

Računalno razmišljanje i uloga učenja pomoću igre na njegov razvoj

Krešo Tomljenović
Učiteljski fakultet Sveučilišta u Zagrebu
Savska cesta 77, 10000 Zagreb
kreso.tomljenovic@ufzg.hr

Sažetak - Računalno razmišljanje je misaona aktivnost formuliranja i rješavanja problema na način kako bi to izvelo računalo, iako ne nužno uz pomoć računala. Smatra se neophodnom vještinom za uspješno funkcioniranje u tehnologijom prožetom društvu 21. stoljeća stoga je važna njegova integracija u obrazovni sustav. U radu se definira računalo razmišljanje, njegove odlike i model te daje pregled recentne literature o aktivnostima pri implementaciji u edukaciji. Ističu se primjeri razvoja računalnog razmišljanja pomoću dostupnih platformi za učenje kroz igru, dizajn igara, natjecanje, robotiku itd., u skladu sa suvremenim teorijama učenja.

Glavne riječi – računalo razmišljanje; STEMI; obrazovanje; informacijsko-komunikacijske tehnologije; učenje pomoću igre

I. UVOD

Tehnološki napredak 21. stoljeća nameće digitalnu pismenost ostalim općepriзнatim temeljnim vještinama: kritičko razmišljanje, pismeno izražavanje, čitalačka pismenost, matematička pismenost.

Prema pedagogu Karlu Fischu, profesoru srednje škole u SAD i autoru motivirajućih prezentacija o promjenama koje 21. stoljeće donosi [1] „Trenutno pripremamo studente za poslove koji još ne postoje, za koje će koristiti tehnologije koje još nisu izumljene da bi rješavali probleme za koje još niti ne znamo da su problemi.“

Prije više od desetak godina u medijima se često čulo kako „tržište rada zahtijeva n-tisuća radnika“ koji znaju koristiti informacijsko-komunikacijske tehnologije (IKT), kako tržište vapi za programerima. Iste vijesti su i danas aktualne, samo su se povećale brojke o kojima se govori. Primjerice, iz Eurostat podataka može se vidjeti da je u periodu od 2013. do 2018. godine postotak nepopunjenim radnih mjesta u EU, u području IKT, narastao sa 2% na 3% [2]. Za svakodnevno funkcioniranje svi danas moramo biti u korak s IKT. Bilo da se trebamo snaći u prijevozu, online predati razne zahtjeve ili pribaviti dokumente, obaviti trgovinu... sve bismo teže obavljali bez IKT.

Nameće se potreba kako postati digitalni „proizvođač“ a ne samo „konzument“ u svijetu u kojem se nove tehnologije stvaraju brzinom nepremostivom za tromi sustav formalnog školovanja.

U tom kontekstu se edukacija u STEM (eng. Science, Technology, Engineering, Mathematics) je područje prirodnih znanosti, tehnologije, inženjerstva i matematike, kojima se ponekad dodaje i područje informatike - STEMI) području apostrofira kao ključno u razvoju inovacija i tehnologija a samim time i gospodarskog napretka [3].

U posljednja četiri ciklusa PISA testiranja hrvatskih učenika u svim ispitivanim kategorijama pokazani su ispodprosječni rezultati u odnosu na prosjek učenika u svijetu (uzrast 15 i 16 godina) [4].

S druge strane anketa provedena među 144 studenta prve i druge godine Učiteljskog fakulteta u Zagrebu (budući učitelji i odgojitelji) je pokazala da je svega 27% studenata na maturi položilo višu razinu matematike (A razina), te da je svega 9% anketiranih izrazilo „izrazitu“ sklonost STEM područjima (88% je izrazilo „osrednju“ a 3% „nimalo“) [5]. Za očekivati je da će pomanjkanje interesa za STEM područje budući učitelji prenijeti na nove generacije učenika te rezultirati lošim uspjehom na objektivnim testiranjima.

Nameće se potreba osnaživanja kritičko-algoritamskog razmišljanja (uz druge aspekte koje potiču STEMI način razmišljanja) kroz cjelokupnu nastavu, ne ciljajući samo kroz matematiku ili programiranje. Ideja je osposobiti učenike i studente nesklone STEMI području da usvoje bitne koncepte i vještine karakteristične upravo za STEMI.

Početkom 21. stoljeća vodeći autoriteti industrije i vlade SAD-a su uočili da gube zamah u inovacijama i kreativnim novim tehnologijama pa je National Science Foundation (NSF) odlučio povećati financiranje studija STEMI područja [6].

Nakon inspirirajućeg rada Jeanette M. Wing iz 2006. godine *Computational thinking* (eng. računalo razmišljanje, nadalje u tekstu CT) [7] koji zagovara univerzalno primjenjive stavove i vještine koje obično povezujemo upravo sa STEMI područjem, NSF je angažirao The International Society for Technology in Education (ISTE) i Computer Science Teachers Association (CSTA) da formuliraju definiciju CT te izdaju smjernice kako bi svi učenici do završetka srednje škole trebali usvojiti vještine računalnog razmišljanja.

2010. godine publiciraju *Computational thinking toolkit*, priručnik s operativnom definicijom CT, smjericama, konkretnim primjerima primjene po dobnim skupinama od predškolske dobi do završetka srednjoškolskog obrazovanja [6].

U ovome radu se analizira pojam CT-a koji je zadnjih 12 godina, od rada J. M. Wing prihvaćen kao temeljni pojam odnosno nadgradnja pojma algoritamskog razmišljanja uz mnoštvo drugih vrijednosti; daje se njegova definicija i domena, te pregled istraživanja i interesa u posljednjih desetak godina.

Naglasak je na CT u obrazovanju; u prvom redu predškolskog do srednjoškolskog (PK-12) i zatim razvoj CT kod studenata, naročito edukacija budućih učitelja i odgojitelja.

II. RAČUNALNO RAZMIŠLJANJE – DEFINICIJA, DOMENA, MODEL

Rad J.M. Wing iz 2006.godine se apostrofira kao najutjecajniji i najcitiraniji, iako je prvi oblikovao termin S. Papert u svojoj teoriji učenja konstrukcionizam [8].

Konstrukcionizam je ujedno i teorijski temelj računalnog razmišljanja; to je novija teorija učenja (izvedena iz konstruktivizma [9]) u kojoj se tehnologiji stavlja naglasak na pedagoškom pristupu u procesu usvajanja znanja. Učenje se odvija na prirodan način kroz aktivno sudjelovanje u dizajnu okoline (mikrosvijeta) i istraživanja svojstva okoline.

Teorija učenja pomoću igre (*game based learning* – GBL) je u tijesnoj vezi s izgradnjom takvih mikrosvijetova. Igranjem se na prirodan način djeca motiviraju za proces učenja [10]. Razvojem IKT događa se prijelaz s igara za učenje (kroz koje će se sadržaj usvojiti) na igre koje učenik treba aktivno izgraditi.

Definicija CT

S obzirom na široko područje primjene CT nema jednoznačnu definiciju nego više skup vještina koje uključuje. Mnogi autori daju svoje inačice pojma CT, koje se manje ili više preklapaju u sastavnicama.

Većina autora se slaže da je CT skup fundamentalnih vještina za svakoga, ne samo informatičare; to je misaona aktivnost formuliranja i rješavanja problema na način kako to čine računala. Pritom postupak rješavanja problema ne uključuje nužno sama računala. [11,12]

Jednu od citiranih definicija su dali Cuny, Snyder i Wing proizašlu iz njihove korespondencije, u neobjavljenom radu iz 2010. g.: „CT je misaoni proces uključen u formuliranje problema i njihovih rješenja u formi koju može efikasno obraditi agent za obradu informacija [13].

Opširnija definicija je dana od strane C. Selby i J. Woollarda u kojoj navode CT kao „misaonu aktivnost u rješavanju problema koja sadržava apstrakciju, dekompoziciju, algoritamski dizajn, evaluaciju i generalizaciju“ [14].

Spomenuta publikacija *Computational thinking toolkit* iz 2010. godine, sinergijom velikog broja vodećih znanstvenika, profesora na edukacijskim fakultetima, školskih profesora i učitelja SAD-a, daje još potpuniju definiciju CT-a [6]:

CT je proces rješavanja problema koji uključuje (među ostalim) karakteristike:

- Formuliranje problema na način da je moguće koristiti računala i druge alate pri rješavanju
- Logičko organiziranje i analiziranje podataka
- Reprezentiranje podataka kroz apstrakcije kao što su modeli i simulacije
- Izvođenje rješenja u algoritamskim koracima
- Identificiranje, analiziranje i primjena mogućih rješenja s ciljem postizanja najefikasnije kombinacije koraka i resursa
- Generalizacija i prijenos postupka na rješavanje drugih problema

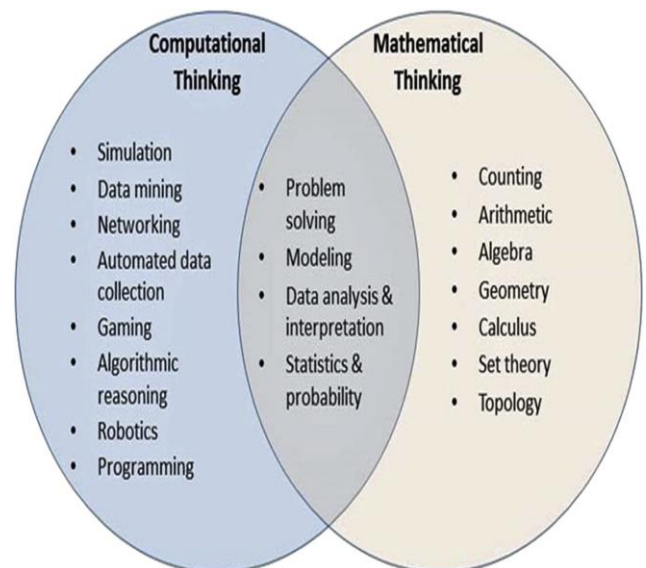
Dimenzije CT-a s obzirom na karakteristike i stavove uključuju [6]:

- Samouvjerenost u kompleksnim problemima
- Upornost u radu s teškim problemima
- Prihvatanje mogućih višestrukih rješenja
- Sposobnost nošenja s problemima
- Sposobnost komunikacije i rada s drugima u cilju postizanja zajedničkog cilja ili rješenja

Usvajanje kompetencija i vještina CT-a može biti motivacijski čimbenik usmjeravanju studenata ka studijima STEMI područja [15, 16], što je i bio jedan od uzroka angažmana velikog broja stručnjaka u SAD-u početkom stoljeća.

Domena CT

Karakteristike CT-a u odnosu na druge oblike razmišljanja se često podudaraju, pa se primjerice razmatraju razlike i sličnosti računalnog u odnosu na matematičko razmišljanje [16,17], u odnosu na inženjerski oblik razmišljanja [18], u odnosu na dizajnerski oblik razmišljanja [19], i td. Na Slici 1. su prikazani aspekti (domena) računalnog razmišljanja u odnosu na matematičko razmišljanje - sličnosti i razlike [16].



Slika 1. Sličnosti i razlike računalnog i matematičkog razmišljanja. Preuzeto iz [16].

Model

Model je način raščlanjivanja aspekata CT-a na konceptijski bitne sastavnice.

CT u sebi sadržava niz karakteristika koje se sistematiziraju modelom aspekata. Ni u tome području još nije postignuto suglasje, već više autora nude svoje modele CT aspekata [20, 21, 22, 23].

U ovom radu prikazan je model V.J.Shute, C. Sun i J. Asbell-Clarke predstavljen u preglednom članku *Demystifying computational thinking* iz 2017.g. Autori su dali opširan pregled literature, uspoređivali i analizirali

modele drugih autora, te temeljem toga sumirali najvažnije aspekte u svoj model CT-a. Model obuhvaća 6 glavnih aspekata: dekompozicija, apstrakcija, algoritmi, otklanjanje pogrešaka, iteracija i generalizacija. [24]

U Tablici 1. je detaljnije prikazan model CT aspekata uz opise ([24] str.153).

TABLICA 1. Aspekti CT i njihova definicija

Aspekt	Definicija
Dekompozicija	Rastaviti kompleksnije probleme/sustave na manje, lakše rješive dijelove-funkcionalne podcjeline
Apstrakcija	Uočiti bit problema. Sastoji se od podcjelina: a) Prikupljanje podataka i analiza: sakupiti najrelevantnije podatke iz više izvora i shvatiti vezu među njima b) Prepoznavanje uzorka: identificirati uzorak/pravilnosti u strukturi podataka c) Modeliranje: izgradnja modela ili simulacija funkcioniranja sustava i/ili kako će sustav funkcionirati u budućnosti
Algoritmi	Dizajniranje logičkih i uređenih instrukcija za postizanje rješenja problema. Instrukcije mogu iznijeti ljudi ili računala. Potkategorije su: a) Dizajn: kreiranje niza određenih koraka ka rješenju problema b) Usporednost: izvršavanje određenog broja koraka u isto vrijeme c) Efikasnost: dizajniranje minimalnog broja koraka za rješavanje problema, uklanjanje nepotrebnih koraka d) Automatizacija: automatiziranje izvođenje pri rješavanju sličnih problema
Otklanjanje pogrešaka	Detektirati, identificirati i popraviti pogreške u slučaju da rješenje problema ne funkcionira.
Iteracija	Ponavljanje procesa dok se ne postigne idealno rješenje.
Generalizacija	Prenošenje CT vještina na široki krug situacija u cilju efikasnog rješavanja problema

Model, na koji se u literaturi referencira najveći broj autora, predlažu Brennan i Resnick [25]. U svojim istraživanjima autori su proučavali aktivnosti učenika u radu sa Scratch alatima, tj. pri dizajniranju interaktivnih priča, igara i simulacija, a zatim i kroz online komunikaciju, dijeljenje i reinterpretiranje svojih uradaka. Učenje i kompleksni razvoj koji se pritom događa sadrži mnoge odlike upravo CT-a te autori smatraju da takve aktivnosti temeljene na dizajnu podupiru razvoj CT u mlađih učenika. U svom modelu autori aspekte CT-a razlažu u tri dimenzije [25]:

- *Koncepti* koje student koristi pri programiranju: nizovi naredbi, petlje, paralelnost, događaji, algoritmi, uvjeti, operatori, podaci
- *Prakse* pri programiranju: eksperimentiranje i iteracije, testiranje i otklanjanje pogrešaka, višekratno korištenje i kompiliranje, apstrahiranje i moduliranje
- *Perspektive* koje uključuju učenikovo razumijevanje sebe u odnosu na druge ljude i digitalni svijet oko njega: izražavanje, povezivanje i propitivanje

III. AKTIVNOSTI U RAZVOJU CT

Iako su otpočetak aspekti i vještine CT apostrofirane kao neophodne za opću populaciju u informacijsko-komunikacijskom dobu današnjice [24], upravo se glavni fokus istraživanja stavlja na edukaciju. To je prilično i razumljivo s obzirom da se uspješno obrazovanje današnjice reflektira na inovativne i kreativne ljude sutrašnjice a time uspješniju i konkurentniju ekonomiju i gospodarstvo

Pritom se posebno razmatra primjena u edukaciji obaveznog školovanja (tzv. K-12 obrazovanju iako u R. Hrvatskoj formalno još nije obaveznih 12 godina školovanja) i edukaciji budućih učitelja.

U svojem članku Araujo i dr. [26] analiziraju radove objavljene u periodu od 7. mjeseca 2015. do 4. mjeseca 2016. godine te analiziraju pristupe promociji CT-a, kojoj ciljnoj skupni su namijenjeni, te koji su instrumenti evaluacije uspješnosti u razvoju CT. Kao relevantne su izdvojili 27 radova u tom periodu. U Tablici 2. su prikazani pristupi i učestalost unutar tih 27 studija.

TABLICA 2. Broj studija s obzirom na pristupi promociji CT-a i ciljnu skupinu

		Broj studija
Pristup	Nastava, radionice, tečajevi	13
	Test	6
	Izrada modela	5
	Alati (igre, platforme)	3
Ciljana skupina	Predškolski uzrast	1
	Učitelji/nastavnici	2
	Studenti	8
	Učenici osnovnih i srednjih škola	15

S obzirom na veliki broj aspekata CT-a većina autora se u svojim istraživanjima fokusira na točno određene aspekte, te dizajniraju istraživanje na način da se razvoj upravo ciljanih aspekata mogu evaluirati. U pregledu dostupne literature Araujo i dr. [26] navode najčešće istraživane aspekte (redom od najučestalijih): algoritam, apstrakcija, dekompozicija, reprezentacija podataka, procedure, usporednost, nizovi i petlje, itd.

U radu Botički i dr. [27] fokusiraju se na razvoj koncepata CT na uzorku 23 učenika prvog razreda osnovne škole. Konkretno su promatrani koncepti: nizovi, algoritmi, objekti i svojstva, petlje i problemski zadaci. Zadaci su vezani za predmete matematika, hrvatski jezik i priroda, te kreirani na platformi *Blockly*, što je JavaScript biblioteka za programiranje [28]. Uspoređujući postignuća na testu s uspjehom iz navedenih predmeta zaključili su da učenici uspješni u matematici i čitanju postižu bolje rezultate u promatranim CT konceptima.

García-Penalvo i Mendes u svom radu [29] predlažu tri pristupa u razvoju računalnog razmišljanja:

- Razvoj modela – Glavna ideja iza ovog pristupa jest da CT nije alternativa učenju programiranja nego način osnaživanja koncepata i ideja obično povezana uz programiranje. Cilj je da učenici

kroz osmišljene projekte razviju misaone modele koji su primjenjivi u širokom rasponu disciplina.

- Alati za programiranje – Iako CT nije vezano samo uz programiranje, ishod dobro planiranih aktivnosti pri programiranju može biti upravo razvoj CT vještina. Pritom je važan odabir samih programskih jezika, konkretno vizualnog tipa, kako bi se istaknula ciljana komponenta CT
- Intervencija u kurikulumu – Ističe se važnost implementacije CT vještina u kurikulumu kao i edukaciju budućih učitelja.

A. Razvoj misaonih modela

CS Unplugged project je projekt osmišljen za učenje kako funkcionira računalo, za uzrast 5 do 10 godina, kroz niz osmišljenih igara i aktivnosti bez upotrebe računala [30]. Autori su razradili niz aktivnosti za razvoj aspekata CT koje se slobodno mogu dijeliti i unaprjeđivati [31].

Bebras natjecanje

Bebras (lit. Dabar) natjecanje je pokrenuto u Litvi 2003.godine s ciljem promocije informatičke edukacije u školama i upotrebe informacijsko-komunikacijskih tehnologija. Natjecanje je već 2006.g. postalo međunarodno, održava se svake godine, s preko 60 zemalja suučesnica i nekoliko milijuna natjecatelja [32]. Zahvaljujući posvećenosti autora vrlo brzo je natjecanje preraslo u kompleksni projekt u razvoju koncepata CT-a [33]. R. Hrvatska se uključila u natjecanje 2016.g. pod nazivom *Dabar*, a 2017. godine je sudjelovalo 15247 učenika u 5 kategorija [34].

Po svom inicijalnom karakteru natjecanje je slično većem i starijem (započeto 1978. g. u Australiji) međunarodnom natjecanju *Matematički Klok* osmišljenim za popularizaciju matematike i motiviranje učenika da se bave matematikom izvan redovitih školskih programa [35].

Oba natjecanja, uz prvotnu orijentaciju na predmet informatike i/ili matematike, pokazala su se vrlo motivirajuća za usvajanje koncepata CT-a. Velika prednost natjecanja *Bebras* je što se provodi na računalima, zadaci su interaktivni, omogućavaju učenicima među ostalim da kroz igru istražuju dinamička svojstva, testiraju svoje rješenje, itd., čime obuhvaćaju veliki broj odlika CT-a kao i učenja pomoću igre.

Ambiciozno se pristupa što kvalitetnijoj pripremi zadataka: moraju biti zanimljivi, interaktivni, adekvatni dobi i vremenskom ograničenju, po mogućnosti duhoviti i da unose element igre, problemski postavljen zadatak iz različitih sfera znanosti ili svakodnevnog života i sa stvarnim podacima. U tu svrhu organiziraju se radionice za izradu zadataka kroz koje je osmišljen i kategoriziran niz kriterija za „dobar *Bebras* zadatak“ [36].

S ciljem stvaranja okoline za efikasno upravljanje samim natjecanjem dizajniran je *Bebras contest management system (CMS)*, dizajniran kao web aplikacija, funkcionalno modularan, sa podsustavima za: [37]

- upravljanje sustavom (sigurnost, resursi)
- upravljanje korisnicima (registracija, profili)
- upravljanje natjecanjem (kreiranje, nadzor)

- administracija popisa škola
- upravljanje zadacima (kreiranje, uvoz)
- upravljanje rezultatima i diskusijama (statistika)

U *Bebras CMS*-u je moguće kreirati zadatke višestrukog izbora, no ne i interaktivne dinamične zadatke. U tu svrhu je dizajniran *Bebras Lodge* - alat za programiranje i implementiranje dinamičkih zadataka [38]. Sami alat se sastoji od dijela s komponentama (JavaScript biblioteka za kreiranje i manipuliranje komponentama unutar zadatka) te sučelja [39].

App Inventor je MIT-ov projekt za programiranje i izradu aplikacija na vizualno jednostavan način. Objekti se slažu drag-and-drop metodom, bez kompleksnog programiranja [40]. Projekt je zamišljen da svatko (a naročito mladi bez pretjeranog znanja programiranja) može u kratkom vremenu napraviti aplikaciju koja će utjecati na zajednicu, što djeluje vrlo motivirajuće i inspirativno na učenike. Tvorci su cijeli projekt izdigli na socijalno angažiranu razinu, te od alata za programiranje postali model koji promovira sustav vrijednosti uz aspekte CT-a.

Lego Mindstorms je niz hardversko-softverskih paketa za kreiranje programabilnih robota. Sadrže računalo za kontrolu sustava, niz senzora i motora. U suradnji s MIT Media Laboratorijem se koriste kao edukacijski alat [41].

B. Alati za programiranje

Vrlo popularan programski jezik za učenje programiranja, modeliranja i matematike je *Scratch* [42], projekt pokrenut od Lifelong Kindergarten Group, MIT Media Lab. Svi sadržaji su besplatni za dijeljenje, istraživanje i nadogradnju. *Scratch* je vrlo jednostavan za korištenje pa je primjeren i za niže uzraste, a nudi smislenu okolinu za razvoj kreativnosti, sistematičnosti i kooperativnosti [24]. Koristeći *Scratch* učenici dizajniraju svoje priče, igre, animacije i simulacije slažući blokove već programiranog koda, kao da slažu *Lego* kockice [25]. Učenje pomoću igre je novija teorija učenja, prihvaćena kao motivirajuća za učenike, no pritom se pravi razlika između igranja igre i izgradnje same igre. U potonjem slučaju motivacija učenika je veća a time i proces učenja efikasniji [43]. *Scratch* je pogodna okolina upravo za konstrukcionistički aspekt GBL, pogodan za učenike nižih razreda i razvoj koncepata programiranja kroz dizajn igre [44].

Googleova inačica *Scratcha* je *Blockly* [28]. U cilju što bolje pristupačnosti i promoviranja CT vještina u edukaciji Google i MIT Media Lab su uspostavili suradnju u razvijanju sadržaja.

U nastavi informatike u osnovnoj školi u Hrvatskoj zastupljeni su uglavnom *Logo* i *Python*. Novim kurikulumom predmeta Informatika nije apostrofirani niti jedan obavezan programski jezik već je učitelju dana autonomija u izboru, u skladu s njegovim kreativnim zamislima [45].

Logo je među prvim programskim jezicima prikladnim za mlađi uzrast učenika a da pritom kroz dizajnirane aktivnosti promovira odlika CT-a (uz mnoge matematičke koncepte) kao što su apstrakcija, iteracija, algoritam, itd. Nastao je još 1960-ih kao edukacijski alat za učenje matematičkih koncepata u skladu s konstruktivističkom teorijom učenja. Osmislili su ga Papert, Seymour i dr. [46] te

je predmetom mnogih istraživanja još od 1980.-ih i široke primjene osobnih računala. Razvojem mnogih edukacijskih softvera i naročito komunikacijske umreženosti gubi na popularnosti.

Igre i dizajn igara je popularan način učenja i na tom tragu su brojna istraživanja kao i nastale brojne aplikacije [47]. Rode i dr. [48] ističu prednosti igrifikacije učenja CT-a, bilo da je riječ o igri u fizičkim svijetu ili računalnoj igri. Papastergiou [49] uspoređuje uspjeh dviju grupa učenika, od kojih je prva gradivo obrađivala pomoću računalne igre, a druga bez. Rezultati pokazuju znatno bolje rezultate prve grupe i u znanju i u motivaciji za gradivo, a nevezano za spol ispitanika.

Dobar primjer gdje se s gotovo nimalo iskustva u programiranju može kreirati vlastite igre je *AgentSheets/AgentCubes* [50]. Pozitivna korelacija između učenja pomoću igre (GBL) i motivacije a samim time i razvoja CT-a je dodatno naglašeno pri kreiranju igre [51].

C. Intervencija u kurikulumu

Većina budućih učitelja nema jasnu viziju kako CT ukomponirati u nastavni proces [52]. Važan je kurikulum koji će voditi računa o suvremenim prohtjevima nastavnog procesa.

U novom kurikulumu predmeta Informatike, koji je postao obavezan predmet za 5. i 6. razred osnovne škole dok za ostale razrede još uvijek nije obavezan, navodi se: „Težište obrazovnog procesa u predmetu Informatika treba biti na rješavanju problema i programiranju kako bi se poticalo razvijanje računalnog načina razmišljanja koje omogućuje razumijevanje, analizu i rješavanje problema odabirom odgovarajućih strategija, algoritama i programskih rješenja. Takvi se načini razmišljanja trebaju prenositi i u druga područja posebice matematičko i prirodoslovno, kao i u svakodnevni život“ [45].

Predmet se organizira u 4 domene: e-Društvo, Digitalna pismenost i komunikacija, Računalno razmišljanje i programiranje te Informacije i digitalna tehnologija. Zatim se daju konkretni odgojno-obrazovni ishodi prema razredima i domenama. Na kraju su smjernice za učenje i poučavanje predmeta, koje daju učitelju autonomiju u odabiru metoda i izvora [45].

Umrežavanjem na globalnoj razini, a pogotovo susretljivošću aktera obrazovnog procesa u dijeljenju sadržaja, sve je više dostupnih materijala i ideja za primjenu u nastavi. Neki od repozitorija su:

Barefoot Computing je projekt za podršku učiteljima u Ujedinjenom Kraljevstvu, a prati propisani plan i program. Namijenjen je razvijanju CT kompetencija za učenike uzrasta 5-11 godina [53].

Computing at School je projekt za promociju edukacije informacijskih znanosti, namijenjen za osnovnu i srednju školu [54].

Teaching London Computing nudi materijale i radionice za učitelje osnovnih i srednjih škola [55].

Computational Thinking for Educators (CS4HS) je Googleov besplatni online tečaj za pomoć nastavnicima u integriranju CT u kurikulum srednjih škola [56].

Resources je skup radionica u programu *Scratch* za nastavnike srednje škole [57].

Istraživanja o poticanju CT-a su najbrojnija za učenike srednjih škola i sveučilišta, u manjem broju se odnose na predmetnu nastavu osnovne škole, a najmanje istraživanja se odnose na predškolski uzrast i razrednu nastavu [58]. To je donekle razumljivo, s obzirom da se još uvijek CT veže uz predmete i aktivnosti u kasnijim godinama školovanja. Najlakše i najuobičajenije je CT uvoditi u sklopu predmeta informatike i programiranja. Međutim nužno je vještine i stavove koji se vezuju uz CT razvijati od najranijeg uzrasta, pa je neophodno intervenirati u kurikulum i staviti naglasak na edukaciju odgojitelja i učitelja [59]. U prijedlogu kurikulumu međupredmetne teme Uporaba informacijsko-komunikacijske tehnologije uglavnom se navodi kritičko vrednovanje i upotreba informacije, što ima dodirnih točaka sa konceptima CT [60].

Igrajući igre djeca demonstriraju mnoge kvalitete CT-a [61]. U ranim godinama školovanja (predškolska dob do 4. razreda) djeca su još motivirana za učenje jer ga doživljavaju više kao igru a ne obavezu. U to doba, dok se još formira njihov interes, važno je usmjeriti njihov proces učenja na načine koje promovira CT. Upravo GBL pristup učenju zahtjevnijih sadržaja kod djece zadržava osjećaj igre a ne prisile, što je u skladu sa suvremenijim teorijama učenja [62].

IV. CT I EDUKACIJA UČITELJA

CT aspekti se mogu učiti u nizu školskih predmeta, ne bi trebalo vezati samo za matematiku, informatiku ili prirodne predmete. Yadav i dr. u svom istraživanju nakon provedene radionice s budućim učiteljima o shvaćanju CT-a zaključuju da budući učitelji doživljavaju CT kao integriranje tehnologije u nastavu, dok eksperimentalna grupa nakon provedene edukacije shvaća CT kao pristup rješavanju problema, univerzalno primjenjiv u nastavi [63].

Umjesto fokusiranja na primjenu novih tehnologija, edukacija budućih učitelja na sveučilištima bi se trebala koncentrirati na usvajanje vještina i kompetencija koje odlikuje CT u sklopu svoje primarne struke i metodike, a koje se lako primjenjuje na novitete u tehnologiji [64].

S obzirom na „univerzalnost“ učitelja razredne nastave, od kojih se očekuje educiranost u svim predmetima, dobra edukacija učitelja je jedna od ključnih čimbenika u promociji CT u nižim razredima osnovne škole [65]. Stalnom i brzom promjenom u novim tehnologijama i mogućnostima koje one donose, neophodna je stalna edukacija učitelja. Praktični su se pokazali tečajevi i radionice u kojima se približavaju ideje CT-a kroz praktične primjere implementacija ideja i alata za provedbu u nastavi. S obzirom na ubrzano rastući broj alata i resursa dostupnih za slobodno korištenje u nastavi, želja i interes učitelja u poučavanju CT-a je najveća prepreka.

Lockwood i Mooney u preglednom radu iz 2017.g. [66] navode preko 50 raznih alata, softvera, programskih jezika itd. koja su bila temelj istraživanja CT-a u edukaciji. U Tablici 3. su nabrojani, uz napomenu da se broj stalno povećava kao i sadržaji za slobodno dijeljenje i korištenje temeljeno na tim alatima.

TABLICA 3. Alati iz istraživanja CT, iz preglednog rada [66]

Agentsheets/AgentCubes	Greenfoot HTML
Algo.Rhythm	Java
Alice	Lego Mindstorms
App Inventor	Lego WeDo
Ardublock	Lightbot
Arduino	Lilypad Arduino
Bebras	littleBits
Binary toy	Logo
BingBee	Maple
Blockly	MATLAB
Bord Game - Pandemic	Minecraft
Bunny Bright	NetLogo
CargoBot	Program your robot
CHERP	Pyonkee
Code Bits	Python
CS Unplugged	RaBit EscApe
CTArcade	RAPTOR
CTSiM CyberPLAYce	RuBot
DigitMile	Scratch
Dragon Architect Drawing Machine Model	Simulation Creation Toolkit
Entry	STAGE
Escape Machine	SUMO
Game Maker	The Incredible Machine
	VPython

Yadav i dr. [63] predlažu da edukacija budućih učitelja mora uključivati suradnju sa stručnjacima informacijskih znanosti, kako bi temeljne kvalitete CT-a ukomponirali u pedagošku i metodičku edukaciju. Predlažu da takva edukacija mora voditi računa o:

- Kurikulumu: kurikulum mora pripremiti učitelje za uključivanje CT u nastavu
- Ideje: buduću učitelji moraju imati predodžbu o temeljnim idejama i konceptima CT
- Metodički kolegiji: dizajnirati kolegije metodike za usvajanje CT principa u kontekstu predmeta
- Suradnja: edukatori informacijskih znanosti i edukatori učitelja moraju surađivati u izradi kurikuluma učiteljskih studija
- Edukacija učitelja: koristiti postojeće resurse i standarde za asimilaciju CT-a u edukaciju budućih učitelja.

V. PROCJENA RAZINE POSTIGNUĆA RAZVOJA CT

S obzirom da je područje računalnog razmišljanja relativno novo, a k tome uključuje mnoge aspekte i koncepte, nije još razvijen općeprihvaćen standard procjene postignuća [24,58]. Razne ankete su najčešći oblik ispitivanja stavova i sklonosti [26].

Programski jezici vizualnog tipa su najpopularniji pristup ne samo stimuliranju CT-a nego i procjeni CT-a [66].

Na temelju *Scratch* programiranja, Brennan i Resnick su razvili metodu procjene CT vještina Scratch korisnika u dobi 8-17 godina. Metoda uključuje tri pristupa, redom: formativnu analizu, anketu i dizajn scenarij. Zatim su svaki pristup analizirali s obzirom na svoj model CT, s obzirom na koncepte, prakse i perspektive. Sami autori su ustvrdili da

metoda ima velika ograničenja (vrijeme potrebno za provođenje evaluacije, broj učesnika) i upitan rezultat [21]. Nakon sva tri pristupa mogli su se zadovoljavajuće procijeniti koncepti CT, u malo manjoj mjeri prakse CT, no gotovo uopće ne i perspektive CT.

Dr.Scratch [67] je alat za evaluaciju Scratch projekata. Moreno-Leon i dr. [68] su uspoređivali rezultate evaluacije u Dr.Scratch platformi s evaluacijom na temelju metrika koje se koriste u projektiranju i evaluaciji kompleksnosti softvera, te su dobili statistički značajne podudarnosti, no sami uočavaju mane i predlažu daljnji rad.

Sve je više radova temeljeno na evaluaciji razvoja CT-a na temelju spomenutog *Bebras* natjecanja, što je i za očekivati s obzirom na sve veću masovnost tog natjecanja i njegovo kvalitetno osmišljavanje. U istraživanju provedeno na studentima prve godine kolegija programiranja u Brazilu nije se dobila nedvosmislena povezanost rezultata na odabranim „Bebras zadacima“ i razvoja CT-a, te se predlaže daljnji rad u osnaživanju takvih kompleksnih zadataka kao instrumenata procjene CT-a [66]. Slično istraživanje na studentima prve godine kolegija programiranja u Finskoj je pokazalo da postoji pozitivna veza rezultata na odabranim zadacima Bebras kao instrument za evaluaciju CT-a [69].

Koh i dr. [70] su predstavili uspješan projekt evaluacije učeničkih projekata dizajniranja igara na AgentSheet platformi, *REACT* sustav (Real time Evaluation and Assessment of Computational Thinking), no i sami autori predlažu daljnji rad i poboljšanja.

VI. ZAKLJUČAK

Iako je pojam računalnog razmišljanja potekao iz informacijsko komunikacijskih znanosti, danas treba razlikovati domenu informatike (programiranje, informacijske znanosti) od domenu računalnog razmišljanja.

Računalnom razmišljanju u obrazovanju nije cilj osposobiti za programiranje ili obrazovati informatičare već učenje vještina rješavanja problema digitalnog svijeta, odnosno obrazovanja učenika 21.stoljeća.

Poseban je naglasak na obrazovanju budućih učitelja kako bi oni u svoje nastavne sadržaje uspješno ukomponirali koncepte računalnog razmišljanja. Pritom nije naglasak na korištenju tehnologije kao takve nego uvođenju vještina i kompetencija računalnog razmišljanja u sve sadržaje i pedagoške temelje edukacije, uz tehnologiju kao potporu a ne kao nužnost [71].

Najvrjedniji rezultat ideje „CT za sve a ne samo za informatičke stručnjake i programere“ je u stvaranju niza aktivnosti – projekata, natjecanja, repozitorija koja implementiraju suvremene načine učenja: učenje temeljeno na igri, kroz razne oblike interaktivne kompeticije, učenje kroz solidarno dijeljenje znanja i resursa među sudionicima procesa učenja itd. Pri tome je suvremena IKT postala katalizator inovativnosti i kreativnosti a ne tek puko sredstvo za rad.

Za sada je problem objektivno mjerenje postignuća, u tom pogledu su istraživanja zasad manjkava. Druga manjkavost je što je velika većina istraživanja u razvoju CT-a usmjereno na učenje u STEMI području (za primjer vidjeti

[64]), gdje je za očekivati da u startu postoji interes i sposobnosti u brojnim aspektima CT-a.

Da bi se ostvario napredak u razvoju CT aspekata u općoj populaciji potrebno je u što ranijoj dobi edukaciju oblikovati u tom smjeru, s naglaskom na edukaciju samih učitelja. Zahvaljujući velikom broju dostupnih alata koji se temelje na suvremenoj IKT te suvremenim pristupima motivaciji kao što je učenje na temelju igre, za očekivati je veći broj istraživanja a time i napredak u tom dijelu obrazovanja.

LITERATURA

- [1] Did You Know by Karl Fisch; <http://theclosetentrepreneur.com/karl-fischs-did-you-know-presentation-remix> (15.6.2018.)
- [2] Eurostat, <https://ec.europa.eu/eurostat/web/labour-market/job-vacancies/database> (8.11.2018.)
- [3] Atkinson, Robert D. and Mayo, Merrilea Joyce, Refueling the U.S. Innovation Economy: Fresh Approaches to Science, Technology, Engineering and Mathematics (STEM) Education (December 9, 2010). The Information Technology & Innovation Foundation, Forthcoming. Available at SSRN: <https://ssrn.com/abstract=1722822>
- [4] PISA-Programme for International Student Assessment , <http://www.oecd.org/pisa/> (15.6.2018)
- [5] K.Tomljenović; Rezultati anketa budućih učitelja i odgojitelja, neobjavljeni rad
- [6] Computer Science Teachers Association (CSTA) and the International Society for Technology in Education (ISTE), Computational thinking for all, 2011. <https://www.iste.org/explore/articleDetail?articleid=152> (15.6.2018.)
- [7] Wing, J. 2006. Computational Thinking. *Commun. ACM*, 49, 3, 33-35.
- [8] Harel, I., & Papert, S. (Eds.). (1991). *Constructionism*. Westport, CT, US: Ablex Publishing
- [9] Constructivism, [https://en.wikipedia.org/wiki/Constructivism_\(philosophy_of_education\)](https://en.wikipedia.org/wiki/Constructivism_(philosophy_of_education)) (15.6.2018.)
- [10] Lee, T. Y., Mauriello, M. L., Ahn, J., & Bederson, B. B. (2014). CTArcade: Computational thinking with games in school age children. *International Journal of Child-Computer Interaction*, 2(1), 26-33.
- [11] Wing, J. 2010. Computational Thinking: What and Why, <https://www.cs.cmu.edu/link/research-notebook-computational-thinking-what-and-why> (15.6.2018.)
- [12] Riley, D. D., & Hunt, K. A. (2014). *Computational thinking for the modern problem solver*. Boca Raton, FL, USA: CRC Press
- [13] Cuny, J., Snyder, L. and Wing, J. 2010. Demystifying Computational Thinking for Non-Computer Scientists, neobjavljeni rad
- [14] Selby, Cynthia and Woollard, John (2013) Computational thinking: the developing definition, University of Southampton (E-prints) 6pp.
- [15] Allan, V., Barr, V., Brylow, D., & Hambrusch, S. (2010, March). Computational thinking in high school courses. In Proceedings of the 41st ACM technical symposium on computer science education (pp. 390e391). ACM
- [16] Sneider, C., Stephenson, C., Schafer, B., & Flick, L. (2014). Computational thinking in high school science classrooms. *The Science Teacher*, 81(5), 53-59
- [17] Barr, D., Harrison, J., & Conery, L. (2011). Computational thinking: A digital age skill for everyone. *Learning & Leading with Technology*, 38(6), 20-23
- [18] Wing, J. M. (2008). Computational thinking and thinking about computing. *Philosophical Transactions of the Royal Society a: Mathematical, Physical and Engineering Sciences*, 366, 3717-3725.
- [19] Razzouk, R., & Shute, V. (2012). What is design thinking and why is it important? *Review of Educational Research*, 82, 330-348
- [20] 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, 661-670
- [21] Brennan, K., & Resnick, M. (2012, April). New frameworks for studying and assessing the development of computational thinking. In Proceedings of the 2012 annual meeting of the American educational research association (Vancouver, Canada)
- [22] 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, 48-54
- [23] Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., et al. (2016). Defining computational thinking for mathematics and science classrooms. *Journal of Science Education and Technology*, 25(1), 127-147
- [24] Shute, Valerie & Sun, Chen & Asbell-Clarke, Jodi. (2017). Demystifying computational thinking. *Educational Research Review*.
- [25] Resnick, M., Maloney, J., Monroy-Hernandez, A., Rusk, N., Eastmond, E., Brennan, K., et al. (2009). Scratch: Programming for all. *Communications of the ACM*, 52(11), 60-67
- [26] A. L. S. O. de Araujo, W. L. Andrade and D. D. Serey Guerrero, "A systematic mapping study on assessing computational thinking abilities," *2016 IEEE Frontiers in Education Conference (FIE)*, Erie, PA, USA, 2016, pp. 1-9
- [27] Botički, I., Pivalica, D. and Seow, P., 2018, January. The Use of Computational Thinking Concepts in Early Primary School. In *Proceedings of the International Conference of Computational Thinking Education 2018 (CTE2018)*.
- [28] Blockly, <https://developers.google.com/blockly/> (20.10.2018.)
- [29] SC Unplugged <https://csunplugged.org/en/> (15.6.2018.)
- [30] Bell, T., Witten, I. H., & Fellows, M. (2016). CS Unplugged. An enrichment and extension programme for primary-aged students. New Zealand: University of Canterbury. CS Education Research Group Version 3.2.2.
- [31] Bebras International Challenge on Informatics and Computational Thinking; www.bebas.org (30.9.2018.)
- [32] Valentina Dagienė, Gabrielė Stupurienė, and Lina Vinikienė. 2016. Promoting Inclusive Informatics Education Through the Bebras Challenge to All K-12 Students. In *Proceedings of the 17th International Conference on Computer Systems and Technologies 2016 (CompSysTech '16)*, Boris Rachev and Angel Smrikarov (Eds.). ACM, New York, NY, USA, 407-414
- [33] Dabar – međunarodno natjecanje u informatici I logičkom razmišljanju; <http://ucitelji.hr/dabar/> (10.10.2018.)
- [34] Klokani bez granica; <http://www.matematika.hr/klokani/> (10.10.2018.)
- [35] Dagienė V., Futschek G. (2008) Bebras International Contest on Informatics and Computer Literacy: Criteria for Good Tasks. In: Mittermeir R.T., Syslo M.M. (eds) *Informatics Education - Supporting Computational Thinking*. ISSEP 2008. Lecture Notes in Computer Science, vol 5090. Springer, Berlin, Heidelberg
- [36] Dagienė, V., Stupurienė, G., & Vinikienė, L. (2017). Implementation of Dynamic Tasks on Informatics and Computational Thinking. *Baltic Journal of Modern Computing*, 5(3), 306
- [37] Bebras Lodge; <https://bebras.licejus.lt/login?next=> (20.10.2018.)
- [38] Dagienė, V., Stupurienė, G., & Vinikienė, L. (2017, October). Informatics Based Tasks Development in the Bebras Contest Management System. In *International Conference on Information and Software Technologies* (pp. 466-477). Springer, Cham
- [39] Lifelong Kindergarten Group, MIT Media Lab <https://scratch.mit.edu/about> (15.6.2018.)
- [40] App Inventor, <http://appinventor.mit.edu/explore/> (20.10.2018.)
- [41] Lego Mindstorms, <https://www.lego.com/en-us/mindstorms> (25.10.2018.)
- [42] AgentSheet; <http://www.agentsheets.com/> (25.10.2018.)
- [43] Vos, N., van der Meijden, H., & Denessen, E. (2011). Effects of constructing versus playing an educational game on student motivation and deep learning strategy use. *Computers & Education*, 56, 127-137.
- [44] Wilson, A., Hainey, T., & Connolly, T. M. (2013). Using Scratch with primary school children: an evaluation of games constructed to gauge understanding of programming concepts. *International Journal of Game-Based Learning (IJGBL)*, 3(1), 93-109.
- [45] Ministarstvo znanosti I obrazovanja, Kurikulum predmeta Informatika, <https://mzo.hr/sites/default/files/dokumenti/...kurikulumi/informatika-6-3-2018.pdf> (8.11.2018.)

- [46] Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. Basic Books, Inc..
- [47] Tobias, S., Fletcher, J.D. and Wind, A.P., 2014. Game-based learning. In *Handbook of research on educational communications and technology* (pp. 485-503). Springer New York.
- [48] Rode, J.A., Weibert, A., Marshall, A., Aal, K., von Rekowski, T., El Mimouni, H. and Booker, J., 2015, September. From computational thinking to computational making. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing* (pp. 239-250). ACM.
- [49] Papastergiou, M. (2009). Digital game-based learning in high school computer science education: Impact on educational effectiveness and student motivation. *Computers & Education*, 52(1), 1-12.
- [50] Basawapatna, Ashok & Kyu, Han & Koh, Kyu Han & Repenning, Alexander & Webb, David & Sekeres Marshall, Krista. (2011). Recognizing computational thinking patterns. SIGCSE'11 - Proceedings of the 42nd ACM Technical Symposium on Computer Science Education.
- [51] Chang, Yu-hui; Peterson, Lana. Pre-service Teachers' Perceptions of Computational Thinking. *Journal of Technology and Teacher Education*, 2018, 26.3: 353-374
- [52] Barefoot Computing <http://www.barefootcas.org.uk> (20.6.2018.)
- [53] Computing at School <http://www.computingatschool.org.uk> (20.6.2018)
- [54] Teaching London Computing <http://www.teachinglondoncomputing.org> (20.6.2018)
- [55] Computational Thinking for Educators https://computationalthinkingcourse.withgoogle.com/course?use_location=true (20.6.2018)
- [56] Resources <http://cs4hs.media.mit.edu/resources.html> (20.6.2018)
- [57] Lindsey Gouws, Karen Bradshaw, and Peter Wentworth. 2013. First year student performance in a test for computational thinking. In *Proceedings of the South African Institute for Computer Scientists and Information Technologists Conference (SAICSIT '13)*, Philip Machanick and Mosiuoa Tsietsi (Eds.). ACM, New York, NY, USA, 271-277
- [58] Lockwood, J., & Mooney, A. (2017). Computational Thinking in Education: Where does it fit? A systematic literary review. *arXiv preprint arXiv:1703.07659*.
- [59] Lee, T. Y., Mauriello, M. L., Ingraham, J., Sopan, A., Ahn, J., & Bederson, B. B. (2012, May). CTArcade: learning computational thinking while training virtual characters through game play. In *CHI'12 Extended Abstracts on Human Factors in Computing Systems* (pp. 2309-2314). ACM.
- [60] Ministarstvo znanosti I obrazovanja, Kurikulum međupredmetne teme Uporaba IKT, <https://mzo.hr/hr/rubrike/medupredmetne-teme> (8.11.2018.)
- [61] Adler, Rachel F. and Hanna Kim. "Enhancing future K-8 teachers' computational thinking skills through modeling and simulations." *Education and Information Technologies* 23 (2017): 1501-1514.
- [62] Yadav, A., Mayfield, C., Zhou, N., Hambrusch, S., & Korb, J. T. (2014). Computational thinking in elementary and secondary teacher education. *ACM Transactions on Computing Education (TOCE)*, 14(1), 5
- [63] Aman Yadav, Chris Stephenson, and Hai Hong. 2017. Computational thinking for teacher education. *Commun. ACM* 60, 4 (March 2017), 55-62.
- [64] Wu, W. , Hsiao, H. , Wu, P. , Lin, C. and Huang, S. (2012), Investigating the learning- theory foundations of game- based learning: a meta- analysis. *Journal of Computer Assisted Learning*, 28: 265-279
- [65] Koh, K.H., Basawapatna, A., Nickerson, H. and Repenning, A., 2014, July. Real time assessment of computational thinking. In *Visual Languages and Human-Centric Computing (VL/HCC), 2014 IEEE Symposium on* (pp. 49-52). IEEE
- [66] Araujo, A. L. S. O., Santos, J. S., Andrade, W. L., Guerrero, D. D. S., & Dagiene, V. (2017, October). Exploring computational thinking assessment in introductory programming courses. In *2017 IEEE Frontiers in Education Conference (FIE)* (pp. 1-9). IEEE.
- [67] Dr.Scratch, <http://www.drscratch.org/> (20.10.2018.)
- [68] Moreno-León, J., Robles, G. and Román-González, M., 2016, April. Comparing computational thinking development assessment scores with software complexity metrics. In *Global Engineering Education Conference (EDUCON), 2016 IEEE* (pp. 1040-1045). IEEE.
- [69] Dolgopolas, V., Jevsikova, T., Savulionienė, L., & Dagiene, V. (2015). On Evaluation of computational thinking of software engineering novice students. In *Proceedings of the IFIP TC3 Working Conference "A New Culture of Learning: Computing and next Generations* (pp. 90-99).
- [70] Linda Mannila, Valentina Dagiene, Barbara Demo, Natasa Grgurina, Claudio Mirolo, Lennart Rolandsson, and Amber Settle. 2014. Computational Thinking in K-9 Education. In *Proceedings of the Working Group Reports of the 2014 on Innovation & Technology in Computer Science Education Conference (ITiCSE-WGR '14)*, Alison Clear Clear and Raymond Lister (Eds.). ACM, New York, NY, USA, 1-29
- [71] Sengupta, P., Kinnebrew, J.S., Basu, S., Biswas, G., & Clark, D.B. (2012). Integrating computational thinking with K-12 science education using agent-based computation: A theoretical framework. *Education and Information Technologies*, 18, 351-380.