Prospettive Teoriche
Theoretical Perspectives

Quale visione educativa del Computational Thinking? Una prospettiva di ricerca ancora aperta

Which educational vision of Computational Thinking? A still open research perspective

Monica Banzato
Università Ca' Foscari, Venezia
banzato@unive.it

ABSTRACT[1]

Computational thinking has recently been introduced by the Educational Reform of La Buona Scuola, scheduled to begin in primary schools (diffusion has already occurred in most European countries and others in the developed world). The addition of this new competence is considered a literacy essential which every child has to learn. It is understood not so much as a technical skill, but as logical thinking across all the disciplines. The problems to be addressed are: What is the definition of computational thinking? Why has computational thinking now assumed a central role in education from primary school through university? Which educational vision of computational thinking for children in their early years of development emerges and is shared by the scientific community of Computer Science? What are the cognitive and educational implications for computational thinking? This work aims to investigate the state of art on computational thinking in education for primary and lower secondary education, examining its spread in Europe and the recent academic literature, with the aim of identifying the critical issues in the research field and the priorities for implementation and future investigation.

Il pensiero computazionale è stato recentemente introdotto dalla Riforma Educativa della Buona Scuola, prevista a partire dalla scuola primaria (diffusione avvenuta nella gran parte dei Paesi europei e in altri Paesi del mondo). L'aggiunta di questa nuova competenza è ritenuta una literacy fondamentale che ogni bambino deve apprendere in quanto viene intesa non tanto come abilità tecnica, ma come pensiero logico trasversale a tutte le discipline. I problemi da affrontare sono: Qual è la definizione di pensiero computazionale? Perché il pensiero computazionale ha assunto ora un ruolo fondamentale in campo educativo su tutta la filiera formativa? Quale visione educativa per la primaria e secondaria inferiore del pensiero computazionale emerge e condivide la comunità scientifica del Computer Science? Quali sono le implicazioni educative e cognitive del pensiero computazionale? Questo lavoro è teso a indagare lo stato dell'arte sul pensiero computazionale in materia di istruzione per la scuola primaria e secondaria di primo grado, esaminando la sua diffusione in Europa e la recente letteratura accademica, con l'obiettivo di identificare le criticità e di individuare le priorità di attuazione e per le indagini future.

KEYWORDS

Computational Thinking; Coding; Primary Education; Cognitive and Educational Implications; Educational Research on Computational Thinking.
Pensiero computazionale; Coding; Istruzione Primaria; Implicazioni cognitive ed educative; Ricerca educativa su pensiero computazionale.

Formazione & Insegnamento XIV – 2 – 2016
ISSN 1973-4778 print – 2279-7505 on line
doi: 107346/-fei-XIV-02-16_02

1. Introduzione

> *Computational Thinking is the thought processes involved in formulating problems and their solutions so that the solutions are represented in a form that can be effectively carried out by an information-processing agent"* (Cuny, Snyder, & Wing, 2010).
> *When I use the term computational thinking, my interpretation of the words 'problem' and 'solution' is broad; in particular, I mean not just mathematically well-defined problems whose solutions are completely analyzable, e.g., a proof, an algorithm, or a program, but also real-world problems whose solutions might be in the form of large, complex software systems. Thus, computational thinking overlaps with logical thinking and systems thinking. It includes algorithmic thinking and parallel thinking, which in turn engage other kinds of thought processes, e.g., compositional reasoning, pattern matching, procedural thinking, and recursive thinking* (Wing, 2011).
> ***Computer programming*** *(when referring to software) and* ***coding*** *are used interchangeably and refer to more or less the same activities of writing the instructions (recipe) for the computer to perform a specific task following a logic. However, based on the definitions above, coding can also be seen as a specific subtask of software computer programming which arranges the implementation of the algorithm in the target programming language* (Schoolnet, 2014).

L'introduzione del pensiero computazionale (*computational thinking*) e del *coding* nella scuola primaria e secondaria di primo grado è una delle novità introdotte sia dal documento della Buona Scuola (2014), sia dalla Riforma (2015). I due termini (che si rincorrono nei due documenti) sono diversi (al momento si fa riferimento alle citazioni dell'articolo in apertura. Successivamente saranno ripresi e ulteriormente sviluppati). Nel documento della Buona Scuola (2014), alcune volte troviamo *coding*, altre volte pensiero computazionale. L'indice fa un chiaro riferimento solo al *coding*, riportato al punto 4.2 che recita: "*La prossima alfabetizzazione: lingue straniere, coding, economia*". A pagina 42, il termine *coding* appare ancora da solo come segue: "*Serve quindi un piano nazionale che consenta di introdurre il coding (la programmazione) nella scuola italiana. A partire dalla primaria: vogliamo che nei prossimi tre anni in ogni classe gli alunni imparino a risolvere problemi complessi applicando la logica del paradigma informatico anche attraverso modalità ludiche (gamification)*". A pagina 131, quando il documento della Buona Scuola (2014) presenta il suo tentativo di sintesi delle maggiori novità della prossima riforma, ritroviamo di nuovo, al punto 10, il *coding* insieme al termine pensiero computazionale, come segue: "*10. Le nuove alfabetizzazioni. Rafforzamento del piano formativo per le lingue straniere, a partire dai 6 anni. Competenze digitali: coding e pensiero computazionale nella primaria e piano "Digital Makers" nella secondaria*". Per la prima volta compare il *coding* e il pensiero computazionale insieme, senza presentare una definizione o una spiegazione della differenza tra i due termini. Il testo fornisce solo un accenno di uso di sinonimo per *coding*, come abbiamo visto a pagina 42: "*coding (la programmazione)*". Ma l'uso di un termine come "programmazione" come sinonimo potrebbe in questo caso forviare, e comunque non è una definizione. Invece, non appare emergere una definizione per il pensiero computazionale.

Nella riforma LEGGE 13 luglio 2015, n. 107. *Riforma del sistema nazionale di istruzione e formazione e delega per il riordino delle disposizioni legislative vigenti*. Riportata in Gazzetta Ufficiale (15/07/2015 Serie generale - n. 162), il termine *coding* scompare, mentre il pensiero computazionale appare a pagina 2, comma 7, punto h) il testo recita: "h) *sviluppo delle competenze digitali degli studenti, con particolare riguardo al pensiero computazionale, all'utilizzo critico e con-*

sapevole dei social network e dei media nonché alla produzione e ai legami con il mondo del lavoro".

I testi ministeriali non hanno l'obiettivo di definire i termini in quanto rispondono ad una funzione legislativa che invece il documento della "La Buona Scuola" avrebbe potuto rispondere a questa esigenza. Comunque, si possono evincere le difficoltà di chi ha steso il documento di riforma in quanto, come vedremo, la stessa comunità scientifica del *Computer Science* a livello internazionale non è giunta a una definizione condivisa del pensiero computazionale (*computational thinking*). "*Although computational thinking has received considerable attention over the past several years, there is little agreement on what a definition for computational thinking might encompass (Allan et al., 2010; Barr & Stephenson, 2011; National Academies of Science, 2010)*" (Brennan, Resnick, 2012).

L'insegnamento del pensiero computazionale/coding nella primaria e secondaria di primo grado appare quindi tra le novità sostanziali e sembra essere uno spazio di lavoro tutto nuovo da costruire (e tra l'altro "in corsa", dato che dovrà andare a regime nei prossimi tre anni). Il problema educativo è: che cosa (pensiero computazionale, programmazione o *coding*? O entrambe?) e come.

La sfida da affrontare a livello educativo appare senza precedenti (fin ora tale settore era considerato una nicchia per appassionati sia tra docenti, sia tra studenti), in quanto il pensiero computazionale (riferimento alle definizioni all'inizio di paragrafo) appare promuovere uno spazio educativo di cruciale importanza che può avere degli sviluppi (1) non solo per l'area scientifica in sé (sviluppo di competenze di programmazione), (2) ma come pensiero trasversale a tutte le aree promuovendo il pensiero logico, di *problem solving* e *decision making* e strategie di pensiero critico (Wing, 2011). Come afferma Werner et al. (2005: p.303), citando NRC, 1999: "*IT fluency should not be regarded as an end state that is independent of domain, but rather as something that develops over a lifetime in particular domains of interest and that has a different character and tome depending on which domains are involved*". Kolodner è ancora più incisiva su questo punto: "*computational thinking is a set of skills that transfer across disciplinary domains*" (CNR, 2011, p. 54). O come asserisce Wing (2006*): "For everyone, everywhere. Computational thinking will be a reality when it is so integral to human endeavors it disappears as an explicit philosophy*". O come recita Resnick, in un suo famoso discorso su TED (2012): "[...] *children aren't just learning to code, they are coding to learn*". Ci sono molte esperienze di ricerca e si sono concentrate fin ora soprattutto in un contesto di scuola secondaria di secondo grado, mentre per le scuole primarie e secondarie di primo grado, esiste un corpo abbastanza consistente di sperimentazioni sull'insegnamento soprattutto del *coding* (tuttavia circoscritte a interventi medio brevi e non su piani temporali annuali o pluriennali), ma non ancora tante sul pensiero computazionale. Secondo Grover et al. (2013) e Wing (2011) questa area di ricerca avrebbe bisogno di maggiori attenzioni e "*much remains to be done to help develop a more lucid theoretical and practical understanding of computational thinking in children*" (Grover et al., 2013).

Il presente lavoro si concentrerà sui seguenti problemi: Qual è la definizione di pensiero computazionale? Perché il pensiero computazionale ha assunto ora un ruolo fondamentale in campo educativo su tutta la filiera formativa? Quale visione educativa del pensiero computazionale emerge e condivide la comunità scientifica del Computer Science in ambito di scolastico per gli studenti in fase di sviluppo (scuola primaria e secondaria di primo grado)? Quali sono le implicazioni educative e cognitive del pensiero computazionale? Questo lavoro è teso a indagare lo stato dell'arte del discorso sul pensiero computazionale in materia di istruzione per la scuola primaria e secondaria di primo grado, esaminando la sua diffusione in Europa e la recente letteratura accademica, con l'obietti-

vo di identificare le criticità nel campo della ricerca e di individuare le priorità di attuazione e piste per le indagini future.

2. Diffusione pensiero computazionale, programmazione e coding in Europa

Dai risultati dell'indagine esplorativa del report dell'European Schoonet pubblicato nel 2014 dal titolo: *Computing our future Computer programming and coding - Priorities, school curricula and initiatives across Europe,* appare che gli obiettivi presentati dell'Agenda Digitale (EU, 2010) siano stati immediatamente condivisi e messi in atto dai Ministeri dell'Istruzione dei diversi Paesi europei, inserendo il pensiero computazionale/*coding* tra gli insegnamenti curriculari, sia a livello obbligatorio o facoltativo (a seconda dei livelli di scuola). Dove è stato attuato un programma di riforma o di aggiornamento del sistema educativo, il *Computational Tinking* (da ora in poi CT) viene considerato in generale una competenza essenziale per lo sviluppo della *literacy* del 21° secolo.

Marianne Thyssen, la commissaria europea responsabile per l'Occupazione, gli affari sociali, competenze e la mobilità del lavoro afferma che "oggi, il 90% di tutti i posti di lavoro richiedono almeno un livello base di competenze nelle TIC e queste ora includono il *coding*" (Jacobsen, 2015).

Il rapporto fornisce una panoramica delle iniziative di *coding/programming /computational thinking* in tutta Europa e appare riportare uno stato dell'arte davvero positivo almeno sulla sua diffusione nei vari Paesi. In fase preliminare, il rapporto cerca di disambiguare i diversi termini utilizzati nei vari Paesi, anche se vedremo non appare semplice. Per *computer programming* si intende "*the process of developing and implementing various sets of instructions to enable a computer to perform a certain task, solve problems, and provide human interactivity. These instructions (source codes which are written in a programming language) are considered computer programs and help the computer to operate smoothly*".

In questo report, i termini *computing* e *coding* vengono usati in modo intercambiabile e si riferiscono ad attività che permettono ai bambini non solo di imparare come utilizzare i programmi specifici, ma a imparare a programmare computer, tablet o altri dispositivi elettronici.

Programmazione è il termine comunemente usato in Norvegia, Estonia, Finlandia, Spagna, Grecia, Danimarca, Cipro, Bulgaria, Belgio (Fiandre), Turchia. Alcuni Paesi come la Lituania e la Polonia preferiscono utilizzare un termine più ampio come la programmazione, invece di *coding*. Molti Paesi hanno messo l'accento sul pensiero computazionale / pensiero algoritmico come Spagna, Cipro, Bulgaria, Belgio (Fiandre) e Portogallo. L'Irlanda è l'unico Paese che si riferisce esclusivamente al termine *coding*.

Sugli aspetti educativi, il report si limita a dichiarare: "*La scelta dei termini da parte dei Paesi riflette un po' la filosofia che ispira il tipo di abilità o competenze da sviluppare*" (Scholnet, 2014).

Dal momento che la prima relazione è stata pubblicata nel 2014, nel frattempo nuovi Paesi hanno aderito a questa tendenza, come la Spagna e la Francia. Repubblica Ceca, Polonia, Lituania e Malta non hanno reso completamente disponibili i dati, ma dal rapporto si evince che esprimono l'intenzione di integrare i programmi educativi in tempi brevi.

Dei 20 Paesi che hanno partecipato l'indagine, 12 hanno dichiarato che il *coding* e il *programming* fanno parte dei curricula come: Bulgaria, Cipro, Repubblica Ceca, Danimarca, Estonia, Grecia, Irlanda, Italia, Lituania, Polonia, Portogallo e Regno Unito. Altri 7 paesi (come Belgio Fiandre, Spagna, Finlandia, Francia, Lussemburgo, Olanda e Turchia), dichiarano che è già in agenda nel prossimo futu-

ro. L'obiettivo di questi Paesi non è solo insegnamento del CT/programmazione/coding fine a se stesso (inteso come programmazione), ma soprattutto promuovere il pensiero logico e il problem solving, oltre a incentivare i corsi di studio nel settore delle scienze dell'informazione nella formazione terziaria per promuovere l'occupabilità.

In tutto, 15 paesi dell'UE hanno integrato nel curriculum il CT/programmazione/*coding*, sia a livello nazionale e regionale e sia a livello locale come: Austria, Bulgaria, Repubblica Ceca, Danimarca, Estonia, Francia, Ungheria, Irlanda, Lituania, Malta, Spagna, Polonia, Portogallo, Slovacchia e Regno Unito.

Il *coding* entrerà a far parte della formazione di base in Finlandia entro il 2016, mentre la regione delle Fiandre in Belgio attualmente stanno discutendo la questione.

L'insegnamento del *coding/programming/computational thinking* è già inserito a livello di scuola primaria in Estonia, Francia, Spagna, Slovacchia, Italia e Inghilterra. Mentre nelle Fiandre, Finlandia, Polonia e Portogallo è in fase di attuazione nell'anno corrente.

Il coding è anche una parte obbligatoria del programma per specifici livelli di istruzione, soprattutto nel quadro di corsi di informatica in Bulgaria, Repubblica Ceca, Danimarca, Portogallo, Slovacchia, Spagna, e Regno Unito. In Danimarca, la conoscenza della programmazione di base è una parte obbligatoria dei curriculum di fisica, chimica e matematica. La Slovacchia va oltre in quanto ha previsto l'integrazione del *coding* su tutta la filiera formativa (ogni ordine e grado) come materia obbligatoria.

Dei 20 Paesi esaminati, 11 dichiarano che l'obiettivo esplicito è di attrarre più studenti a scienze informatiche, mentre 8 Paesi dichiarano che hanno anche l'obiettivo di promuovere l'occupabilità nel settore.

Altri Paesi nel mondo appaiono coinvolti in questo rinnovo del curriculum, come: Israele, Nuova Zelanda, Australia, Sud Corea, India (Jones, 2011).

3. Definizioni e analisi della letteratura

Le domande che hanno guidato l'analisi delle definizioni e della letteratura sono le seguenti: quali sono i significati attribuiti al pensiero computazionale? Come mai solo ora il pensiero computazionale è diventato così importante tanto che questa disciplina è entrata in quasi tutti i curriculum delle scuole primarie e secondarie di primo grado di molti Paesi a livello globale? Esiste una definizione condivisa di pensiero computazionale? Nei diversi Paesi l'introduzione del CT è stata anticipata da una discussione allargata a livello educativo, pedagogico e formativo? C'è una visione educativa sull'introduzione del pensiero computazionale lungo la filiera formativa?

Quando ci si addentra nella letteratura di ricerca soprattutto degli ultimi dieci anni, ci si accorge subito che la discussione seppure sfaccettata e complessa, è rimasta in generale all'interno dell'ambito del Computer Science e quindi l'ambito accademico è disciplinare. La discussione sugli aspetti educativi e pedagogici appare ancora poco condivisa a livello interdisciplinare. Anche se, come vedremo, ci sono stati dei tentativi di promuovere e condividere la questione, come i due workshop organizzati dal *National Research Council* (NRC) di cui, il primo rivolto a discutere la definizione di pensiero computazionale (NRC, 2011) e il secondo ha avuto l'obiettivo di indagare "*the nature of computational thinking and its cognitive and educational implications*" (National Research Council [NRC], 2011, p. viii).

Comunque, l'interesse sulla visione educativa del pensiero computazione sembra che sia partito da un articolo breve e influente scritto da Jeannette Wing

"*Computational Thinking*", apparso nella sezione Punto di vista (*Viewpoint section*) del numero di marzo 2006 del *Communications of the ACM* (*Association for Computing Machinery*) con il seguente pronunciamento: "[The Computational Thinking] *represents a universally applicable attitude and skill set everyone, not just computer scientists, would be eager to learn and use*" (p. 33).

Le argomentazioni di Wing hanno catturato l'attenzione della vasta comunità accademica, ma anche quella politica (fatto mai accaduto prima). Sotto la spinta del suo articolo e supportata da una crescente comunità di ricercatori, educatori, il pensiero computazionale è diventato una "questione educativa cruciale", oltre a un "programma educativo" da mettere tra le priorità dell'agenda politica (negli Stati Uniti il pensiero computazionale è citato nei documenti sui curricula scolastici nel 2012; in Gran Bretagna è entrato nel curriculum nel 2013; in Estonia, nel 2013, etc.).

La Wing (2006) propone la seguente definizione di pensiero computazionale:

- "*Si riferisce a concettualizzare e non a programmare*. Informatica non significa programmazione dei computer. Pensare come un informatico significa molto più che esser capaci a programmare con il computer e richiede soprattutto **saper pensare a livelli multipli di astrazione.**
- *Si riferisce a competenze fondamentali*, **non capacità meccaniche** di basso livello. Una abilità fondamentale è qualcosa di cui ciascuno deve disporre per operare nella società moderna.
- È *un modo di pensare degli uomini, non dei computer*. Si riferisce cioè a una modalità in cui gli uomini possono **risolvere i problemi**, non è un tentativo di fare in modo che gli uomini pensino come i computer; i computer sono intelligenti e noiosi mentre gli uomini hanno intelligenza e fantasia.
- **Punta più sulle idee** che sugli specifici artefatti. Non sono gli artefatti che produciamo che possono esser presenti ovunque e influenzare le nostre vite. Sono invece i concetti computazionali che utilizziamo per **risolvere i problemi**, per gestire le nostre vite quotidiane e per comunicare e interagire con gli altri.
- È per tutti ed è applicabile ovunque. Il pensiero computazionale diventerà una realtà quando sarà così integrato con l'attività umana da cessare di essere una filosofia esplicita" (Wing, 2006).

Appare emergere con chiarezza l'importanza del pensiero computazionale, come una nuova literacy non solo di natura "tecnica", ma una literacy di "pensiero trasversale e critico" che punta a sviluppare capacità di ragionamento, astrazione, risoluzione dei problemi, rappresentazione e generalizzazione. L'idea comunque non è nuova come sottolineato da diversi esperti (Olimpo, 2015). Nel 1960, Alan Perlis ha sostenuto la necessità per gli studenti universitari di tutte le discipline dovessero imparare a programmare e conoscere la "teoria della computazione" (Guzdial, 2008). Tuttavia, nel contesto della formazione primaria, la prima volta che l'informatica ha guadagnato attenzione "popolare" è stato grazie a tutto il lavoro del MIT e soprattutto di Seymour Papert nel 1980. Papert è il pioniere dell'idea che anche nei bambini in fase di sviluppo possono imparare il pensiero computazionale attraverso la programmazione LOGO (Papert, 1980 1991).

Al di là del fatto che l'idea sia nuova o meno, questa recente attenzione sul tema assume una prospettiva nuova nel 21° secolo rispetto al passato, e l'articolo di Wing del 2006 costituisce un punto di partenza.

4. Quali implicazioni cognitive e educative del pensiero computazionale?

Secondo Grover e Pea, nel loro articolo *Computational Thinking in K–12: A Review of the State of the Field* (2013), ritengono che ci sia stata una convergenza positiva di intenti sia in ambito accademico, sia a livello politico che sono stati il trampolino di lancio per l'inserimento del pensiero computazionale nel curriculum scolastico. Il processo è stato talmente rapido che tutt'ora questa introduzione del pensiero computazionale o del *coding* (vedi sessione 2), seppure necessaria e importante nella società ad economia digitale, è afflitta da due problemi cruciali: il primo, non esiste ancora definizione condivisa di CT tra gli esperti del settore; secondo, non è ancora definita quale sia la visione educativa del pensiero computazionale.

Wing è stata capace di catalizzare l'attenzione sul pensiero computazionale ed è servita (come accennato nel paragrafo 3) come punto di partenza per due workshops del *National Academy of Sciences*: il primo si è concentrato sulla definizione del pensiero computazionale e il secondo ha avuto l'obiettivo di esplorare "la natura del pensiero computazionale e le sue implicazioni cognitive ed educative" (National Research Council [NRC], 2011, p. viii). In particolar il secondo workshop sembra essere più interessante per entrare in merito a come viene discussa la relazione tra pensiero computazionale ed educazione, non tanto perché i partecipanti siano arrivati a delle soluzioni su questo preciso obiettivo (motivo stesso del workshop), anzi il contrario (gli stessi esperti che hanno partecipato ne hanno lamentato le mancate ricadute). Ma seguire la discussione del workshop consente di entrare in merito agli stessi approcci degli autorevoli esponenti che hanno partecipato e come questi affrontano le questioni educative. Dalle discussioni emergono interessanti elementi da cui partire che potrebbero essere condivisi anche qui in Italia, in una discussione allargata con un approccio interdisciplinare.

5. Quale visione educativa del pensiero computazionale?

A questa domanda appare ancora prematuro rispondere, tanto più nel workshop del 2011 (NRC, 2011). Dal workshop, emerge che molti esperti sostengono che la comunità scientifica non sia stata ancora in grado di definire esattamente che cosa si intenda con pensiero computazionale e tale passo è fondamentale, in quanto consentirebbe di orientarsi su quali programmi di ricerca educativi investire le energie. In generale, molti contributi in questo workshop appaiono molto interessanti, ma quasi sempre orientati alla disciplina in sé piuttosto che sugli aspetti cognitivi, educativi e di apprendimento, pur presentando molte ricerche educative nel settore e esperimenti formativi di successo.

Tra i diversi contributi, l'intervento della Kolodner (le sue ricerche si concentrano nell'area delle scienze cognitive e delle tecnologie educative) appare tra coloro che abbiano affrontato in modo diretto la questione (senza posizionarsi tra coloro che definiscono o coloro che presentano esperienze/sperimentazioni educative), puntando il dito sulla criticità del simposio, ovvero sulla faticosa convergenza tra definizioni e aspetti educativi e cercando arrivare a una proposta.

Secondo la Kolodner, partire da più definizioni è un buon inizio per la discussione, ma non è abbastanza per poter decidere quale tipo di approccio/strategie cognitive, quale visione educativa e come affrontare i processi di apprendimento/insegnamento del pensiero computazionale con bambini in fase di sviluppo. Kolodner afferma che ritiene il pensiero computazionale "*un set di competenze trasversali tra i domini disciplinari*" (NRC, 2011, p. 54). Siamo chiaramente agli esordi ma questa sarebbe una buona occasione per ripensare alla *literacy* di base.

Il merito del contributo della Kolodner è di aver tentato di portare a sintesi le definizioni emergenti nel simposio e farle convergere sulle questioni dell'apprendimento/insegnamento e dell'educazione, rispetto agli altri interventi in generale molto orientati e preoccupati ai contenuti in sé del pensiero computazionale. A suo avviso si possono sintetizzare le diverse definizione in due posizioni, che sono utili anche per noi per un ragionamento successivo.

1. Nel gruppo della prima definizione ci sono esperti come Tinker e Edelson (citati dalla stessa Kolodner), secondo i quali il pensiero computazionale consiste nel decomporre i problemi in parti più piccole che sono risolvibili da dispositivi computazionali: "[…]*computational thinking as fundamentally about breaking problems into smaller and smaller problems that are solvable by rather simplistic computational devices*" (p. 53). Secondo la Kolodner, questa definizione è condivisibile ma limita il pensiero computazionale all'ambito della disciplina in sé. In effetti, questo passaggio della Kolodner mira anche qualcosa di più evidentemente, in particolare agli aspetti di trasversalità e ad interpretare allo stesso modo l'apprendimento del problem solving non collegato necessariamente a un ambito disciplinare. (Un problema esiste quanto lo stato corrente delle cose non è quello desiderato, ovvero lo stato obiettivo. La risoluzione di un problema consiste nella trasformazione dello stato corrente nello stato obiettivo attraverso metodi soddisfacenti. Ma l'aspetto ancora più importante: la nostra capacità di risolvere un problema è influenzata dalla nostra rappresentazione del problema stesso. Le teorie dell'apprendimento spiegano che ogni nostra azione corretta o non corretta nei confronti di un problema parte sempre dalla rappresentazione che abbiamo del problema stesso).
2. L'altro approccio al pensiero computazionale che rappresenta un passo in avanti rispetto alla posizione precedente è quello proposto da Mitch Resnick. Secondo Resnick, il pensiero computazionale non è solo limitato alla soluzione dei problemi in ambito scientifico, ma significa esprimere se stessi utilizzando il calcolo in modo fluente in tutti gli ambiti di studio, lavoro e in ultimo della vita. Per Resnick, il pensiero computazionale è permettere a tutti (e non solo quelli che sono buoni risolutori di problemi), di esprimersi attraverso una varietà di mezzi di comunicazione. In questa prospettiva pedagogicamente inclusiva dei diversi talenti, motivazioni e interessi degli studenti (non si può immaginare una scuola primaria dove si formano solo dei futuri scienziati del Computer Science), il pensiero computazionale significa essere in grado di creare, costruire, inventare presentazioni e rappresentazioni con e attraverso il calcolo (ma non esclusivamente). Ciò richiede fluidità con i media computazionali. L'approccio di Resnick è quello che ha una tensione verso gli aspetti educativi, pedagogici, formativi in modo esplicito, oltre a presentare un approccio inclusivo, nel senso che tiene conto delle diversità cognitive, emotive e relazionali in una classe. La definizione di Resnick appare essere diversa rispetto alla posizione precedente, non solo per la definizione in sé, ma anche per il background delle esperienze e sperimentazioni educative da cui deriva, avendo una lunga storia partita dalle sperimentazione educative di Papert (Resnick è stato infatti allievo di Papert). L'approccio educativo all'apprendimento del bambino è di matrice costruttivista. Infatti, anche la soluzione educativa e didattica di Resnick (una delle poche, forse l'unica, presenti nel report che in qualche modo apporta il suo contributo alla questione educativa del workshop e che tiene conto dello sviluppo cognitivo del bambino) è basata sull'apprendimento situato ed *embedded*. L'approccio educativo di Resnick appare sì più largo, ma comprende anche l'approccio precedente (ovvero la definizione di CT) senza escludere che il pensiero computazionale consista nella soluzioni di problemi (a diversi livelli di complessità) e per contribuire a rompere problemi a pezzi e comporre i pezzi in modo nuovo.

La Kolodner ritiene che sia necessario arrivare a una sintesi delle due posizioni ma nel suo tentativo si sente che rimane a metà del guado:

> *Computational thinking is a kind of reasoning in which one breaks problems/goals/challenges into smaller pieces that are doable by a stupid computational device. This, in general, means thinking in terms of functions that need to be carried out to achieve a goal or solve a problem (not functions in the mathematical sense, but rather in terms of how things work) and pulling apart those problems/goals/challenges into smaller pieces that are functionally separate from each other and where the functions that are pulled out tend to repeat over many different situations. Computational thinkers tend to break problems into functional pieces that have meaning beyond the particular situation in which they are being used. These functional pieces can then be called on repeatedly in solving the problem or combined in new ways to solve new problems and achieve new goals and challenges.*

Il tentativo di ridurre la tensione tra le due posizioni da parte della Kolodner ha il merito di continuare lo sforzo di affrontare la questione, anche se sembra concentrarsi soprattutto sulla prima posizione, aggiungendo altre abilità cognitive al computational thinking, ma l'aspetto educativo appare rarefarsi nella ridefinizione.

Risulta un po' difficile mettere in relazione le due posizioni senza trovare ostacoli (ma la sfida è affascinate) in quanto queste appaiono definire il pensiero computazionale, partendo a monte da presupposti diversi. La prima posizione appare esprimere una definizione di contenuto che in qualche modo potrebbe essere utile come punto di partenza, ma senza arrivare al dunque sugli aspetti educativi; e soprattutto, non parte da un riferimento pedagogico/educativo o da una teoria dell'apprendimento (proprio non c'è). Il gap pedagogico educativo della prima posizione è evidente rispetto alla seconda posizione (quella presentata da Resnick) che invece nella sua definizione (che è sempre epistemologica) ma deriva e parte da una solida tradizione di ricerca educativa e di apprendimento (costruzionismo, definito da Papert, partendo dagli studi di J. Piaget sul costruttivismo). Un evidente vantaggio rispetto agli altri per l'obiettivo posto dalla comunità scientifica del simposio tesa a trovare non solo una definizione, ma una visione educativa. Tuttavia la posizione di Resnick appare dominante in quanto non può confrontarsi con un'altra. Infatti, come sottolineano Grover e Pea (2013):

> *Barring some recent studies, such as Fadjo, Lu, and Black (2009) and Berland and Lee (2011), few others have taken into account contemporary research in the learning sciences in socio-cultural and situated learning, distributed and embodied cognition, as well as activity, interaction and discourse analyses.*

6. Conclusioni aperte

Bruner afferma:

> *Finally, a theory of instruction seeks to take account of the fact that a curriculum reflects not only the nature of knowledge itself but also the nature of the knower and the knowledge-getting process. A curriculum is the enterprise par excellence where the line between subject matter and the method grows necessarily indistinct. A body of knowledge, enshrined in a university faculty and embodied in a series of authoritative volumes, is the result of much prior intellectual activity. To instruct someone in these disciplines is not a matter of getting him to commit results to mind. Rather, it is to teach him to participate in the process that makes possible the establish-*

> *ment of knowledge. We teach a subject not to produce little living libraries on the subject but, rather, to get a student to think mathematically for himself, to consider matters as historian does, to embody the process of knowledge-getting. Knowing is a process, not a product* (Bruner, 1970, p. 72).

Questo pensiero di Bruner pubblicato quasi cinquanta anni fa, appare ancora valido e non ha perso assolutamente la sua incisività, da quanto sembra emergere dallo stesso simposio.

Sul piano della ricerca educativa si apre un terreno davvero interessante. L'introduzione del pensiero computazionale, come area trasversale di *literacy* e di competenze, richiede necessariamente uno sviluppo e un confronto interdisciplinare di ricerca. Un progetto di visione educativa comune e condivisa gioverebbe a rilanciare un confronto proficuo e di collaborazione, dove l'accento non sia solo nella definizione del contenuto (chiaramente necessario) se non relazione alla conoscenza come processo, tenendo conto di tutte le dimensioni educative: cognitivo, sociale ed emotivo e contestuale degli studenti in fase di sviluppo.

Anche in Europa e in Italia, permangono le problematiche che sono state (parzialmente) presentate in questo lavoro (Banzato & Tosato, 2016; Banzato e Tosato, 2017), come ad esempio: quale visione educativa adottare per introdurre il pensiero computazionale nella scuola? Quali modelli educativi e pedagogici per l'insegnamento e apprendimento del *coding/programming/computational thinking*? Modelli che si basano su quali strategie di insegnamento e apprendimento? Quale tipo di formazione iniziale per gli insegnanti della primaria e secondaria di primo grado? Quale tipo di formazione è prevista per gli insegnanti in servizio sul pensiero computazionale e sul coding o su entrambe? La formazione del pensiero computazionale è prevista solo per gli insegnanti di materie scientifiche o anche umanistiche? Altro problema, quale tipo di aggiornamento? Come insegnare il pensiero computazionale e il coding per gli studenti in fase di sviluppo cognitivo? Quale progettazione del curriculum del pensiero computazionale e/o coding lungo tutta la filiera formativa? Rimangono altre questioni in sospeso come: che cosa fare e come includere i BES (ovvero i soggetti disabili, o con disturbi di apprendimento o con disagi)? Di converso, anche agli studenti più dotati dovrebbero essere sostenuti a perseguire una carriera nel campo della scienza computazionale. Al momento, molti concorsi in ambito scolastico (esempio sul coding, robotica, etc.) sono offerti in diversi Paesi europei come un modo per premiare gli studenti che eccellono con grande talento in questo settore, ma una particolare attenzione dovrebbe essere rivolta anche agli studenti con un interesse generale e dovrebbero essere altrettanto sostenuti con pari opportunità di espressione e creatività. Il gap gender è una questione di cruciale importanza e comprendere quali siano le azioni che incentivano il coinvolgimento delle ragazze allo studio di questa nuova literacy. Con l'introduzione di questa materia nel curriculum, sarebbe interessante indagare e monitorare quali siano i diversi fattori che incidono su questa introduzione del pensiero computazionale/*coding*, attraverso studi longitudinali a medio e lungo periodo per valutarne le ricadute. Qui solo per citare alcuni problemi, senza la pretesa di essere esaustivi. Visti i tempi di attuazione entro i prossimi tre anni, non si può ignorare che le singole scuole in Italia si stanno già muovendo da sole[1] ad attivare esperienze, attività e la-

1 In rete si possono trovare diversi corsi gratuiti di *coding*, come quelli offerti da istituzioni accademiche (esempio, Coursera [https://www.coursera.org/], Open University), o alla Comunità Europea (European Coding Initiative [www.allyouneediscode.eu/.] o da organizzazioni e associazioni indipendenti (esempio CODE [https://code.org/], Code Club [www.codeclub.org.uk/]), CAS - Computing at School [www.computingatschool.org.uk] e commerciali (es. da Google o CodeAcademy).

boratori in questo settore, basandosi sull'aiuto o di docenti interni o di formatori esterni che hanno competenze in coding, programmazione e robotica educativa. "Dietro ogni problema c'è un'opportunità" (Galileo Galilei).

Nota: Il presente lavoro è una parte della presentazione dell'intervento tenutosi alla SIREF Summer School, *I futuri della scuola e la ricerca pedagogica*, Catania, 5 7-8-9 Settembre 2015. La selezione bibliografia (normative, articoli scientifici, volumi) fa riferimento fino alla data della medesima conferenza.

Riferimenti bibliografici

Banzato, M., & Tosato, P. (2017). An Exploratory Study of the Impact of Self-Efficacy and Learning Engagement in Coding Learning Activities in Italian Middle School. *AACE - International Journal on E-Learning* (IJEL) 17 (1) (In corso di pubblicazione).

Banzato, M., & Tosato P. (2016). Self-efficacy degli insegnanti in attività di coding: uno studio di caso nella primaria e secondaria di primo grado. P., Limone & D., Parmigiani (eds.), *Atti del convegno Sirem 2016. L'educazione digitale*. (In corso di pubblicazione).

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*.

Bruner, J. S. (1970). Some theorems on instruction in reading.'. *Educational Psychology. London: Methuen*.

Chioccariello, A. (2013). Il pensiero computazionale. In V. Midoro e D. Persico (a cura di), *La pedagogia nell'era digitale*. Pescara: Menabò.

Cuny, J., Snyder, L., & Wing, J.M. (2010). *Demystifying computational thinking for noncomputer scientists*. Unpublished manuscript in progress, referenced in http://www.cs.cmu.edu/~CompThink/resources/TheLinkWing.pdf

EU (2010). *Digital Agenda for Europe: Communication from the Commission* (26/08/2010). Eur-Lex, Available: http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:52010DC0245R(01):EN:NOT

European Commission (2015, July 20). *Digital Agenda for Europe, Coding - the 21st century skill* [online]. Available: http://ec.europa.eu/digital-agenda/en/coding-21st-century-skill

Grover, S., & Pea, R. (2013). Computational Thinking in K–12 A Review of the State of the Field. *Educational Researcher*, *42*(1), 38-43.

Guzdial, M., Ericson, B. J., McKlin, T., & Engelman, S. (2012, September). A statewide survey on computing education pathways and influences: factors in broadening participation in computing. In *Proceedings of the ninth annual international conference on International computing education research* (pp. 143-150). ACM.

Jacobsen, H. (2015). Coding classes trending across EU schools. In EU code week 2015. Aivaible: http://www.euractiv.com/sections/eu-code-week-2015/coding-classes-trending-across-eu-schools-318374

Jones, S. P. (2011). Computing at School. International comparisons. *Retrieved May*, *7*, 2013.

MIUR (2014, July 25). La Buona Scuola. Facciamo crescere il Paese [online]. Available: https://labuonascuola.gov.it/documenti/La%20Buona%20Scuola.pdf?v=d0f805a

MIUR (2015). *Riforma del sistema nazionale di istruzione e formazione e delega per il riordino delle disposizioni legislative vigenti*. Gazzetta Ufficiale (LEGGE 13 luglio 2015, n. 107).

National Research Council (US). Committee on Information Technology Literacy. (1999). Being *fluent with information technology*. National Academies Press.

National Research Council. (2010). *Committee for the Workshops on Computational Thinking: Report of a workshop on the scope and nature of computational thinking*. Washington, DC: National Academies Press.

National Research Council. (2011). *Committee for the Workshops on Computational Thinking: Report of a workshop of pedagogical aspects of computational thinking*. Washington, DC: National Academies Press.

National Research Council. (2012). *A framework for K–12 science education: Practices, crosscutting concepts, and core ideas*. Washington, DC: National Academies Press.

Olimpo, G. (2015). Pensiero computazionale = buona programmazione e non solo. In V. Midoro (a cura di), *La scuola ai tempi del digitale*. Milano: Franco Angeli.

Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. Basic Books, Inc..Chicago.
Papert, S., & Harel, I. (1991). Situating constructionism. *Constructionism*,*36*, 1-11.
Pólya, G. (1957). *How to Solve It*. Garden City, NY: Doubleday: 253.
Resnick M. (2012). *Let's teach kids to code*. [TED Talks). Video: https://www.ted.com/talks/mitch_resnick_let_s_teach_kids_to_code?language=en
Schoolnet, E. (2014). *Computing our future. computer programming and coding priorities, school curricula and initiatives across europe*. Technical report European Schoolnet.
Werner, L., Denner, J., & Campe, S. (2006). IT fluency from a project-based program for middle school students. Journal of Computer Science Education Online, 2, 205-206.
Wing, J. M. (2006). Computational thinking. *Communications of the ACM*,*49*(3), 33-35.
Wing, J. M. (2011, March). Computational thinking. In *VL/HCC* (p. 3).