να χρησιμοποιηθεί επαρκώς μόνο αυτό, καθώς θα μπορούσε κανείς με ένα θεωρητικά άχρηστο πρόγραμμα να πάρει πολύ υψηλό τελικό βαθμό (Moreno-León & Robles, 2015a).

Πίνακας 2. Εργαλεία αξιολόγησης ΥΣ σε προγραμματιστικά περιβάλλοντα

Μέθοδος	Είσοδος	Έννοιες	Ηλικίες
Computational Thinking Pattern (CTP) graph Real Time Evaluation & Assessment of CT (REACT)	Agent Sheets, Agent Cubes	Πρότυπα: παραγωγή, απορρόφηση, σύγκρουση, μεταφορά, ώθηση, έλξη, διάχυση, αναρρίχηση λόφου, έλεγχος δείκτη	Middle schools, K-12
Progression of Early Computational Thinking (PECT) Model	Scratch	Διαδικασίες και αλγόριθμοι, αποσύνθεση προβλήματος, παραλληλισμός και συγχρονισμός, αφαίρεση, αναπαράσταση δεδομένων	Grades 1-6 (6-11 ετών)
Dr. Scratch		Αφαίρεση και αποσύνθεση προβλήματος, παραλληλισμός, λογική σκέψη, συγχρονισμός, έλεγχος ροής, αλληλεπίδραση με το χρήση, αναπαράσταση δεδομένων	Grades 5-10 (10-15 ετών)
Wilson, Hainey & Connolly		Προγραμματιστικές έννοιες, οργάνωση του κώδικα, σχεδιασμός για επαναχρησιμοποίηση	Grades 4-7 (9-12 ετών)
Scratch Analysis Tool (SAT)		Αφαίρεση - αποσύνθεση, παραλληλισμός, λογική σκέψη, συγχρονισμός, έλεγχος ροής, διάδραση χρήστη, αναπαράσταση δεδομένων	Δεν αναφέρουν
Ninja Code Village (NCV)		Συνθήκες, βρόχοι, διαδικασία, κοινή διαδικασία, δεδομένα, γεγονότα, παρ/σμός, Διεπαφή χρήστη	Δημοτικό
Functional Understanding Navigator! (FUN!)		Παραλληλισμός, λογική σκέψη, συγχρονισμός, επανάληψη, πρότυπα – γενίκευση	K-12, Grades 5-8 (10-13 ετών)
Fairy Performance Assessment	Alice	Αλγοριθμική σκέψη, αποτελεσματική χρήση της αφαίρεσης και μοντελοποίησης	Middle schools, K-12
Game Computational Sophistication (GCS)		Προγραμματιστικές κατασκευές, πρότυπα, μηχανική παιχνιδιών	Middle schools
Rowe & Cunningham	Zoombinis	Αποσύνθεση, αναγνώριση προτύπων, αφαίρεση, αλγοριθμικός σχεδιασμός	Δημοτικό-Γυμνάσιο, ειδικοί
Commutative Assessment	Blocks, Κείμενο	Βασικά (μεταβλητές, αναθέσεις, κτλ), επιλογή, ορισμένες/αόριστες επαναλήψεις, παράμετροι συναρτήσεων – διαδικασιών	Grades 9-12 (14-17 ετών)
Mobile Computational Thinking (MCT)	App Inventor	Διαστάσεις ΥΣ (ονοματολογία, διαδικαστική αφαίρεση, μεταβλητές, βρόχοι, συνθήκες, λίστες) –8 επιπλέον διαστάσεις Κινητής ΥΣ (Mobile CT)	Δεν αναφέρουν
Quiz Maker – Quizly		Δομές του App Inventor	14-16 ετών, Φοιτητές (18-22 ετών)

Για το Dr. Scratch έχουν ήδη γίνει ή βρίσκονται σε εξέλιξη, αρκετοί έλεγχοι εγκυρότητας (Moreno-León, Robles & Román-González, 2016· Moreno-León et al., 2017· Moreno-León, Robles & Román-González, 2017). Χρησιμοποιείται σε έρευνες, αν και κάποιες αναφέρουν αδυναμία εμφάνισης της πλήρους εικόνας μαθητή, ειδικά σε πολύπλοκους σχεδιασμούς, ενώ τα αποτελέσματά του διαφοροποιούνται από αυτά ποιοτικής αξιολόγησης (Hoover et al., 2016). Χρησιμοποιείται επίσης για ενδιάμεση αξιολόγηση κι όχι μόνο για τελική, βοηθώντας να εξαχθούν συμπεράσματα σχετικά με την ανάπτυξη εννοιών ΥΣ (Troiano et al., 2019).

Γενικότερα, υπάρχει τάση αυτοματοποίησης των εργαλείων αξιολόγησης, ειδικά στα προγραμματιστικά περιβάλλοντα. Έτσι, δημιουργήθηκε το αυτοματοποιημένο σύστημα αξιολόγησης ΥΣ Ninja Code Village for Scratch (NCV) για το δημοτικό σχολείο (Ota, Morimoto & Kato, 2016), το οποίο προτείνει πλαίσιο αξιολόγησης με προγραμματιστικές έννοιες (Πίνακας 2), ενώ πρόσφατα αναπτύχθηκε το Scratch Analysis Tool (SAT) (Chang et al., 2018). Επιπλέον, το Functional Understanding Navigator! (FUN!) (Brasiel et al., 2017) αναλύει αυτόματα προγράμματα Scratch, δίνοντας μάλιστα ελεύθερα τον κώδικά του. Παράγει ως έξοδο φύλλα εργασίας κι έχει χρησιμοποιηθεί με μαθητές 10-13 ετών.

Το εργαλείο Commutative Assessment αξιολογεί μαθητές λυκείου σε έννοιες προγραμματισμού (Πίνακας 2) με συγκρίσεις μεταξύ μορφών κώδικα (blocks-κείμενο) (Weintrop & Wilensky, 2015). Οι Rowe & Cunningham (2017a· 2017b) προτείνουν ένα πλαίσιο αξιολόγησης μέσα από το παιχνίδι Zoombinis, αξιολογώντας έννοιες ΥΣ (Πίνακας 2), καθώς και στάδια από τα οποία περνούν οι παίκτες για την ολοκλήρωση του παιχνιδιού. Τα στάδια αναγνωρίζονται από βίντεο του παιχνιδιού και μελλοντικά σχεδιάζεται αυτοματοποίηση της διαδικασίας, ενώ χρησιμοποιείται ερευνητικό δείγμα όλων των ηλικιών και εμπειρίας.

Ο Basu (2019) προτείνει μια ρουμπρίκα αξιολόγησης στο πρόγραμμα CSForAll, η οποία αξιολογεί κυρίως προγραμματιστικές έννοιες σε block based περιβάλλοντα, όπως το Scratch και το App Inventor, και για την οποία έχουν γίνει κάποιοι πρώτοι έλεγχοι εγκυρότητας.

Στον τομέα του προγραμματισμού κινητών συσκευών προτείνεται για το App Inventor η ρουμπρίκα αξιολόγησης MCT (Mobile Computational Thinking), που χρησιμοποιεί 14 διαστάσεις, στις οποίες αναλύουν οι συγγραφείς την ΥΣ (Πίνακας 2) (Sherman & Martin, 2015). Το Quizly επίσης είναι ένα διαδικτυακό εργαλείο με ερωτήσεις κι αυτοματοποιημένη αξιολόγηση απαντήσεων για το App Inventor (Maiorana, Giordano & Morelli, 2015).

Νέες προσεγγίσεις προτείνουν την αξιολόγηση μέσω ανάλυσης δεδομένων χαμηλού επιπέδου από αρχεία καταγραφής της διαδικασίας προγραμματισμού σε Scratch (Srinivas et al., 2018), με ταυτόχρονη χρήση εργαλείου οπτικοποίησης των καταγεγραμμένων δεδομένων.

Πρόσφατα επίσης ανακοινώθηκε η πρόθεση χρήσης τεχνολογίας παρακολούθησης οφθαλμοκίνησης (eye-tracking) ως αντικειμενικό εργαλείο μέτρησης της κατανόησης εννοιών ΥΣ (Arslanyilmaz & Corpier, 2019), ειδικά για περιβάλλοντα όπως το Scratch κι έννοιες όπως η ακολουθία, επανάληψη, γεγονότα, τελεστές, κίνηση, κ.ά. Οι Papavlasopoulou, Sharma & Giannakos (2019) όμως χρησιμοποίησαν την ανίχνευση οφθαλμοκίνησης σε συνδυασμό με ποιοτικά δεδομένα, χωρίς να αποδείξουν από την οφθαλμοκίνηση τη διαφοροποίηση στις στρατηγικές και τις πρακτικές των μαθητών που κατέγραψαν ποιοτικά.

Κριτήρια αξιολόγησης ΥΣ και ψυχομετρικά εργαλεία

Εκτός από τα εργαλεία που στοχεύουν κυρίως σε προγραμματιστικά περιβάλλοντα, υπάρχουν αρκετά κριτήρια που προσπαθούν αυτόνομα να αξιολογήσουν την ΥΣ, σχετικά ψυχομετρικά εργαλεία, αλλά και μικτές, συνδυασμένες προσεγγίσεις. Σταδιακά εμφανίζονται

Πίνακας 3. Κριτήρια αξιολόγησης ΥΣ - Ψυχομετρικά εργαλεία

Μέθοδος	Έννοιες	Ηλικίες
Test for Measuring Basic Programming Abilities	Δομές ελέγχου (ακολουθία, επανάληψη, επιλογή)	Grades 7-10 (12-15 ετών)
Computational Thinking Test (CTt)	Βασικές ακολουθίες, βρόγχοι, επαναλήψεις, απλές και σύνθετες συνθήκες, συναρτήσεις και μεταβλητές	Grades 5-10 (10-15 ετών)
PCNT	Βρόχοι, αλγόριθμοι, πρότυπα(ακολουθία-δεδομένα), συνθήκες, βήματα(ακολουθία- τελεστές), οδηγίες, αυτόματα(ακολουθία-συνθήκες)	Μικρότερη των 10
Computational Thinking Scales (CTS)	Δημιουργικότητα, αλγοριθμική σκέψη, κριτική σκέψη, επίλυση προβλημάτων, συνεργατικότητα	Grades 5-12, Φοιτητές Τριτοβάθμιας
Computational Thinking Test	Χωρικός συλλογισμός, επαγωγή, μνήμη εργασίας	Δεν αναφέρει
Yagci CTS scale	Επίλυση προβλήματος, δημιουργική σκέψη, αλγοριθμική σκέψη και συνεργατική μάθηση μαζί με κριτική σκέψη	Λύκειο (15-17 ετών)

ψυχομετρικά εργαλεία για την ΥΣ, ενώ αρκετές προσεγγίσεις ανήκουν στο πεδίο της επιστήμης υπολογιστών, καθώς βασικές έννοιες προγραμματισμού εμπεριέχονται σε προσεγγίσεις της ΥΣ.

Στον Πίνακα 3 εμφανίζονται αντιπροσωπευτικά εργαλεία της κατηγορίας, με τις έννοιες που αξιολογούν και τις ηλικίες στις οποίες απευθύνονται.

Το ψυχομετρικό εργαλείο Test for Measuring Basic Programming Abilities (Mühling, Ruf & Hubwieser, 2015) βασίζεται σε υλικό του διαγωνισμού Bebras (Vilniaus Universitetas, n.d.) και χρησιμοποιείται τόσο για αρχική, όσο και για τελική αξιολόγηση, χρησιμοποιώντας έννοιες προγραμματισμού (Πίνακας 3).

Το Computational Thinking Test (CTt) (Román-González, Pérez-González & Jiménez-Fernández, 2017) αποτελείται από ερωτήσεις πολλαπλής επιλογής και καλύπτει μερικώς υπολογιστικές έννοιες της ΥΣ (Πίνακας 3), και καθόλου πρακτικές και οπτικές, όπως αναλύονται από τους Brennan & Resnick (2012). Χρησιμοποιείται για αρχική αξιολόγηση κι έχει συσχετιστεί με γνωστικές ικανότητες και την ικανότητα επίλυσης προβλημάτων. Προτείνεται να μην χρησιμοποιείται μόνο του, αλλά σε συνδυασμό με άλλα αντίστοιχα εργαλεία, όπως το Dr. Scratch. Για το CTt έχει γίνει έρευνα για συσχετίσεις με μεταβλητές της γνωστικής ψυχολογίας, αλλά και μη γνωστικούς παράγοντες, ενώ έχουν γίνει αρκετοί έλεγχοι εγκυρότητάς του για τις ηλικίες 10-15 ετών, κάτι που το καθιστά το μοναδικό ίσως έγκυρο εργαλείο αξιολόγησης της ΥΣ (Román-González et al., 2016· 2018a· 2018b· Tsarava et al., 2019). Συμπληρωματικά στο CTt, οι Pérez-Marín et al. (2018) δημιουργούν το PCNT για την αξιολόγηση της ΥΣ σε μικρότερους μαθητές και χρησιμοποιούν και τα δύο μαζί για αρχική και τελική αξιολόγηση σε διδακτικές παρεμβάσεις σε μαθητές 9-12 ετών.

Ο διαγωνισμός Bebras (Vilniaus Universitetas, n.d.) διοργανώνεται σε πάνω από 40 χώρες κι έχει ως αντικείμενο την πληροφορική και την ΥΣ. Τα θέματά του αντιστοιχίζονται σε έννοιες ΥΣ, όπως αφαίρεση, αποσύνθεση, αλγοριθμική σκέψη, αναγνώριση προτύπων ή γενίκευση, ενώ έχουν συγκεκριμένες προδιαγραφές, όπως ηλικία λύτη, απαιτούμενος χρόνος κ.ά. Θέματα του Bebras έχουν χρησιμοποιηθεί ήδη σε αξιολογήσεις (Dagiene & Stupuriene, 2016) κι αποτελούν πλέον αντικείμενο έρευνας στην αξιολόγηση της ΥΣ. Έχουν χρησιμοποιηθεί στο Test for Measuring Basic Programming Abilities όπως αναφέρθηκε, καθώς και σε

προτάσεις αρχικής και τελικής αξιολόγησης (Perez & Valladares, 2018), σε συνδυασμό και με άλλα υποστηρικτικά εργαλεία. Αναφέρεται όμως αδυναμία εξαγωγής μετρήσεων ΥΣ από τον Bebras, κι ότι ίσως όμως μελλοντικά μέσω έρευνας μετεξελιχθεί σε εργαλείο αξιολόγησης ΥΣ (Araujo et al., 2017). Οι Djambong et al. (2018) χρησιμοποιούν θέματα του Bebras και δικά τους, σε αξιολόγηση ΥΣ μαθητών 11-14 ετών, χωρίς σημαντικά αποτελέσματα, ενώ σε άλλη έρευνα για την αξιολόγηση ΥΣ μέσω θεμάτων του Bebras δεν επιβεβαιώνονται στατιστικά όλες οι έννοιες της ΥΣ του διαγωνισμού, παρά μόνο δύο παράγοντες, η ικανότητα αξιολόγησης (αφαίρεση, γενίκευση και αποσύνθεση) και μαζί η αλγοριθμική σκέψη με τη λογική αιτιολόγηση (Araujo et al., 2019). Σε άλλη παρόμοια έρευνα, επιβεβαιώνονται μόνο δύο παράγοντες, η ανάπτυξη της αλγοριθμικής σκέψης και η εύρεση προτύπων (Palts & Pedaste, 2017).

Επίσης, ερευνητές συνδυάζουν θέματα του Bebras με άλλες προσεγγίσεις, όπως π.χ. με το CTt (Wiebe et al., 2019) για αρχική αξιολόγηση ΥΣ σε μαθητές 11-13 ετών, αναφέροντας υψηλή συσχέτιση του CTt με το Dr. Scratch και του CTt με επιλεγμένα θέματα του Bebras. Τέλος, οι Rojas-Lopez & Garcia-Penalvo (2016) συνδυάζουν με τη σειρά τους θέματα από τον Bebras και την Ολυμπιάδα "Talent Search" για την αξιολόγηση νέων φοιτητών τριτοβάθμιας εκπαίδευσης, ενώ ερωτηματολόγιο βασισμένο επίσης σε θέματα της Ολυμπιάδας "Talent Search" για αρχική και τελική αξιολόγηση σε πρωτοετείς φοιτητές προτείνουν και οι Gouws, Bradshaw & Wentworth (2013).

Προχωρώντας, η κλίμακα Computational Thinking Scales (CTS) (Korkmaz, Cakir & Yasar Özden, 2017) χρησιμοποιεί ερωτήματα κλίμακας Likert, βασίζεται σε έννοιες ΥΣ των ISTE και CSTA (2015) (Πίνακας 3) κι έχει υποστεί ελέγχους εγκυρότητας και αξιοπιστίας. Απευθύνεται αρχικά σε φοιτητές, ενώ έχει προσαρμοστεί και για τη δευτεροβάθμια εκπαίδευση (Durak & Saritepeci, 2017).

Επιπλέον, το Computational Thinking Test (διαφορετικό από το ομώνυμο παραπάνω) είναι γνωστικό τεστ που ορίζεται μέσα στο πλαίσιο νοημοσύνης Cattell-Horn-Carroll (CHC), κι αξιολογεί γνωστικές παραμέτρους (Πίνακας 3) που συνδέονται με τον προγραμματισμό, καθώς και την οπτική αντίληψη και μνήμη (Ambrósio et al., 2014).

Οι Chen et al. (2017) σχεδιάζουν την αρχική και τελική αξιολόγηση ΥΣ μαθητών 10 ετών σε ένα πρόγραμμα ρομποτικής, χρησιμοποιώντας ερωτήματα προγραμματισμού σε μορφή κειμένου και μπλοκ, αλλά και καθημερινά σεμινάρια, θέλοντας ταυτόχρονα να ελέγξουν και την έννοια της μεταφοράς της γνώσης σε άλλους τομείς.

Το PCTT (Psychometric Computational Thinking Test) βρίσκεται σε αρχικό στάδιο, στοχεύοντας να γίνει σταθισμένο εργαλείο για τη μέτρηση της ΥΣ (Santisteban & Santisteban-Munoz, 2018). Επίσης, ο Yagci (2019) δημιούργησε μια κλίμακα μέτρησης της ΥΣ με ερωτήματα κλίμακας Likert για έννοιες ΥΣ (Πίνακας 3), η οποία στοχεύει σε μαθητές λυκείου, για την οποία ήδη έχει ξεκινήσει η διαδικασία ελέγχων εγκυρότητας.

Σε άλλη έρευνα ελέγχεται η αυτοεκτίμηση, στάσεις και κίνητρα για τον προγραμματισμό μικρών μαθητών, σε συνδυασμό με αποτελέσματα του CTt, χωρίς όμως να προκύπτουν σημαντικά αποτελέσματα (Leifheit et al., 2019).

Εκτός από την αξιολόγηση εννοιών ΥΣ προτείνεται κι η αξιολόγηση ανεξάρτητων από την ΥΣ δεξιοτήτων, όπως δεξιότητες για σχέσεις, γνωστικές δεξιότητες στη ζωή, ώστε να αξιολογηθεί πιο ολοκληρωμένα η ΥΣ (Basso et al., 2018). Τέλος, εμφανίζονται προβλεπτικοί παράγοντες για την ΥΣ, όπως ο τρόπος σκέψης, η ακαδημαϊκή επιτυχία στα μαθηματικά και η στάση απέναντι στα μαθήματα μαθηματικών (Durak & Saritepeci, 2017).

Πίνακας 4. Πολλαπλές μέθοδοι αξιολόγησης

Όνομα Μεθόδου	Είσοδος	Έννοιες	Ηλικίες
Πολλαπλές μορφές αξ/σης (Brennan & Resnick)	Scratch	Έννοιες (ακολουθία, επαναλήψεις, παραλληλισμός, γεγονότα, συνθήκες, τελεστές, δεδομένα), Πρακτικές (πειραματισμός και επανάληψη, έλεγχος και αποσφαλμάτωση, επαναχρησιμοποίηση και ανακάτεμα, αφαίρεση και αποσύνθεση), Οπτικές (έκφραση, σύνδεση, ερωτήματα)	Μη ρητή αναφορά – έρευνα σε ηλικίες 8-17 ετών
Foundations for Advancing Computational Thinking (FACT)		Αλγοριθμική σκέψη (αποσύνθεση, ακολουθίες, επαναλήψεις και συνθήκες), ορισμοί εννοιών, εκτέλεση κώδικα, αποσφαλμάτωση	Middle schools: Grades 6-8 (11-13 ετών)
Personalized Assessment Worksheets for Scratch (PAWS)		Ακολουθίες, γεγονότα, βρόχοι, συνθήκες	Έρευνα σε Grade 4 (9 ετών)
Βιντεοσκοπημένες συζητήσεις μαθητών (Portelance & Bers)	ScratchJr	Δεν αναφέρονται αναλυτικά	Έρευνα σε Grade 2 (7 ετών)
Πολλαπλές μορφές αξ/σης (Atmatzidou & Demetriadis)	Lego Mindstorms	Αφαίρεση, γενίκευση, αλγόριθμος, άρθρωμα, τμηματοποίηση, αποσφαλμάτωση, βασικές προγραμματιστικές έννοιες	Έρευνα σε 15-18 ετών
Mobile Computational Thinking (MCT)	App Inventor	Διαστάσεις ΥΣ (ονοματολογία, διαδικαστική αφαίρεση, μεταβλητές, βρόχοι, συνθήκες, λίστες) –8 επιπλέον διαστάσεις Κινητής ΥΣ (Mobile CT)	Δεν αναφέρουν
Quiz Maker – Quizly		Δομές του App Inventor	14-16 ετών, Φοιτητές (18-22 ετών)

Πολλαπλές μέθοδοι αξιολόγησης

Οι Brennan & Resnick (2012) προτείνουν για την αξιολόγηση της ΥΣ ποιοτική προσέγγιση με: α) ανάλυση φακέλου εργασιών, β) συνεντεύξεις σχετικά με τα έργα και γ) σχεδιαστικά σενάρια, συσχετίζοντας τις έννοιες, πρακτικές κι οπτικές με το προτεινόμενο πλαίσιο αξιολόγησης (Πίνακας 4). Τονίζεται η σπουδαιότητα του φακέλου εργασιών σε σχέση με την

αξιολόγηση μεμονωμένης εργασίας, που μπορεί να είναι παραπλανητική για τις γνώσεις ή τον κόπο των μαθητών. Το πλαίσιο αξιολόγησης χρησιμοποιείται ευρύτερα, όπως για παράδειγμα σε περιβάλλον Arduino και μαθητές 15-16 χρονών (Curasma et al., 2019), ενώ μέσα από νέες έρευνες στην υποχρεωτική εκπαίδευση, επεκτείνεται με την προσθήκη συμπληρωματικών δεξιοτήτων (Zhang & Nouri, 2019).

Ο Allsop (2018) προτείνει ένα μοντέλο αξιολόγησης που επίσης χρησιμοποιεί πολλαπλές προσεγγίσεις: α) υπολογιστικές έννοιες, β) μεταγνωστικές πρακτικές, γ) μαθησιακές συμπεριφορές, και τέλος, για την περίπτωση αξιολόγησης σε περιβάλλον κατασκευής παιχνιδιών, δ) σχεδιασμό παιχνιδιού, το οποίο εφαρμόζει σε μαθητές 11-12 ετών.

Πολλαπλοί μηχανισμοί αξιολόγησης προτείνονται επίσης στο Foundations for Advancing Computational Thinking (FACT) για μαθητές γυμνασίου, με δομημένα εργαλεία ενδιάμεσης και τελικής αξιολόγησης, σε συνδυασμό με εργασίες προγραμματισμού και ανοιχτά projects στο Scratch, καθώς και συνεντεύξεις μαθητών (Grover, 2017· Grover, Cooper & Pea, 2014). Γίνεται χρήση ρουμπρίκων κι όχι αυτόματη αξιολόγηση εργασιών, ενώ στην τελική αξιολόγηση χρησιμοποιούνται pre-test και post-test εργαλεία, αξιολογούνται βασικά στοιχεία αλγοριθμικής σκέψης (Πίνακας 4), καθώς και ορισμοί, κατανόηση δομής-λειτουργίας μικρών τμημάτων κώδικα Scratch, εκτέλεση κώδικα κι ικανότητες αποσφαλμάτωσης. Η όψη της μάθησης είναι πιο ολοκληρωμένη στα συστήματα αξιολογήσεων (Grover, Pea & Cooper, 2015).

Πολλαπλές μέθοδοι αξιολόγησης ΥΣ ακολουθούνται επίσης σε προγράμματα εκπαιδευτικής ρομποτικής. Μέσω ενός μοντέλου εννοιών ΥΣ (Πίνακας 4), γίνεται αρχική, ενδιάμεση και τελική αξιολόγηση με ποιοτικές και ποσοτικές μεθόδους, ρουμπρίκες, ερωτηματολόγια πολλαπλών επιλογών, ατομικές συνεντεύξεις και καταγραφή παρατηρήσεων εκπαιδευτικού (Atmatzidou & Demetriadis 2014· 2016). Αντίστοιχα, σε παρεμβάσεις εκπαιδευτικής ρομποτικής (Lego Mindstorms) κι εξελιγμένου προγραμματισμού (App Inventor για απομακρυσμένο χειρισμό, τρισδιάσταση σάρωση ανθρώπινων κινήσεων και μεταφορά τους ως εντολές στο ρομπότ) σε μαθητές 14-15 ετών, χρησιμοποιήθηκαν ποιοτικά και ποσοτικά δεδομένα για την αξιολόγηση της ΥΣ, αρχική και τελική αξιολόγηση, συνεντεύξεις, βιντεοσκοπήσεις, καθώς και ανάλυση των τελικών έργων (Merkouris & Chorianopoulos, 2019).

Οι πολλαπλές προσεγγίσεις βοηθούν στην εγκυρότητα των μετρήσεων και την τριγωνοποίησή τους, κι έτσι στο CEO (Code traces, Exit tickets, field Observations), για μαθητές γυμνασίου, χρησιμοποιούνται τρεις προσεγγίσεις για την επίτευξη τριγωνοποίησης (Lytle et al., 2019): α) καταγραφή κώδικα μαθητών σε διάφορα στάδια, β) ερωτηματολόγια αυτοαξιολόγησης των μαθητών και γ) παρατήρηση πεδίου, αναφέροντας θετικά αποτελέσματα στη μαθησιακή διαδικασία κι επιτυχημένες στρατηγικές ολοκλήρωσης των ασκήσεων. Αντίστοιχα, το PAWS (Personalized Assessment Worksheets for Scratch) (Salac, 2019) εφαρμόζεται στο δημοτικό και βασίζεται σε ανάλυση έργων, γραπτές αξιολογήσεις και συνεντεύξεις. Σε όλες τις φάσεις αξιολόγησης χρησιμοποιούνται τμήματα κώδικα από τα ίδια τα έργα των μαθητών, ώστε να ελεγχθεί το βάθος στο οποίο κατανοούν αυτά που χρησιμοποιούν, αλλά και να συνδεθούν όλες οι προσεγγίσεις μεταξύ τους.

Όπως προκύπτει, η ποιοτική έρευνα αναδεικνύεται ως σημαντικός παράγοντας στην αξιολόγηση. Πολλές έρευνες για την αξιολόγηση ΥΣ εφάρμοσαν στο παρελθόν μόνο παρατήρηση πεδίου, ενώ προτείνεται η χρήση του πρωτοκόλλου think-aloud και καταγραφή δραστηριοτήτων στον υπολογιστή, καθώς και η χρήση προκαθορισμένων κατηγοριών στην ανάλυση περιεχομένου (Lye & Koh, 2014). Εξελίσσοντας τη χρήση του πρωτοκόλλου think-aloud, ερευνητές προτείνουν συγκεκριμένες ερωτήσεις που στοχεύουν στη συνεχή και βαθύτερη κατανόηση της ικανότητας ΥΣ των μαθητών κατά τη διάρκεια της διδασκαλίας, καθώς ως διεργασία η ΥΣ συνεχώς εξελίσσεται και δεν μπορεί να αντιμετωπίζεται ως τελικό προϊόν (Mueller et al., 2017).

Σε μικρότερες ηλικίες, οι Portelance & Bers (2015) χρησιμοποιούν το ScratchJr και προτείνουν για την αξιολόγηση της ΥΣ βιντεοσκοπημένες συνεντεύξεις μαθητών ανά δύο μεταξύ τους, τις οποίες αναλύουν μετά οι ερευνητές κι εξάγουν συμπεράσματα για τις προσληφθείσες έννοιες ΥΣ. Σε ακόμα μικρότερες ηλικίες (3-5 ετών) προτείνεται το εργαλείο Coding Development (CODE) Test 3-6 για τη διερεύνηση της γνωσιακής πλευράς του προγραμματισμού (Marinus et al., 2018).

Εκτός από την ηλικία, ως παράγοντας διαφοροποίησης στην αξιολόγηση ΥΣ, εμφανίζεται η διδακτική προσέγγιση. Έτσι, για την αξιολόγηση ΥΣ μέσω STEM προτείνονται ασκήσεις με Motion Charts κι ένα Μοντέλο Κλιματικής Αλλαγής σε NetLogo (Weintrop et al., 2014). Αντίστοιχα, οι Swanson et al. (2019) αναφέρονται σε πρακτικές ΥΣ στο STEM (CT-STEM) χρησιμοποιώντας ποιοτικές αναλύσεις γραπτών εργασιών, καθώς κι ανάλυση αρχείων καταγραφής της NetLogo και της δουλειάς των παιδιών. Υπάρχουν γενικότερα αναφορές για την ανάπτυξη της ΥΣ μέσω STEM (Basu, Kinnebrew & Biswas, 2014· Zhang & Biswas, 2019). Οι τελευταίοι προτείνουν συγκεκριμένο πλαίσιο για την ΥΣ μέσω STEM για μαθητές γυμνασίου, καθώς και μικτή μεθοδολογία τελικής και διαμορφωτικής αξιολόγησης, που βασίζεται στις λύσεις των μαθητών και τα μοντέλα που δημιουργούν, αλλά και στη δραστηριότητά τους μέσα στο περιβάλλον του CTSiM που χρησιμοποιείται.

Οι Snow et al. (2017) αναφέρουν τη διαδικασία σχεδιασμού της αξιολόγησης ΥΣ που ακολουθούν στο πρόγραμμα λυκείου Exploring Computer Science, διαχωρίζοντας την αξιολόγηση ΥΣ από την απλή αξιολόγηση γνώσεων Πληροφορικής. Οι Park et al. (2016) χρησιμοποιούν τρία εργαλεία για να αξιολογήσουν την ικανότητα φοιτητών στην ιεραρχική δομή σε κώδικα, την οποία θεωρούν ως βασική δεξιότητα ΥΣ, εντασσόμενη στην αφαίρεση, ενώ οι Rich, Egan & Ellsworth (2019) μελετούν την αποσύνθεση, προτείνοντας ένα πλαίσιο που θεωρεί την αποσύνθεση ως μια διαδικασία που περιλαμβάνει κατηγορίες, στρατηγικές ανάπτυξης και συνεχή αξιολόγηση της πορείας που ακολουθείται, διευκρινίζοντας ότι η αποσύνθεση μπορεί να συμβεί πριν ή και μετά την εφαρμογή άλλων εννοιών ΥΣ, όπως η αφαίρεση ή η αναγνώριση προτύπων.

Παρά τις πολλές προσεγγίσεις που υπάρχουν για την ΥΣ σε προγραμματιστικά περιβάλλοντα, η ΥΣ ως γνωσιακή διεργασία δεν συνδέεται απαραίτητα με τη χρήση υπολογιστή. Το Computer Science Unplugged (University of Canterbury, n.d.) στοχεύει στη διδασκαλία εννοιών της επιστήμης υπολογιστών και στην ανάπτυξη της ΥΣ, χωρίς χρήση υπολογιστή, ενώ υπάρχουν σχέδια μαθημάτων για εφαρμογή στην τάξη (CS Unplugged at Mines, n.d.), στα οποία χρησιμοποιούνται φύλλα εργασίας για την αξιολόγησή της (Rodriguez, Rader & Camp, 2016). Επιπλέον, στην αξιολόγηση ΥΣ μέσω του CS Unplugged χρησιμοποιούνται ρουμπρίκες αξιολόγησης (αλγοριθμική σκέψη, αφαίρεση κι αναπαράσταση δεδομένων) και φύλλα εργασίας για την αναγνώριση προτύπων (Rodriguez et al., 2017).

Τέλος, στο Principled Assessment of Computational Thinking (PACT) προτείνονται πρότυπα σχεδιασμού πρακτικών για την ανάπτυξη της ΥΣ και δύο υποστηρικτικά πρότυπα για την υλοποίηση αξιολογήσεων στη δευτεροβάθμια εκπαίδευση, ενώ προτείνονται παραδείγματα με πραγματικές προσεγγίσεις για την οικοδόμηση ΥΣ μέσω STEM, Scratch, παιχνιδιών, προσομοιώσεων κι αφήγηση ιστοριών σε AgentSheets, Alice (Bienkowski et al., 2015).

Συμπεράσματα

Συνοψίζοντας τα συμπεράσματα της παρούσας έρευνας, παρατηρούμε ότι τα υπάρχοντα αυτοματοποιημένα εργαλεία δεν μπορούν ακόμα να αξιολογήσουν επαρκώς κι αυτόνομα την ΥΣ. Οι πολλαπλές μέθοδοι αξιολόγησης, όπως περιεγράφηκαν προηγουμένως, δείχνουν να είναι ο πλέον ενδεδειγμένος τρόπος αξιολόγησης της ΥΣ και συστήνονται ακόμα κι από ερευνητές που υλοποιούν αυτοματοποιημένα κριτήρια αξιολόγησης (Moreno-León & Robles, 2015a), ενώ άλλες έρευνες έχουν αναφέρει τη μη σύγκλιση των αυτοματοποιημένων αποτελεσμάτων με αυτά των ποιοτικών προσεγγίσεων (Hoover et al., 2016). Για τις πολλαπλές μεθόδους αξιολόγησης η πιο ασφαλής προσέγγιση δείχνει να είναι η χρήση ποιοτικών μεθόδων αξιολόγησης της ΥΣ, συνήθως με προσωπικές συνεντεύξεις αλλά και τη

χρήση του πρωτοκόλλου think-aloud, και με ταυτόχρονη χρήση παρατήρησης πεδίου, ρουμπρικών αξιολόγησης και φακέλου εργασιών μαθητών (Brennan & Resnick, 2012· Grover, 2017· Grover, Cooper & Pea, 2014). Αναφέρθηκαν ήδη αρκετές παρόμοιες προσεγγίσεις και προτάσεις βελτιστοποίησης, όπως είναι η χρήση προκαθορισμένων κατηγοριών στην ανάλυση περιεχομένου, αλλά και προκαθορισμένων ερωτήσεων στο πρωτόκολλο think-aloud (Lye & Koh, 2014· Mueller et al., 2017). Τα ερευνητικά δεδομένα δείχνουν να συγκλίνουν σε μια τέτοια προσέγγιση πολλαπλής αξιολόγησης, η οποία θα εμπεριέχει σαφώς διαμορφωτική αξιολόγηση κατά τη διάρκεια των παρεμβάσεων, αλλά και τελική αξιολόγηση των έργων και του φακέλου εργασιών των μαθητών.

Έλλειψη κοινού τόπου αξιολόγησης της ΥΣ

Η έλλειψη κοινά αποδεκτού ορισμού της ΥΣ δεν βοηθά στις προσπάθειες αξιολόγησής της. Η διαφορετική αφετηρία (ορισμός ΥΣ) οδηγεί πολλές φορές σε αξιολόγηση διαφορετικών εννοιών, οι οποίες μπορεί να αποτελούν ένα μόνο τμήμα της ΥΣ του αξιολογούμενου. Το σημείο που φαίνεται να συνδέει τις διαφορετικές προτάσεις, είναι η αλγοριθμική σκέψη και οι βασικές προγραμματιστικές δομές (ακολουθία, επανάληψη και συνθήκες), ενώ ακολουθεί η αφαιρετική ικανότητα και τέλος η αποσύνθεση. Το συμπέρασμα αυτό συγκλίνει με αντίστοιχο των Araujo, Andrade & Guerrero (2016), ότι οι περισσότερες προτάσεις αξιολόγησης ΥΣ αφορούν στον προγραμματισμό, με κύριες έννοιες την επίλυση προβλημάτων, την αφαίρεση και τους αλγόριθμους, αλλά και των Cutumisu et al. (2019), οι οποίοι αναφέρουν ως συχνότερες έννοιες αξιολόγησης την αλγοριθμική σκέψη, την αφαίρεση, την αποσύνθεση, τη λογική σκέψη και τα δεδομένα. Σημειώνεται ότι δεν έχει ερευνηθεί σχεδόν καθόλου ο συσχετισμός μεταξύ των εννοιών αυτών, όσον αφορά την κατάκτησή τους από μεριάς μαθητών. Ενδιαφέρον ερώτημα αποτελεί το αν θα μπορούσε να υπάρχει οικοδόμηση κάποιας βασικής συνιστώσας της ΥΣ χωρίς την ανάγκη ταυτόχρονης/παράλληλης οικοδόμησης κάποιας άλλης ή αν κάποιες έννοιες οικοδομούνται πάντα μαζί. Αυτά τα ερωτήματα, με εξαίρεση ελάχιστες έρευνες όπως αυτή των Rich et al. (2019), δεν απαντώνται σε σχετικές έρευνες. Κάποιες προσεγγίσεις αξιολογούν την ΥΣ ως μια γενικότερη έννοια, αδυνατώντας όμως να δώσουν επιμέρους αποτελέσματα για κάθε μια από τις κύριες συνιστώσες της.

Ως απάντηση στην έλλειψη κοινού τόπου στις προσπάθειες αξιολόγησης της ΥΣ έρχεται η πρόταση που εκλαμβάνει ως ουσιαστικό κριτήριο αξιολόγησης της ΥΣ την δυνατότητα μεταφοράς γνώσης, δηλαδή το να μπορούν οι μαθητές να εφαρμόσουν τη γνώση τους και σε άλλους τομείς (Chen et al., 2017· Koh et al., 2010). Αυτόνομες προσπάθειες αξιολόγησης όπως ο διαγωνισμός Bebras, κατατάσσονται από τους Román-González et al. (2019) στην κατηγορία μεταφοράς ικανοτήτων, καθώς οι αξιολογούμενοι καλούνται να εφαρμόσουν την αποκτηθείσα γνώση σε άλλους τομείς, όπως προβλήματα της καθημερινής ζωής. Έχοντας αυτή την παράμετρο υπόψη, μπορεί κι η αξιολόγηση να γίνει ανεξάρτητη από συγκεκριμένες πλατφόρμες κι έτσι οι Chen et al. (2017) μιλούν για την ανάγκη για εργαλεία αξιολόγησης που είναι διαλειτουργικά (cross-platform), ώστε να μπορούν να ελέγχουν τη διάσταση αυτή.

Υπό προϋποθέσεις και υπό διαμόρφωση μέθοδοι αξιολόγησης ΥΣ

Ερευνητές αναφέρουν για τα σενάρια σε προγραμματιστικά περιβάλλοντα ότι προϋποθέτουν εξοικείωση, κι ότι δεν μπορούν να χρησιμοποιηθούν από την έναρξη, αλλά ούτε κι ενδιάμεσα αν το επίπεδο χειρισμού δεν είναι ικανοποιητικό (Webb, 2010· Werner et al., 2012). Η αυτοματοποιημένη διαδικασία αξιολόγησης επίσης δεν είναι ώριμη, κι έτσι μαζί με το, αυτοματοποιημένο ή μη, σύστημα προτείνεται να ακολουθείται ταυτόχρονα κι άλλη μορφή αξιολόγησης (Moreno-León et al., 2017· Román-González et al., 2017). Σύγκριση τριών δημοφιλών εργαλείων αξιολόγησης (CTt, Dr. Scratch, επιλεγμένα θέματα Bebras)

καταλήγει σε μερικώς συγκλίνουσα εγκυρότητα, δηλαδή δεν θα μπορούσε κάποιο εργαλείο να αντικαταστήσει κάποιο άλλο, και προτείνεται η χρήση όλων για πληρέστερη εικόνα (Román-González et al., 2019). Οι Mueller et al. (2017) αναφέρουν επίσης ότι δεν θα έπρεπε να αξιολογείται και να αντιμετωπίζεται η ΥΣ ως τελικό στάδιο, αφού είναι διεργασία η οποία έχει στάδια και συνεχώς εξελίσσεται, κι επιχειρηματολογούν υπέρ της (συνεχούς) ενδιάμεσης αξιολόγησης. Οι Hadad et al. (2019) προτείνουν μεθοδολογία και μελετούν ενδιάμεσα στάδια άτυπης διαμορφωτικής αξιολόγησης, που δείχνουν ελπιδοφόρα για την ανάπτυξη δεξιοτήτων και στάσεων ΥΣ στη μαθησιακή διαδικασία.

Τέλος, οι Giordano et al. (2015) αναφέρονται ειδικότερα στις παραμέτρους που πρέπει να λαμβάνονται υπόψη στο σχεδιασμό αξιολογήσεων στον προγραμματισμό, όπως είναι η αποφυγή ελέγχου σύνταξης, η απαγκίστρωση από συγκεκριμένη γλώσσα προγραμματισμού κι η προτίμηση ψευδοκώδικα, η χρήση ρουμπρίκων αξιολόγησης, η χρήση παιχνιδοποίησης και διαγωνισμών για να είναι συναρπαστικότερη η αξιολόγηση, αλλά κι επικύρωση και διαμοιρασμός των εργαλείων αξιολόγησης μεταξύ ιδρυμάτων για ποιοτικότερη εξέλιξη.

Ζητήματα αξιοπιστίας μεθοδολογιών και γενίκευσης αποτελεσμάτων αξιολόγησης ΥΣ

Ένα σημαντικό ζήτημα που απασχολεί είναι η δυνατότητα γενίκευσης των προτεινόμενων μεθόδων αξιολόγησης. Οι περισσότερες έρευνες υλοποιούνται με μικρά, βολικά τις περισσότερες φορές, δείγματα, με αποτέλεσμα να μην είναι εφικτή η γενίκευση των αποτελεσμάτων. Επίσης, ελάχιστες χρησιμοποιούμενες μεθοδολογίες και εργαλεία έχουν μέχρι στιγμής υποβληθεί σε ελέγχους εγκυρότητας και αξιοπιστίας, ενώ όπου υπάρχουν αναφορές ελέγχων, αφορούν σε περιορισμένες χώρες και πληθυσμούς (π.χ. Dr. Scratch στην Ισπανία, CTS στην Τουρκία), αφήνοντας ανοικτά ερωτήματα για την ισχύ τους σε άλλες χώρες, άλλα εκπαιδευτικά συστήματα. Αντίστοιχα, οι Petri & Gresse von Wangenheim (2017), αναφερόμενοι στην αξιολόγηση της υπολογιστικής εκπαίδευσης (computing education) μέσω παιχνιδιών, καταλήγουν ότι στις περισσότερες έρευνες η αξιολόγηση δεν είναι καλά καθορισμένη, ενώ τα δείγματα είναι μικρά, θέτοντας επίσης θέματα αδυναμίας γενίκευσης κι εκφράζοντας την άποψη ότι χρειάζεται ενίσχυση της επιστημονικής αυστηρότητας για τις αξιολογήσεις αυτές. Επιπλέον, οι Kalelioglu, Gulbahar & Kukul (2016) αναφέρουν ότι γενικά οι έρευνες για την ΥΣ πάσχουν από ανάλυση σε βάθος, και δεν συζητείται η επιστημονική αξία τους.

Ηλικιακές ομάδες εφαρμογής αξιολόγησης της ΥΣ

Από τις έρευνες αξιολόγησης της ΥΣ που μελετήθηκαν αναδεικνύεται ότι η πλειοψηφία των ερευνών αυτών αφορά σε μαθητές των μεγαλύτερων τάξεων του δημοτικού σχολείου, καθώς και μαθητών τάξεων του γυμνασίου, ειδικά στις έρευνες για αξιολόγηση εννοιών προγραμματισμού. Ελάχιστες έρευνες αφορούσαν στην προσχολική ηλικία, ενώ κάποιες λίγες αφορούσαν ενήλικες, κυρίως προπτυχιακούς φοιτητές. Με εξαίρεση ελάχιστα εργαλεία που έχουν ελεγχθεί για τη εγκυρότητά τους, οι περισσότερες προτάσεις εφαρμόζονται σε συγκεκριμένες τάξεις (βολικές δειγματοληψίες), κι έτσι, ακόμα κι αν ένα εργαλείο υποθετικά καλύπτει μια διάσταση της ΥΣ, δεν μπορεί να εφαρμοστεί γενικά, χωρίς να ληφθεί υπόψη η ηλικία για την οποία έχει σχεδιαστεί ή εφαρμοστεί ερευνητικά. Χρειάζεται να γίνει αρκετή έρευνα ακόμα, με επιστημονική τεκμηρίωση για τη θεμελίωση των ερευνητικών αποτελεσμάτων στις ηλικίες που αναφέρονται.

Μελλοντικές κατευθύνσεις

Τα τελευταία χρόνια γίνεται αρκετή έρευνα για την αξιολόγηση της ΥΣ, αλλά χρειάζεται να γίνει σε μεγαλύτερη κλίμακα, με επιστημονικά τεκμηριωμένες προσεγγίσεις, για όλες τις ηλικίες. Κλείνοντας, μια ολοκληρωμένη προσέγγιση αξιολόγησης ΥΣ θα πρέπει να λάβει

υπόψη όλες τις παραπάνω παραμέτρους, να σχεδιαστεί εξ αρχής για συγκεκριμένη ηλικιακή ομάδα, να εφαρμοστεί σε ικανό αριθμό συμμετεχόντων και να διασφαλίσει την εγκυρότητα και αξιοπιστία της με επιστημονικές μεθόδους, ενώ οι πολλαπλές μέθοδοι αξιολόγησης με χρήση ποιοτικών προσεγγίσεων, παρατήρησης πεδίου, αλλά και φακέλου εργασιών μαθητή με αρχική, ενδιάμεση και τελική αξιολόγηση, κρίνονται επιβεβλημένες.

Αναφορές

Aho, A. V. (2012). Computation and computational thinking. *Computer Journal*, 55, 832–835.

Allsop, Y. (2018). Assessing computational thinking process using a multiple evaluation approach. *International Journal of Child-Computer Interaction*, 19, 30–55.

Ambrósio, A.P., Georges, F., & Xavier, C. (2014). Digital ink for cognitive assessment of computational thinking. In M. Castro & E. Tovar (eds.), 2014 *IEEE Frontiers in Education Conference* (pp. 1520-1526). New Jersey: IEEE.

Araujo, A.L.S.O., Andrade, W.L., & Guerrero, D.D.S. (2016). A systematic mapping study on assessing computational thinking abilities. In S. Frezza, D. Onipede, K. Vernaza & M. Ford (eds.), *2016 IEEE Frontiers in Education Conference* (pp. 1-9). New Jersey: IEEE.

Araujo, A.L.S.O., Andrade, W.L., Guerrero, D.D.S., & Melo, M.R.A. (2019). How many abilities can we measure in Computational Thinking?: A study on Bebras Challenge. In E.K. Hawthorne, M.A. Pérez-Quiñones, S. Heckman & J. Zhang (eds.), *Proceedings of the 50th ACM Technical Symposium on Computer Science Education* (pp. 545-551). New York: ACM.

Araujo, A.L.S.O., Santos, J.S., Andrade, W.L., Guerrero, D.D.S., & Dagienė, V. (2017). Exploring computational thinking assessment in introductory programming courses. In W. Oakes (ed.), *2016 IEEE Frontiers in Education Conference* (pp. 1-9). New Jersey: IEEE.

Arslanyilmaz, A., & Corpier, K. (2019). Eye tracking to evaluate comprehension of computational thinking (poster). In B. Scharlau, R. McDermott, A. Pears & M. Sabin (eds.), *Proceedings of the 2019 ACM Conference on Innovation and Technology in Computer Science Education* (pp. 296). New York: ACM.

Atmatzidou, S., & Demetriadis, S. (2014). How to support students' computational thinking skills in educational robotics activities. In D. Alimisis, G. Granosik & M. Moro (eds.), *Proceedings of 4th International Workshop "Teaching Robotics & Teaching with Robotics" & 5th International Conference "Robotics in Education"* (pp. 43-50). Padova: TRTWR & RiE.

Atmatzidou, S., & Demetriadis, S. (2016). Advancing students' computational thinking skills through educational robotics: A study on age and gender relevant differences. *Robotics and Autonomous Systems*, 75(B), 661-670.

Barr, V., & Stephenson, C. (2011). Bringing computational thinking to K-12: What is involved and what is the role of the computer science education community. *ACM InRoads*, 2(1), 48-54.

Basawapatna, A., Koh, K.H., & Repenning, A. (2010). Using Scalable Game Design to teach computer science from middle school to graduate students. In R. Ayfer. J. Impagliazzo & C. Laxer (eds.), *ITiCSE '10: Proceedings of the fifteenth annual conference on Innovation and technology in computer science education* (pp. 224-228). New York: ACM.

Basawapatna, A., Repenning, A., & Koh, K.H. (2015). Closing the cyberlearning loop: Enabling teachers to formatively assess student programming projects. *Proceedings of the 46th ACM Technical Symposium on Computer Science Education SIGCSE '15* (pp. 12-17). Kansas City: ACM.

Basso, D., Fronza, I., Colombi, A., & Pahl, C. (2018). Improving assessment of computational thinking through a comprehensive framework. In M. Joy & P. Ihantola (eds.), *Proceedings of the 18th Koli Calling International Conference on Computing Education Research* (pp. 1-5). New York: ACM.

Basu, S. (2019). Using Rubrics Integrating Design and Coding to Assess Middle School Students' Open-ended Block-based Programming Projects. In E.K. Hawthorne, M.A. Pérez-Quiñones, S. Heckman & J. Zhang (eds.), *Proceedings of the 50th ACM Technical Symposium on Computer Science Education* (pp. 1211-1217). New York: ACM.

Basu, S., Kinnebrew, J.S., & Biswas, G. (2014). Assessing student performance in a computational-thinking based science learning environment. In S. Trausan-Matu, K.E. Boyer, M. Crosby & K. Panourgia (eds.), *Intelligent Tutoring Systems* (pp. 476-481). Cham: Springer.

Bell, T., & Lodi, M. (2019). Constructing computational thinking without using computers. *Constructivist Foundations*, 14(3), 342-351.

Bienkowski, M., Snow, E., Rutstein, D. W., & Grover, S. (2015). *Assessment design patterns for computational thinking practices in secondary computer science: A first look (SRI technical report)*. Menlo Park: SRI International.

Bocconi, S., Chioccariello, A., Dettori, G., Ferrari, A., & Engelhardt, K. (2016). *Developing computational thinking in compulsory education – Implications for policy and practice*. Retrieved 9 September 2018, from https://publications.jrc.ec.europa.eu/repository/bitstream/JRC104188/jrc104188_computhinkreport.pdf.

Brasiel, S., Close, K., Jeong, S., Lawanto, K., Janisiewicz, P., & Martin, T. (2017). Measuring Computational Thinking Development with the FUN! Tool. In P.J. Rich & C.B. Hodges (eds.), *Emerging Research, Practice, and Policy on Computational Thinking, Educational Communications and Technology: Issues and Innovations* (pp. 327-347). Cham: Springer.

Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. In A.F. Ball & C.A. Tyson (eds.), *Proceedings of the 2012 Annual Meeting of the American Educational Research Association* (pp. 1-25). Washington: AERA.

Bryman, A. (2008). *Social Research Methods* (3rd edition). New York: Oxford University Press.

Carnegie Mellon (n.d.). *Center for Computational Thinking*. Retrieved 9 September 2018, from https://www.cs.cmu.edu/~CompThink.

Chang, Z., Sun, Y., Wu, T., & Guizani, M. (2018). Scratch Analysis Tool (SAT): A modern scratch project analysis tool based on ANTLR to assess computational thinking skills. In M. Gerla, G. Hadjichristofi, C. Chrysostomou & M. Guizani (eds.), *Proceedings of the 14th International Wireless Communications & Mobile Computing Conference* (pp. 950-955). New Jersey: IEEE.

Chen G., Shen, J., Barth-Cohen, L., Jiang, S., Huang, X., & Eltoukhy, M. (2017). Assessing elementary students' computational thinking in everyday reasoning and robotics programming. *Computers & Education*, 109, 162-175.

Computational Thinking with Scratch (n.d.). *Computational thinking*. Retrieved 9 September 2018, from http://scratched.gse.harvard.edu/ct/defining.html.

Computing at School (n.d.). Retrieved 11 November 2019, from https://www.computingatschool.org.uk.

Computing at School (2015). *Computational Thinking. A guide for Teachers*. Retrieved 11 November 2019, from https://community.computingatschool.org.uk/files/8550/original.pdf.

CS Unplugged at Mines (n.d.). *CS Unplugged*. Retrieved 9 September 2018, from http://csunplugged.mines.edu/index.html.

Curasma, R.P., Jara, N.J., Curasma, H.P., & Ornetta, V.C. (2019). Assessment of Computational Thinking in regular basic education: case I.E.T.P. "José Obrero". In C. Gallegos & C. Silva (eds.), *INTERCON 2019: IEEE XXVI International Conference on Electronics, Electrical Engineering and Computing* (pp. 1-4). New Jersey: IEEE.

Cutumisu, M., Adams, C., & Lu, C. (2019). A Scoping Review of Empirical Research on Recent Computational Thinking Assessments. *Journal of Science Education and Technology*, 28, 651–676.

Dagienė, V., & Sentance, S. (2016). It's Computational Thinking! Bebras tasks in the curriculum. In A. Brodnik & F. Tort (eds.), *Informatics in Schools: Improvement of Informatics Knowledge and Perception* (pp. 28-39). Cham: Springer.

Dagiene, V., & Stupuriene, G. (2016). Bebras – A sustainable community building model for the concept based learning of informatics and computational thinking. *Informatics in Education*, 15(1), 25-44.

Djambong, T., Freiman, V, Gauvin, S., Paquet, M., & Chiasson, M. (2018). Measurement of Computational Thinking in K-12 education: The need for innovative practices. In D. Sampson, D. Ifenthaler, J. Spector & P. Isaías (eds.), *Digital Technologies: Sustainable Innovations for Improving Teaching and Learning* (pp. 193-222). Cham: Springer.

Durak, H.Y., & Saritepeci, M. (2017). Analysis of the relation between computational thinking skills and various variables with ste structural equation model. *Computers & Education*, 116, 191-202.

European Commission (2016). *The Computational Thinking Study*. Retrieved 9 September 2018, from https://ec.europa.eu/jrc/en/computational-thinking.

Fessakis, G., Komis, V., Mavroudi, E., & Prantsoudi, S. (2018). Exploring the scope and the conceptualization of Computational Thinking at the K-12 classroom level curriculum. In M.S. Khine (ed.), *Computational Thinking in the STEM Disciplines: Foundations and Research Highlights* (pp. 181-212). Cham: Springer.

Fessakis, G., Mavroudi, E., & Prantsoudi, S. (2017). Current Conception of Computational Thinking at the K-12 classroom level curriculum (poster). In V. Dagienė & A. Hellas (eds.), *Informatics in Schools: Focus on Learning Programming* (pp. 1-6). Cham: Springer.

Giordano, D., Maiorana, F., Csizmadia, A., Marsden, S., Riedesel, C., Mishra, S., & Vinikiene, L. (2015). New horizons in the assessment of computer science at school and beyond: Leveraging on the ViVA platform. In N. Ragonis & P. Kinnunen (eds.), *Proceedings of the 2015 ITiCSE on Working Group Reports* (pp. 117-147). New York: ACM.

Gouws, L., Bradshaw, K., & Wentworth, P. (2013). First year student performance in a test for Computational Thinking. In J. McNeill, K. Bradshaw, P. Machanick & M. Tsietsi (eds.), *Proceedings of the South African Institute for Computer Scientists and Information Technologists Conference* (pp. 271-277). New York: ACM.

Grover, S. (2017). Assessing algorithmic and computational thinking in K-12: Lessons from a middle school classroom. In P.J. Rich & C.B. Hodges (eds.), *Emerging Research, Practice, and Emerging Research, Practice, and Policy on Computational Thinking. Educational Communications and Technology: Issues and Innovations* (pp. 269-288). Cham: Springer.

Grover, S., & Basu, S. (2017). Measuring Student Learning in Introductory Block-Based Programming: Examining Misconceptions of Loops, Variables, and Boolean Logic. In M.E. Caspersen, S.H. Edwards, T. Barnes & D.D.

Garcia (eds.), *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education* (pp. 267-272). New York: ACM.

Grover, S., Basu, S., Bienkowski, M., Eagle, M., Diana, N., & Stamper, J. (2017). A framework for using hypothesis-driven approaches to support data-driven learning analytics in measuring computational thinking in block-based programming environments. *ACM Transactions on Computing Education*, 17(3), 1-25.

Grover, S., Cooper, S., & Pea, R. (2014). Assessing computational learning in K-12. In Å. Cajander, M. Daniels, T. Clear & A.N. Pears (eds.), *Proceedings of the 2014 Conference on Innovation & technology in computer science education* (pp. 57-62). New York: ACM.

Grover, S., & Pea, R. (2013). Computational Thinking in K–12: A review of the state of the field. *Educational Researcher*, 42(1), 38–43.

Grover, S., Pea, R., & Cooper, S. (2015). "Systems of Assessments" for deeper learning of Computational Thinking in K-12. In J.E. King & B.M. Gordon (eds.), *Proceedings of the 2015 Annual Meeting of the American Educational Research Association* (pp. 1-10). Washington: AERA.

Hadad, R., Thomas, K., Kachovska, M., & Yin, Y. (2019). Practicing formative assessment for computational thinking in making environments. *Journal of Science Education and Technology*, 29, 162–173.

Heintz, F., Mannila, L., & Färnqvist, T. (2016). A review of models for introducing computational thinking, computer science and computing in K-12 education. In S. Frezza, D. Onipede, K. Vernaza & M. Ford (eds.), 2016 *IEEE Frontiers in Education Conference* (pp. 1-9). New Jersey: IEEE.

Hoover, A.K., Barnes, J., Fatehi, B., Moreno-León, J., Puttick, G., Tucker-Raymond, E., & Harteveld, C. (2016). Assessing Computational Thinking in students' game designs. In J. Kaye et al. (eds.), *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems* (pp. 173-179). New York: ACM.

ISTE, & CSTA (2011). *Operational definition of Computational Thinking in K-12 education*. Retrieved 9 September 2018, from https://id.iste.org/docs/ct-documents/computational-thinking-operational-definition-flyer.pdf?sfvrsn=2.

ISTE, & CSTA (2015). *Computational thinking leadership toolkit*. Retrieved 9 September 2018, from https://id.iste.org/docs/ct-documents/ct-leadershipt-toolkit.pdf?sfvrsn=4.

Kalelioglu, F., Gulbahar, Y. & Kukul, V. (2016). A framework for computational thinking based on a systematic research review. *Baltic J. Modern Computing*, 4(3), 583-596.

Koh, K.H., Basawapatna, A., Bennett, V., & Repenning, A. (2010). Towards the automatic recognition of computational thinking for adaptive visual language learning. In C. Hundhausen, E. Pietriga, P. Díaz & M.B. Rosson (eds.), *Proceedings of the 2010 IEEE Symposium on Visual Languages and Human-Centric Computing* (pp. 59-66). New Jersey: IEEE.

Koh, K.H., Basawapatna, A., Nickerson, H., & Repenning, A. (2014). Real Time assessment of computational thinking. In S. Fleming, A. Fish & C. Scaffidi (eds.), *Proceedings of the 2014 IEEE Symposium on Visual Languages and Human-Centric Computing* (pp. 49-52). New Jersey: IEEE.

Koh, K.H., Nickerson, H., Basawapatna, A., & Repenning A. (2014). Early validation of Computational thinking pattern analysis. In Å. Cajander, M. Daniels, T. Clear & A.N. Pears (eds.), *Proceedings of the 2014 conference on Innovation & technology in computer science education* (pp. 213-218). New York: ACM.

Korkmaz, Ö., Cakir, R., & Yasar Özden, M. (2017). A validity and reliability study of the computational thinking scales (CTS). *Computers in Human Behavior*, 72, 558-569.

Leifheit, L., Tsarava, K., Moeller, K., Ostermann, K., Golle, J., Trautwein, U., & Ninaus, M. (2019). Development of a Questionnaire on Self-concept, Motivational Beliefs, and Attitude Towards Programming. Q. Cutts & T. Brinda (eds.), *Proceedings of the 14th Workshop in Primary and Secondary Computing Education* (pp. 1-9). New York: ACM.

Lye, S.Y., & Koh, J.H.L. (2014). Review on teaching and learning of computational thinking through programming: What is next for K-12?. *Computers in Human Behavior*, 41, 51-61.

Lytle, N., Catete, V., Dong, Y., Boulden, D., Akram, B., Houchins, J., Barnes, T., & Wiebe, E. (2019). CEO: A triangulated evaluation of a modeling-based CT-infused CS activity for non-CS middle grade students. In M. Zhang, B. Yang, S. Cooper & A. Luxton-Reilly (eds.), *Proceedings of the ACM Conference on Global Computing Education* (pp. 58-64). New York: ACM.

Maiorana, F., Giordano, D., & Morelli, R. (2015). Quizly: A live coding assessment platform for App Inventor. In E. Kraemer, C. Ermel & S. Fleming (eds.), *Proceedings of the 2015 IEEE Symposium on Visual Languages and Human-Centric Computing (Blocks and Beyond workshop)* (pp. 25-30). New Jersey: IEEE.

Marinus, E., Powell, Z., Thornton, R., McArthur, G., & Crain, S. (2018). Unravelling the cognition of coding in 3-to-6-year olds: The development of an assessment tool and the relation between coding ability and cognitive compiling of syntax in natural language. In L. Malmi et al. (eds.), *Proceedings of the 2018 ACM Conference on International Computing Education Research* (pp. 133-141). New York: ACM.

Merkouris, A., & Chorianopoulos, K. (2019). Programming embodied interactions with a remotely controlled educational robot. *ACM Transactions on Computing Education*, 19(4), 1-19.

Moreno, J., & Robles, G. (2014). Automatic detection of bad programming habits in scratch: A preliminary study. In M. Castro & E. Tovar (eds.), 2014 *IEEE Frontiers in Education Conference* (pp. 403-406). New Jersey: IEEE.

Moreno-León, J., & Robles, G. (2015a). Analyze your Scratch projects with Dr. Scratch and assess your Computational Thinking Skills. In C. Terosier et al. (eds.), *5th International Scratch Conference* (pp. 1-7). Amsterdam: Scratch2015AMS.

Moreno-León, J., & Robles, G. (2015b). Dr. Scratch: a web tool to automatically evaluate Scratch projects. In J. Gal-Ezer, S. Sentance & J. Vahrenhold (eds.), *Proceedings of the Workshop in Primary and Secondary Computing Education* (pp. 1-2). New York: ACM.

Moreno-León, J., Robles, G., & Román-González, M. (2016). Comparing computational thinking development assessement scores with software complexity metrics. In M. Al-Mualla, M.E. Auer & S. Al-Samahi (eds.), *Proceedings of the IEEE 2016 Global Engineering Education Conference* (pp. 1040-1045). New Jersey: IEEE.

Moreno-León, J., Robles, G., & Román-González, M. (2017). Can we measure computational thinking with tools? Present and future of Dr. Scratch. In G. Robles, H. Osman, A. Chis & F. Hermans (eds.), *SATToSE 2017: Seminar Series on Advanced Techniques & Tools for Software Evolution* (pp. 1-5). Madrid: Universidad Rey Juan Carlos.

Moreno-León, J., Román-González, M., Harteveld, C., & Robles, G. (2017). On the automatic assessment of computational thinking skills: A Comparison with Human Experts. In G. Mark et al. (eds.), *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems* (pp. 2788-2795). New York: ACM.

Mueller, J., Beckett, D., Hennessey, E., & Shodiev, H. (2017). Assessing computational thinking across the curriculum. In P.J. Rich & C.B. Hodges (eds.), *Emerging Research, Practice, and Policy on Computational Thinking, Educational Communications and Technology: Issues and Innovations* (pp. 251-267). Cham: Springer.

Mühling, A., Ruf, A., & Hubwieser, P. (2015). Design and first results of a psychometric test for measuring basic programming abilities. In J. Gal-Ezer, S. Sentance & J. Vahrenhold (eds.), *Proceedings of the Workshop in Primary and Secondary Computing Education* (pp. 2-10). New York: ACM.

Ota, G., Morimoto, Y., & Kato, H. (2016). Ninja code village for scratch: Function samples/function analyser and automatic assessment of computational thinking concepts. In A. Blackwell, G. Stapleton & B. Plimmer (eds.), *2016 IEEE Symposium on Visual Languages and Human-Centric Computing* (pp. 238-239). New Jersey: IEEE.

Palts, T., & Pedaste, M. (2017). Tasks for assessing skills of computational thinking. In R. Davoli, M. Goldweber, G. Rößling & I. Polycarpou (eds.), *Proceedings of the 2017 ACM Conference on Innovation and Technology in Computer Science Education* (pp. 367). New York: ACM.

Papavlasopoulou, S., Sharma, K., & Giannakos, M.N. (2019). Coding activities for children: Coupling eye-tracking with qualitative data to investigate gender differences. *Computers in Human Behavior*, 105, 1-11.

Papert, S. (1993). *Mindstorms: Children, Computers, and Powerful Ideas (2nd ed.)*. New York: Basic Books.

Park, T.H., Kim, M.C., Chhabra, S., Lee, B., & Forte, A. (2016). Reading hierarchies in code: Assessment of a basic computational skill. In A. Clear, E. Cuadros-Vargas, J. Carter & Y. Tupac (eds.), *Proceedings of the 2016 ACM Conference on Innovation and Technology in Computer Science Education* (pp. 302-307). New York: ACM.

Pérez, A.D.F., & Valladares, G.M. (2018). Development and assessment of computational thinking: A methodological proposal and a support tool. In C.S. González González, M. Castro, M. Llamas Nistal (eds.), *Proceedings of the 2018 IEEE Global Engineering Education Conference* (pp. 787-795). New Jersey: IEEE.

Pérez-Marín, D., Hijón-Neira, R., Bacelo, A., & Pizarro, C. (2018). Can computational thinking be improved by using a methodology based on metaphors and scratch to teach computer programming to children?. *Computers in Human Behavior*, 105, 1-10.

Petri, G., & Gresse von Wangenheim, C. (2017). How games for computing education are evaluated? A systematic literature review. *Computers & Education*, 107, 68-90.

Portelance, D.J., & Bers, M.U. (2015). Code and tell: Assessing young children's learning of computational thinking using peer video interviews with ScratchJr. In M.U. Bers & G.L. Revelle (eds.), *Proceedings of the 14th International Conference on Interaction Design and Children* (pp. 271-274). New York: ACM.

Rich, P.J., Egan, G., & Ellsworth, J. (2019). A framework for decomposition in Computational Thinking. In B. Scharlau, R. McDermott, A. Pears & M. Sabin (eds.), *Proceedings of the 2019 ACM Conference on Innovation and Technology in Computer Science Education* (pp. 416-421). New York: ACM.

Rodriguez, B., Kennicutt, S., Rader, C., & Camp, T. (2017). Assessing Computational Thinking in CS unplugged activities. In M.E. Caspersen, S.H. Edwards, T. Barnes & D.D. Garcia (eds.), *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education* (pp. 501-506). New York: ACM.

Rodriguez, B., Rader, C., & Camp, T. (2016). Using student performance to assess CS unplugged activities in a classroom environment. In A. Clear, E. Cuadros-Vargas, J. Carter & Y. Tupac (eds.), *Proceedings of the 2016 ACM Conference on Innovation and Technology in Computer Science Education* (pp. 95-100). New York: ACM.

Rojas Lopez, A., & Garcia-Penalvo, F. (2016). Relationship of knowledge to learn in programming methodology and evaluation of computational thinking. In F.J. García-Peñalvo (ed.), *Proceedings of the Fourth International Conference on Technological Ecosystems for Enhancing Multiculturality* (pp. 73-77). New York: ACM.

Román-González, M., Moreno-Leon, J., & Robles, G. (2017). Complementary tools for computational thinking assessment. In S.C. Kong, J. Sheldon & K.Y. Li (eds.), *Proceedings of the International Conference on Computational Thinking Education* (pp. 154-159). Hong Kong: The Education University of Hong Kong.

Román-González, M., Moreno-Leon, J., & Robles, G. (2019). Combining assessment tools for a comprehensive evaluation of computational thinking interventions. In S.C. Kong & H. Abelson (eds.), *Computational Thinking Education* (pp.79-98). Singapore: Springer.

Román-González, M., Pérez-González, J., & Jiménez-Fernández, C. (2017). Which cognitive abilities underlie computational thinking? Criterion validity of the Computational Thinking Test. *Computers in Human Behavior*, 72, 678-691.

Román-González, M., Pérez-González, J., Moreno-Leon, J., & Robles, G. (2016). Does computational thinking correlate with personality?: The non-cognitive side of computational thinking. In F.J. García-Peñalvo (ed.), *Proceedings of the Fourth International Conference on Technological Ecosystems for Enhancing Multiculturality* (pp. 51-58). New York: ACM.

Román-González, M., Pérez-González, J., Moreno-Leon, J., & Robles, G. (2018a). Extending the nomological network of computational thinking with non-cognitive factors. *Computers in Human Behavior*, 80, 441-459.

Román-González, M., Pérez-González, J.C., Moreno-Leon, J. & Robles, G. (2018b). Can computational talent be detected? Predictive validity of the Computational Thinking Test. *International Journal of Child-Computer Interaction*, 18, 47–58.

Rowe, E., Asbell-Clarke, J., Cunningham, K., & Gasca, S. (2017a). Assessing implicit computational thinking in Zoombinis gameplay. In S. Deterding et al. (eds.), *Proceedings of the 12th International Conference on the Foundations of Digital Games* (pp. 1-4). New York: ACM.

Rowe, E., Asbell-Clarke, J., Cunningham, K., & Gasca, S. (2017b). Assessing Implicit Computational Thinking in Zoombinis Gameplay: Pizza Pass, Fleens & Bubblewonder Abyss. In B. Schouten et al. (eds.), *CHI PLAY '17 Extended Abstracts: Extended Abstracts Publication of the Annual Symposium on Computer-Human Interaction in Play* (pp. 195-200). New York: ACM.

Royal Society (2012). *Shut down or restart: The way forward for computing in UK schools*. Retrieved 9 September 2018, from https://royalsociety.org/~/media/Royal_Society_Content/education/policy/computing-in-schools/2012-01-12-Computing-in-Schools.pdf.

Salac, J. (2019). Personalized Assessment Worksheets for Scratch (PAWS): Exploring a bridge between interviews, written assessments, and artifact analysis. In McCartney et al. (eds.), *Proceedings of the 2019 ACM Conference on International Computing Education Research* (pp. 351-352). New York: ACM.

Santisteban, J., & Santisteban-Munoz, J. (2018). Psychometric Computational Thinking Test. In I. Polycarpou, J.C. Read, P.G. Andreou & M. Armoni (eds.), *Proceedings of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education* (p. 393). New York: ACM. https://doi.org/10.1145/3197091.3205823.

Seiter, L., & Foreman, B. (2013). Modeling the Learning Progressions of Computational Thinking of Primary Grade Students. In B. Simon, A. Clear & Q.I. Cutts (eds.), *Proceedings of the ninth Annual International ACM Conference on International Computing Education Research* (pp. 59-66). New York: ACM.

Selby, C., & Woollard, J. (2013). *Computational thinking: the developing definition*. University of Southampton. Retrieved 12 September 2018, from https://eprints.soton.ac.uk/356481.

Sherman, M., & Martin, F. (2015). The assessment of mobile computational thinking. *Journal of Computing Sciences in Colleges*, 30(6), 53-59.

Snow, E., Rutstein, D., Bienkowski, M., & Xu, Y. (2017). Principled assessment of student learning in high school computer science. In J. Tenenberg et al. (eds.), *Proceedings of the 2017 ACM Conference on International Computing Education Research* (pp. 209-216). New York: ACM.

Srinivas, M.J., Roy, M.M, Sagri, J.N., & Kumar, V. (2018). Assessing Scratch programmers' development of computational thinking with transaction-level data. In S. Chakraverty, A. Goel & S. Misra (eds.), *Towards Extensible and Adaptable Methods in Computing* (pp. 399-407). Singapore: Springer.

Swanson, H., Anton, G., Bain, C., Horn, M., & Wilensky, U. (2019). Introducing and assessing computational thinking in the secondary science classroom. In S.C. Kong & H. Abelson (eds.), *Computational Thinking Education* (pp. 99-117). Singapore: Springer.

Teaching London Computing (n.d.). *Teaching London Computing: A resource hub from CAS London and CS4FN*. Retrieved 20 August 2019, from https://teachinglondoncomputing.org.

Troiano, G.M., Snodgrass, S., Argimak, E., Robles, G., Smith. G., Cassidy, M., Tucker-Raymond, E., Puttick, G., & Harteveld, C. (2019). Is my game OK Dr. Scratch?: Exploring programming and computational thinking development via metrics in student-designed serious games for STEM. In J.A. Fails (ed.), *Proceedings of the 18th ACM International Conference on Interaction Design and Children* (pp. 208-219). New York: ACM.

Tsarava, K., Leifheit, L., Ninaus, M., Román-González, M., Butz, M.V., Golle, J., Trautwein, U., & Moeller, K. (2019). Cognitive correlates of Computational Thinking: Evaluation of a blended unplugged/plugged-In course. In Q. Cutts & T. Brinda (eds.), *WiPSCE'19: Proceedings of the 14th Workshop in Primary and Secondary Computing Education* (pp. 1-9). New York: ACM.

University of Canterbury (n.d.). *CS Education Research Group: Computer Science Unplugged*. Retrieved 9 September 2018, from https://csunplugged.org/en.

Vilniaus Universitetas (n.d.). *Bebras - International Challenge on Informatics and Computational Thinking*. Retrieved 9 September, 2018, from https://www.bebras.org.

Webb, D.C. (2010). Troubleshooting assessment: an authentic problem solving activity for it education. *Procedia - Social and Behavioral Sciences*, 9, 903-907.

Weintrop, D., Beheshti, E., Horn, M., Orton, K., Trouille, L., Jona, K., & Wilensky, U. (2014). Interactive assessment tools for computational thinking in high school STEM classrooms. *INTETAIN 2014, Springer LNICST*, 136, 22-25.

Weintrop, D., & Wilensky, U. (2015). Using commutative assessments to compare conceptual understanding in blocks-based and text-based programs. In B. Dorn, J. Sheard & Q.I. Cutts (eds.), *Proceedings of the Eleventh Annual International Conference on International Computing Education Research* (pp. 101-110). New York: ACM.

Werner, L., Denner, J., & Campe, S. (2015). Children programming games: A strategy for measuring computational learning. *ACM Transactions on Computing Education*, 14(4), 1-22.

Werner, L., Denner, J., Campe, S., & Kawamoto, D.C. (2012). The fairy performance assessment: Measuring computational thinking in middle school. In L.A. Smith King, D.R. Musicant, T. Camp & P.T. Tymann (eds.), *Proceedings of the 43rd ACM Technical Symposium on Computer Science Education* (pp. 215-220). New York: ACM.

Wiebe, E., London, J., Aksit, O., Mott, B.W., Boyer, K.E., & Lester, J.C. (2019). Development of a lean computational thinking abilities assessment for middle grades students. In E.K. Hawthorne, M.A. Pérez-Quiñones, S. Heckman & J. Zhang (eds.), *Proceedings of the 50th ACM Technical Symposium on Computer Science Education* (pp. 456-461). New York: ACM.

Wilson, A., Hainey, T., & Connolly, T. (2012). Evaluation of computer games developed by primary school children to gauge understanding of programming concepts. In P. Felicia (ed.), *6th European Conference on Games Based Learning* (pp. 549-558). Cork: ACPL.

Wing, J.M. (2006), Computational thinking. *Communications of the ACM*, 49(3), 33–35.

Wing, J.M. (2011). *Research Notebook: Computational Thinking - What and Why?. The Link*. Retrieved 9 September 2018, from https://www.cs.cmu.edu/link/research-notebook-computational-thinking-what-and-why.

Yadav, A., Burkhart, D., Moix, D., Snow, E., Bandaru, P., & Clayborn, L. (2015). *Sowing the Seeds: A Landscape Study on Assessment in Secondary Computer Science Education*. New York: CSTA.

Yagci, M. (2019). A valid and reliable tool for examining computational thinking skills. *Education and Information Technologies*, 24, 929-951.

Zhang, L., & Nouri, J. (2019). A systematic review of learning computational thinking through Scratch in K-9. *Computers & Education*, 141, 1-25.

Zhang, N., & Biswas, G. (2019). Defining and assessing students' computational thinking in a learning by modeling environment. In S.C. Kong & H. Abelson (eds.), *Computational Thinking Education* (pp. 203-221). Singapore: Springer.

Αναφορά στο άρθρο ως: Πουλάκης, Ε., & Πολίτης, Π. (2019). Συστήματα αξιολόγησης της Υπολογιστικής Σκέψης στην εκπαίδευση: Βιβλιογραφική επισκόπηση. *Θέματα Επιστημών και Τεχνολογίας στην Εκπαίδευση*, 12(2), 99-119.

http://earthlab.uoi.gr/thete/index.php/thete