

Pensiero computazionale vs Apprendimento sistemico

Computational thinking vs Systemic learning

Demetrio Ria

Università del Salento, Lecce
demetrio.ria@unisalento.it

ABSTRACT

Information and communication technologies (ICTs) are conducive to a transformational renovation of society. For a sustainable society, however, it is necessary to transform the way of thinking beyond ordering, preserving and interacting with the information. Computer science (CS) professionals already have a set of conceptual tools designed to solve their problems. This kit is sometimes referred to as “computational thinking”. Computational thinking, however, tends to see the world in terms of a number of problems (or problem-types) that have computational solutions or in any case can be traced back to known resolution models. In this paper, we try to argue in support of an idea of a bridge needed to unite computational thought to the practice of sustainability. Sustainability, understood in a systemic sense, is an approach that recognizes the close interrelationship between technology, human behavior and the environment. In the pages that follow, we seek to develop, albeit as a sketch, an idea that provides a regional pedagogical ontology of the sustainable relationship between community, environment and technology starting from a critical analysis of computational thinking (CT).

Le tecnologie dell'informazione e della comunicazione (TIC) sono portatrici di un cambiamento trasformatore della società. Per una società sostenibile è necessario, però, trasformare il modo di pensare oltre che quello di ordinare, conservare e interagire con l'informazione. I professionisti della computer science (CS) hanno già un kit di strumenti concettuali finalizzati alla soluzione dei loro problemi. Questo kit, a volte, viene definito genericamente come “pensiero computazionale”. Il pensiero computazionale, però, tende a vedere il mondo in termini di una serie di problemi (o di problem-tipi) che hanno soluzioni computazionali o comunque possono essere ricondotti a modelli risolutivi noti. In questo lavoro, si cerca di argomentare a sostegno di una idea di ponte necessario che unisca il pensiero computazionale alla pratica della sostenibilità. La sostenibilità, intesa in senso sistemico, è un approccio che consente di riconoscere la stretta interrelazione-interdipendenza tra tecnologia, comportamento umano e ambiente. Nelle pagine che seguono si cerca di sviluppare, sia pure come abbozzo, una idea che fornisca una ontologia regionale pedagogica della relazione sostenibile tra comunità, ambiente e tecnologie a partire da una analisi critica del computational thinking (CT).

KEYWORDS

Computational thinking, Systems Thinking, Computers and Society, Computer Science Education, Computational Sustainability.
Pensiero Computazionale, Pensiero Sistemico, Computer e Società, Educazione Informatica, Sostenibilità Computazionale.

1. Introduzione

La legge 107 del 2015, nota anche come “La Buona Scuola”, esalta in modo evidente il ruolo delle tecnologie dell’informazione per la scuola del XXI° secolo. Vi sono aspetti che evidenziano l’importanza delle tecnologie per la didattica in generale e per alcune specifiche discipline che sono spesso fanalino di coda nelle rilevazioni INVALSI o OCSE-PISA. Con questa scelta politica si è posta significativa attenzione al Coding e in generale al “pensiero computazionale”. Ciò ha generato un flusso, sempre più intenso, di percorsi integrativi e/o curriculari di attività formative in quella direzione. In questi due anni trascorsi dalla promulgazione della legge è possibile notare che nelle scuole di ogni ordine e grado, si sono attivati percorsi formativi sul coding e sulla programmazione che hanno concentrato l’attenzione su alcuni aspetti che richiedono una opportuna attenzione pedagogica. Ciò che appare evidente è che la pervasività della tecnologia che fino ad oggi aveva influito in modo indiretto sulla scuola, adesso sta entrando direttamente in ambiente scolastico senza che però vi sia una adeguata analisi critica sull’argomento. È infatti noto a tutti che qualsiasi tecnologia che diventi troppo pervasiva alla fine minaccia la sostenibilità. In particolare è importante riconoscere che la tecnologia informatica (più di ogni altra) è uno strumento di educazione del pensiero e come tale richiede che ci sia una adeguata riflessione sui modi con cui ci si può consapevolmente rapportare a questi strumenti. Occorre favorire quindi sistemi di attività umane in cui la dinamica di relazione con questi tools consenta di governare la dipendenza tecnologica e non si lasci sopraffare. È un dato, infatti, che l’aumento dei canali d’infrastrutture software abbia “inavvertitamente” creato il loop di feedback del nostro comportamento sociale. Ciò ha impedito, di fatto, il cambiamento trasformativo vanificando, finora, ogni tentativo di adottare stili di vita più sostenibili.

Il “pensiero computazionale” nella traduzione operativa che si sta diffondendo nella scuola italiana, almeno così come emerge da alcuni primi e superficiali dati di ricerca su alcune scuole campione del Salento, appare soggetto ad alcune problematiche pedagogiche importanti. Ad un livello macro possiamo affermare che non ci permette di comprendere e di agire le trasformazioni necessarie per raggiungere la sostenibilità. Ad un livello “meso” non consente di sviluppare adeguatamente la crescita equilibrata delle intelligenze, emotive e creative, in primis, ma inficia anche quelle razionali e relazionali. Infine ad un livello micro, pare ridursi alla attuazione di atteggiamenti solutori e risolutivi del problema allontanandosi dalla visione complessiva del problem solving. Non si pone adeguata attenzione ad aspetti altrettanto importanti come il finding, setting, oppure la successiva funzione decisionale. Le esperienze positive appaiono là, dove si è creato il percorso all’interno di un vero e proprio contesto sistemico.

Il presente lavoro intende tracciare alcune riflessioni che potrebbero costituire una linea di ricerca e di approfondimento pedagogico intorno al tema del “pensiero computazionale”. Si tratteranno gli elementi essenziali del dibattito internazionale sulla questione. Si metteranno in luce alcune prospettive di integrabilità (almeno teorica) tra “pensiero computazionale” e “visione sistemica”. Infine, si tratterà una bozza di linea di ricerca che potrebbe sostenere un percorso di approccio al pensiero computazionale maggiormente orientato ad una visione pedagogica ed emancipante della formazione scolastica.

2. Il pensiero computazionale

Il dato da cui partire è che “pensiero computazionale” ed informatica non sono sinonimi. Il pensiero computazionale è l’esercizio di una pratica molto specifica,

dove si mettono a punto delle “procedure” che un “esecutore” (umano o macchina) all’interno di un “contesto” noto mette in atto per raggiungere un “obiettivo”. Una parte dell’informatica si occupa di questo argomento. Va anche detto che gli informatici tendono ad affrontare i problemi in un modo particolare. Vale a dire cercano soluzioni algoritmiche in termini di manipolazione dei dati e di controllo di processo. Jeanette Wing (2006), in un documento nel 2006, che ha fatto epoca, ha affermato che il pensiero computazionale può essere considerato il più importante contributo dell’informatica al mondo. Ella si è spinta fino a sostenere che questo modo di pensare dovrebbe essere insegnato a tutti gli studenti di tutte le discipline. Ha fatto molti esempi su come gli scienziati informatici affrontano i problemi. Quello che spesso viene ricordato è: “Quando tua figlia va a scuola la mattina, mette nel suo zaino le cose di cui ha bisogno per la giornata; questo è prefetching (recuperare) e caching. Quando tuo figlio perde i suoi guanti, e gli suggerisci di tornare sui suoi passi; questo è back-tracking. [...]”. Estrapolando da questi esempi, il messaggio generale è che gli scienziati del computer hanno una serie di strumenti e di metodi per generare la corrispondenza tra situazioni problematiche e tipi standard di soluzione. Inoltre, un aspetto altrettanto importante è che utilizzano una terminologia standard per descrivere questi modelli astratti di problem-solution.

A seguito di queste riflessioni e al dibattito (anche politico) che si è generato, la comunità informatica degli Stati Uniti ha prontamente adottato il pensiero computazionale come un importante obiettivo culturale. Per queste ragioni, ad esempio, la Computer Science Teachers Association (CSTA) ha istituito una task force per esplorare e diffondere l’insegnamento e l’apprendimento del pensiero computazionale. L’opinione considerata migliore, quella più diffusa e condivisa, considera il pensiero computazionale come un processo di problem solving¹. Considerano che il CT sia (anche se non si limita a) la capacità di formulare problemi in modo che un computer ci possa aiutare a risolverli. Ciò richiede che si sviluppino alcune capacità:

1. Organizzare ed analizzare dati.
2. Rappresentare dati attraverso astrazioni, modelli o simulazioni.
3. Automatizzare attraverso il pensiero algoritmico le soluzioni.
4. Identificare, analizzare e combinare le misure e le risorse.

La CSTA aggiunge che queste abilità sono supportate da una serie di atteggiamenti più ampi che, almeno nelle indicazioni operative, non si ravvisano. Tra queste abilità ne segnaliamo alcune che ci paiono veramente incidenti:

- La capacità di affrontare la complessità e problemi indeterminati,
- La tolleranza per l’ambiguità,
- L’abilità di lavorare con gli altri per raggiungere un obiettivo comune.

Il pensiero computazionale si concentra sull’applicazione al problema di un insieme di astrazioni. A questa capacità di pensare con molti livelli di astrazione si è aggiunto l’utilizzo della matematica per lo sviluppo di algoritmi (Denning, 2009). Tuttavia, dal momento in cui il concetto è stato introdotto, c’è stato ben poco dibattito critico su di esso. Le poche osservazioni che sono state scritte ten-

1 Va precisato che questa è una delle possibili accezioni del pensiero computazionale. È quella che viene maggiormente condivisa, ma non è l’unica.

dono a concentrarsi sulla vaghezza del termine (Jones, 2011), o sulla preoccupazione di non ridurre tutta la computer science ad uno solo dei suoi strumenti pratici. In questo senso, molto probabilmente, potremmo trovarci nella condizione come sostiene Maslow che “se tutto quello che hai è un martello, allora tutto sembra un chiodo” (Maslow, 1966). In altre parole, colui che opera con il pensiero computazionale cerca soltanto quei problemi che possono essere affrontati con i computer. Immediatamente questo fornisce una lente selettiva attraverso cui osservare il mondo. I problemi di cui è improbabile avere soluzioni computabili (ad esempio dilemmi etici, giudizi di valore, cambiamento sociale, ecc.) vengono ignorati.

Quello che emerge, in fondo, è che il CT si pone da un punto di vista intrinsecamente riduzionista. I problemi computazionali sono affrontati riducendoli a un insieme di variabili discrete che possono essere mappate su tipi di dati astratti, e una serie di fasi algoritmiche utili a manipolare questi dati. Sta di fatto che nel processo, prospettive multiple sulla natura del problema si perdono, così come qualsiasi conoscenza contingente locale circa la “situazione” problematica (Goguen, 1992; Easterbrook, 1991).

Nella definizione del CSTA la complessità che viene considerata è quella algoritmica, che offre poco alla comprensione del più ampio studio di sistemi adattativi complessi (Holling, 2001). Manson suggerisce che “la complessità algoritmica offre due contributi relativamente accessori alla teoria per complessità generale” (Manson, 2001). Da una parte misura lo sforzo necessario per risolvere un problema matematico, e la comprensione dei limiti di memorizzare e comunicare i dati forniti dalla teoria dell’informazione. Egli, però, identifica altre due forme di complessità necessarie per la comprensione del comportamento dei sistemi complessi, non lineari. Da una parte la complessità deterministica della teoria del caos e delle catastrofi. Dall’altra la complessità aggregata che sorge con il comportamento emergente dell’interazione di molti componenti all’interno di un sistema². Entrambe queste forme di complessità possono essere utilmente studiate attraverso i mezzi computazionali, principalmente attraverso l’utilizzo di modelli di simulazione³. Tuttavia, il CT, almeno come definito in letteratura, non fornisce un’opportuno insieme di concetti con cui è possibile realizzare tutto questo.

La riduzione dei problemi ai loro componenti computazionali conduce spesso verso pratiche che minano la sostenibilità. Si consideri per esempio, l’importanza attribuita all’ottimizzazione dei processi che sorge a causa del modo in cui si trattano le tecniche di scaling. Se si risolve un problema riducendolo a un insieme di astrazioni di base, è necessario scalare la soluzione computazionale di fronte a due risorse fondamentalmente limitate: spazio di memoria e di tempo del processore. Quindi, gli informatici sono addestrati per individuare soluzioni computazionali che si ottimizzino per spazio e tempo. La gestione del traffico in una città moderna offre un esempio illustrativo. Tramite il CT il problema della congestione del traffico viene approcciato come un problema di ottimizzazione: la congestione può essere ridotto ottimizzando flusso di traffico. L’idea è che se vengono raccolti più dati circa i volumi di traffico, sarà possibile regolare dinami-

- 2 Ad esempio, un problema veramente hard è quello posto da Chalmers che scrive: “Quando noi pensiamo o percepiamo c’è un’enorme attività di elaborazione dell’informazione, ma c’è anche un aspetto soggettivo. Per dirla come Nagel, *si prova qualcosa* a essere un organismo cosciente. Questo aspetto soggettivo è l’esperienza”. (Chalmers, 1996, p.201).
- 3 Su questi temi molto è stato detto da Domenico Parisi, per un approfondimento si rimanda pertanto a Parisi, 2001.

camente i segnali stradali ad ogni incrocio. Purtroppo, questa soluzione di solito ha l'effetto opposto: aumenta la congestione.

Morozov prende di mira tutto questo nel suo recente libro "To Save Everything, Click Here: The Folly of Technological Solutionism" (Morozov, 2013)⁴. Egli definisce il "Technological Solutionism" (TS) come quella convinzione che problemi sociali complessi possono essere ri-espressi attraverso porzioni ben definite con soluzioni calcolabili oppure con processi trasparenti ed evidenti facilmente ottimizzabili. In un certo senso con i giusti algoritmi tutti i problemi possono essere "calcolati". La riflessione di Morozov è più sottile di un semplice osservazione sui limiti del pensiero computazionale. L'idea chiave è che il nostro rapporto con la tecnologia ri-modella la nostra "infrastruttura di problem-solving" in modi inaspettati. Per esempio: l'idea di "gamification" (cambiare i comportamenti sociali, offrendo premi e incentivi, come in un gioco) è pericolosa. In questo modo ci si allena a pensare in termini di "regolazione" del singolo cittadino. Oppure, l'idea che siamo in grado di ottimizzare i processi sociali di raccolta e analisi di sempre più grandi insiemi di dati sul comportamento umano è pericolosa. Ci si allena a credere che le soluzioni ai nostri problemi possono essere trovate attraverso un modello automatizzato di corrispondenza dei comportamenti. Perdendo di vista completamente che il comportamento si colloca all'interno di un processo cooperativo di co-costruzione della comprensione delle forze che lo modellano.

Inoltre, il dato è esaltato dal successo di questi approcci su piccoli e limitati problemi. Ciò tende a far aumentare la nostra fiducia nel TS, riducendo così la nostra capacità di affrontare le sfide più importanti. Il termine "solutionism" serve anche a sottolineare che pensare in termini di problemi e di soluzioni in sé è spesso controproducente. Rittel e Webber (1973) sostengono che molte questioni sociali sono meglio pensate come dilemmi, a cui dobbiamo rispondere in modo intelligente, piuttosto che come i problemi da risolvere. Quindi ci si colloca tra problemi "addomesticabili", che hanno una chiara, definitiva formulazione e possibilità di una soluzione oggettivamente corretta, e "dilemmi" che sono profondamente radicati in una situazione complessa. I dilemmi hanno le seguenti proprietà:

- Non hanno una formulazione definitiva. Spesso, l'unico modo per comprendere appieno il problema è quello di tentare di risolverlo, e quindi non lo possiamo indicare chiaramente in anticipo.
- Non hanno una regola di arresto. Poiché non c'è fine alle catene causali che legano i sistemi aperti, l'unico limite alla soluzione è un giudizio di valore: "questo è quanto si può ragionevolmente raggiungere".
- Non è possibile valutare completamente una soluzione in qualsiasi periodo di tempo ragionevole.
- Ogni soluzione è un'operazione one-shot. Ogni soluzione tentata cambia la natura del problema, quindi non c'è possibilità di tentare d'imparare per tentativi ed errori.
- Non è possibile elencare tutte le soluzioni possibili.
- Ogni problema è essenzialmente unico. Mentre un problema specifico potrebbe essere simile ai precedenti, i fattori contestuali locali variano, e questi sono abbastanza significativi per impedire soluzioni generiche da lavorare.

4 Il libro è un gioco sulla convinzione che la Silicon Valley ci aiuterà a affrontare alcune delle sfide più grandi del mondo (fame, la povertà, la distruzione degli ecosistemi, i cambiamenti climatici), offrendoci nuove applicazioni per i nostri smartphone.

- Ogni dilemma può essere trattato come un sintomo di un altro problema.
- Ci sono diversi modi di spiegare la natura del problema. La scelta di come descrivere il problema determinerà la natura delle soluzioni accettabili. Inoltre, la scelta di spiegazione è solitamente determinata dalla visione della persona che descrive il problema.

Va precisato che il mancato studio dei dilemmi è diffuso in molte discipline⁵. Quindi si può ragionevolmente affermare che un eccessivo affidamento sul pensiero computazionale rischia di portare non solo a soluzioni computazionali che ignorano la sostenibilità sociale e ambientale, ma più spesso la minano attivamente.

3. Integrare il CT con il PENSIERO SISTEMICO

Il nucleo epistemologico della tesi qui sostenuta parte dal dato che il CT non ha completato la sua trasformazione in un vero e proprio sistema di pensiero. I sistemi di pensiero sono una componente essenziale di ogni tentativo di portare avanti il cambiamento trasformativo e performativo di una società sostenibile. I sistemi di pensiero possono superare le debolezze del CT come base concettuale per la progettazione di ICT. Per comprendere adeguatamente la ragioni della sostenibilità, quindi, abbiamo bisogno di capire i sistemi e le loro proprietà emergenti.

Da quanto si è detto prima si comprende che il CT soffre di tre carenze specifiche per le quali i sistemi di pensiero forniscono un toolkit concettuale più appropriato. Queste sono: particolarità della ontologia di dominio, poca apertura al cambiamento, quasi totale assenza di analisi critica. Ad una riflessione attenta si comprende che queste sono il frutto di una visione utilitaristica della formazione. Ma, proprio allo scopo di comprendere meglio la cornice concettuale, scendiamo un po' più nello specifico.

Si è già visto che il CT fornisce un'ontologia dei concetti di calcolo, e una serie di condizioni. Ad esempio, astrazioni procedurali e dati forniscono gli elementi costitutivi di soluzioni computazionali, e la composizione sequenziale e parallela provvedono a curare i modi di metterli insieme. La scomposizione gerarchica viene utilizzata per ridurre i problemi complessi e l'incapsulamento è utile allo scopo di creare soluzioni riutilizzabili. Tuttavia, questa ontologia si focalizza su soluzioni computazionali, di solito a scapito della analisi dettagliata dei problemi e dei contesti problematici. Inoltre, il sottocampo dei requisiti ingegneristici (RE) estende l'ontologia dei concetti computazionali (Nuseibeh and Easterbrook, 2000, pp. 35-46). In particolare, i RE aggiungono concetti come parti interessate, gli obiettivi (e la soddisfazione o la negazione di obiettivi), gli scenari, i compiti, e mezzi-fini di analisi. Fornisce anche gli strumenti per la modellazione di situazioni problematiche, come le attività di livello enterprise, e le interdipendenze tra gli attori in un'organizzazione. Quindi, le soluzioni computazionali vengono raramente valutate su più tempi o da diversi punti di vista. Si avverte pertanto la parzialità e il riduttivismo fenomenologico.

5 Ad esempio, la teoria economica classica è spesso criticata per la sua dipendenza da modelli matematici astratti che catturano e idealizzano il comportamento, piuttosto che ciò che accade realmente nel mondo. Sotto la lente di indicatori globali, come il PIL, tali processi tendono a separarci totalmente da un reale senso di benessere.

Il pensiero sistemico indirizza questa lacuna attraverso un insieme di concetti più adeguati a comprendere e ragionare sul comportamento. Questi concetti includono:

1. lo stock e i flussi. Il modo più semplice per iniziare una analisi dei sistemi è quello di esplorare in termini di scorte (quantità che variano nel tempo) e flussi (ingressi e uscite di titoli che riguardano la loro livelli). Gli stock possono essere quantitativi concreti, come il livello di acqua in un lago o la migrazione di persone di una città. Oppure, possono essere quantità astratte, come le credenze o desideri, le relazioni sociali, o il benessere.
2. il comportamento emergente. I sistemi tendono ad avere proprietà che non possono essere ricondotti a singoli componenti o gruppi di componenti, ma piuttosto derivare dall'interazione di tali componenti con l'intero sistema. Ad esempio, l'effetto rimbalzo è una proprietà emergente dell'interazione di misure di efficienza energetica con il comportamento economico dell'uomo. Vale a dire, il denaro risparmiato dalle misure di efficienza viene spesso utilizzato per altre attività ad alto consumo energetico (Chitnis, Sorrell, Druckman, Firth, and Jackson, 2013). Le persone raramente utilizzano le TIC nel modo in cui i progettisti si aspettano, ma il pensiero computazionale non affronta questo problema. Anche se spesso è difficile prevedere comportamenti emergenti, indicazioni utili possono essere acquisite abbandonando il riduzionismo, pensando in termini di sistemi adattativi complessi, e la modellazione del loro comportamento con dinamiche non lineari (Arthur, Durlauf, Lane and Lane, 1997).
3. Anelli di retroazione. La struttura di un sistema costringe i suoi comportamenti. Rinforzo gli anelli di retroazione tendono ad amplificare qualsiasi modifica all'interno del sistema attraverso una catena di causa ed effetto che alla fine fornisce ulteriore cambiamento nella stessa direzione. Balancing gli anelli di retroazione tendono a resistere al cambiamento, attraverso una catena di causa ed effetto che alla fine produce un cambiamento nella direzione opposta.

Il più grande difetto del CT è che non offre alcuno strumento concettuale per ragionare su come il cambiamento avviene nei sistemi complessi. Con il CT si tende ad assumere che il cambiamento avviene attraverso l'innovazione (vale a dire la creazione di una nuova soluzione di calcolo). Spesso si accompagna questa condizione con il presupposto che gli individui, se hanno strumenti migliori per aiutarli a diventare più sostenibili, agiscono in ambito sociale e ambientale.

Al contrario, l'approccio sistemico fornisce una serie di teorie del cambiamento estremamente ricche. Ad esempio, perché alcuni tentativi d'intervenire in un sistema sono più propensi a portare avanti il cambiamento rispetto ad altri? Meadows utilizza i concetti di stock e flussi per costruire una dettagliata analisi dei punti di leva (Meadows, 1999). In generale, le modifiche che alterano la struttura sottostante dei cicli di retroazione hanno maggiori probabilità di generare cambiamenti nel comportamento a livello di sistema rispetto alla modifica di valori delle variabili all'interno dello stesso. Una ulteriore teoria del cambiamento dei sistemi si basa sulle idee dalla teoria del caos. I sistemi sono spesso osservati per avere modelli stabili di comportamenti che durano nel tempo, indotti dai loro anelli di retroazione. Un sistema può riprendersi dopo un evento catastrofico, se le strutture che lo hanno creato in primo luogo sono ancora supportate. Ad esempio, un bosco ri-cresce dopo un incendio, perché esistono ancora le condizioni che hanno favorito la crescita delle foreste. Questo permette a un sistema di essere resistente di fronte di alcuni tipi di cambiamento (Walker and Salt, 2006). Tuttavia, piccoli cambiamenti possono nel tempo sopraffare que-

sti loop di feedback e guidare il sistema a una nuova serie di comportamenti. Il punto di svolta è indotto da un accumulo di piccoli cambiamenti all'interno del sistema che può essere invisibile agli osservatori esterni. Come con le proprietà emergenti, i punti di non ritorno possono essere facili da identificare dopo il fatto, ma difficili da prevedere (Dai, Vorselen, Korolev and Gore, 2012, pp. 1175–1177). Forse la teoria più completa di cambiamento nei sistemi di pensiero è il modello Panarchia (Gundersson and Holling, 2002). Questo modello nasce dalla constatazione che molti sistemi naturali passano attraverso un ciclo di sfruttamento e di crescita. Quindi da una fase stabile in cui le strutture del sistema diventano sempre più rigide e resistenti al cambiamento, e quindi in definitiva un collasso e una ristrutturazione. La teoria della Panarchia suggerisce che questo ciclo opera a più livelli in una gerarchia di sistema. Con una progressione di distruzione di creatività che è più lenta su grande scala, più rapida a livelli più ridotti.

Come si è detto vi è poi una terza debolezza del pensiero computazionale: non incoraggia il pensiero critico. Questa mancanza di pensiero critico è evidente nella letteratura tecnica, e ancora di più nelle storie delle TIC nei media, dove ogni nuovo pezzo di tecnologia è descritta in termini di funzionalità immediata. Alcun riguardo è riservato a valutazioni circa il suo impatto sulle divisioni più ampie della società, o sulla vita o sull'ambiente.

La teoria sistemica si presta naturalmente a un approccio critico, perché incoraggia l'idea che qualunque sistema può essere visto come componente in un sistema più grande. Weinberg chiama questo il principio di complementarità (Weinberg, 2001). In teoria, dovrebbe essere possibile per due persone che hanno una vista complementare di un sistema poter conciliare completamente i loro punti di vista. Tuttavia, in pratica, è spesso impossibile fare sufficienti osservazioni di un sistema complesso che permettano di conciliare completamente tutte discrepanze tra i punti di vista. È anche chiaro che i nostri modelli mentali di un sistema limitano la nostra capacità d'impegnarci con esso. La nostra comprensione di un sistema determina il tipo di osservazioni che siamo propensi a fare di esso. Non soltanto, ma tali osservazioni, a loro volta plasmano la nostra comprensione.

4. La proposta di una ricerca

Quindi il dato è che all'interno della cornice formale tutto appare perfettamente regolato. Ma oggi ha senso sviluppare capacità di problem solving, sia pure attraverso il CT, senza integrarne la dimensione del finding, del setting, del posing e, attraverso queste, sostenere i processi decisionali? Una delle conseguenze più evidenti dell'uso delle nuove tecnologie è proprio la necessità di sviluppare il pensiero strategico, una attitudine a compiere scelte e a prendere decisioni. Questo è particolarmente importante per sostenere l'imprenditorialità delle persone. Quindi, si ritiene che occorra integrare l'approccio computazionale con una più ex-cedente visione sistemica⁶. Concludendo, i limiti tracciati dal CT sono da

6 In quest'ottica da ormai diversi anni opera un gruppo di ricerca che, intorno a tematiche di confine della pedagogia, sviluppa ricerche sul campo orientate a individuare protocolli di attivazione di competenze. In ogni ricerca, infatti, si cercano vie d'integrazione tra innovazione sociale e tecnologica non perdendo di vista le specificità dell'intervento pedagogico. Una priorità che innova partendo dalla tradizione e dalla cultura propria delle comunità che vengono messe sotto la lente della ricerca. Vi sono molti

iscriversi sostanzialmente all'interno di una cornice concettuale che ha già avuto eminenti espressioni nella storia del pensiero⁷.

esempi, ma la matrice culturale è pressoché la stessa: rendere la comunità un ambiente in grado di essere attivato all'innovazione. Si scopre così che, ad esempio, le tecnologie, la computabilità dei problemi devono fare i conti con la singolarità degli universi personali e sociali. E soprattutto che, parafrasando Mozorov, non ci sono soltanto chiodi da piantare, così come non ci sono soltanto martelli da utilizzare. Ciò non significa che il CT non costituisca uno strumento utile, ma per diventare una competenza di base, come vuole Wing, necessita di una espansività che ancora non è stata sufficientemente indagata. Il gruppo di ricerca, coordinato dal Prof. Salvatore Colazzo dell'Università del Salento, è composto da diversi specialisti e ricercatori. Di recente hanno pubblicato per i tipi di Pearnon Editore il volume, "Istituzioni di Pedagogia e Didattica". Un manuale che è pensato per gli studenti, ma che mostra molto chiaramente il posizionamento scientifico del gruppo per il quale l'innovazione è l'esito di un processo insieme sistemico, generativo, sociale ed educativo. (Annacontini et alii, 2016.)

- 7 Leibniz, nel diciassettesimo secolo, sosteneva di poter escogitare una scrittura universale, mediante la quale fosse possibile eseguire calcoli su ogni genere di argomenti e trovare dimostrazioni come nell'algebra e nell'aritmetica. Ne la "Characteristica universalis" e il "Calculus ratiocinator" afferma che un ragionamento dimostrativo può essere trattato alla stregua di un problema matematico: le premesse corrispondono ai dati iniziali, la conclusione alla soluzione. Come per risolvere un problema si applicano, a partire dai dati iniziali, un certo numero di regole di calcolo per ricavare l'elemento incognito, così nel ragionamento da un gruppo di premesse attraverso regole di deduzione si giunge alla conclusione. Ancora più stupefacente è il fatto che già nel quarto secolo avanti Cristo, Platone (nel Filebo) abbia intravisto una estensione del significato del verbo calcolare per comprendere l'attività del pensiero coinvolta nella organizzazione del discorso argomentativo. Platone afferma che tra tutte le scienze il primato spetta alla dialettica, la più alta forma di conoscenza, «in grado di investigare la chiarezza, la precisione, e il massimo grado di verità». Ma, se non sai calcolare non riuscirai a discutere del bene e del male e la tua vita non sarà quella di un uomo, ma quella di un'ostrica o di una medusa". Nei procedimenti di calcolo e nei discorsi argomentativi si passa da un anello all'altro o con regole di calcolo o con regole logiche: da ciò che si sa e si conosce essere vero, si produce nuova verità. In entrambi i casi occorre disporre prioritariamente di un opportuno linguaggio preciso e rigoroso in cui formalizzare la questione, cioè tradurla in simboli, da trattare con regole di manipolazione che non facciano riferimento al significato. È un dato che ci sono diversi punti di vista da cui le persone muovono per risolvere i problemi. Ad esempio, alcune persone hanno una prospettiva idealista, il che significa che cercano di risolvere il problema ponendosi dall'interno di un quadro d'ideali e valori, di solito di natura religiosa o politica. Altre persone muovono da un punto di vista pratico o pragmatico, evidenziando soltanto i risultati realistici. L'approccio più diffuso è, comunque, la combinazione di un metodo analitico e una prospettiva idealista. Il metodo per prove ed errori è la forma più diffusa di risolvere i problemi, ma anche quella meno efficiente. L'approccio di base è esaminare il problema e cercare qualche idea per risolverlo. Se la prima idea non funziona, provare un'altra idea e così via fino a risolvere il problema. Sarebbe che questa forma di problem solving faccia parte della nostra genetica. Tutti noi lo usiamo in un momento o in un altro. In particolare di fronte a qualche nuova situazione che non comprendiamo pienamente e per la quale non siamo in grado d'individuare uno schema nella nostra memoria a lungo termine. Ma il metodo della prova ed errore di solito è un buon approccio ai problemi in cui il tempo non è un fattore importante. Al contrario non è una buona tecnica per problemi che non consentono una sufficiente ripetibilità dei tentativi o richiedono breve tempo per essere risolti. Uno dei vantaggi più interessanti di questa tecnica è che non richiede come prerequisito una grande mole di conoscenze. Al contrario necessita di grande pazienza, e non è un vero pensiero critico. L'approccio analitico alla soluzione dei problemi è invece l'essen-

Far diventare il CT una delle capacità di base dell'apprendimento e legarla alle naturali capacità lettura, scrittura e calcolo rappresenta una riduzione delle potenzialità dei nostri processi di pensiero. Occorre sostenere di più l'approccio analitico e l'utilizzo dell'intuizione, perché è molto meno distorto da giudizi e pregiudizi. L'approccio analitico alla soluzione dei problemi è di solito suddiviso in una serie di passaggi. Alcuni autori suggeriscono quattro fasi, altri fino a 13. Per esempio, il famoso matematico Georg Polya ne ha proposte quattro:

1. Capire il problema.
2. Escogitare un piano.
3. Effettuare il piano.
4. Guardare indietro (riflettere su ciò che ha funzionato e cosa no).

Polya ha elaborato ognuno dei suoi quattro passi in veri e propri elenchi di azioni specifiche che un potenziale risolutore di problemi dovrebbe prendere in considerazione. John Dewey ha proposto sette punti del problem-solving plan. Il metodo proposto da Dewey è un processo di pensiero riflessivo che è una serie organizzata e strutturata di domande. Seguendo questo schema concettuale ogni membro del gruppo si deve predisporre a rispondere a ciascuna delle domande nei primi quattro steps per il proprio specifico argomento.

Lo schema prevede:

Fase uno: definire il problema

1. Qual è il problema specifico che il gruppo deve affrontare?
2. Quali termini, concetti o idee devono essere definiti?

Fase due: analizzare il problema

1. Qual è la storia del problema?
2. Quali sono i sintomi del problema?
3. Quali sono le cause note del problema?
4. Quali metodi, (approcci, leggi, politiche) esistono attualmente per affrontare il problema?
5. Quali sono i limiti di questi metodi?

Fase tre: Determinare i criteri per la soluzione ottimale

Quali sono le linee guida per una soluzione praticabile? (Criteri di esempio possono includere il costo, la capacità di essere attuata, fatta rispettare).

Fase quattro: proporre soluzioni

Dopo che il gruppo ha analizzato il problema e criteri per una soluzione proposta, si dovrebbe cominciare a suggerire possibili soluzioni in termini ipotetici. Le attività dovrebbero portare ad una varietà di soluzioni possibili, senza una valutazione preventiva. (Brainstorming).

Fase cinque: Valutare la soluzione proposta

Dopo che il gruppo ha compilato un elenco di possibili soluzioni, alla luce dei criteri che sono stati sviluppati nella fase tre, si dovrebbe essere pronti per selezionare la migliore soluzione possibile.

za del pensiero critico. Quando ci avviciniamo a un problema dal punto di vista analitico, pensiamo razionalmente e in modo organizzato. Stiamo applicando criteri solidi per il nostro lavoro, in particolare la logica, la precisione e la chiarezza.

1. La soluzione porta dei vantaggi? Ci sono degli svantaggi? Gli svantaggi superano i vantaggi?
2. La soluzione è conforme ai criteri formulati dal gruppo? (Il gruppo può decidere di modificare i criteri).

Passo sei: scegliere una soluzione

1. Pesare meriti e mancanze della soluzione.
2. Se la soluzione viene adottata quali sono i suoi effetti a lungo termine, e a breve termine?

Passo Sette: suggerire strategie per implementare la soluzione

I membri del gruppo devono essere sicuri che la soluzione sarà davvero in grado di rispondere al problema. Dopo che il gruppo individua la soluzione si devono determinare i criteri con cui mettere la soluzione in vigore.

1. Come può il gruppo ottenere il sostegno del pubblico e l'approvazione per la sua proposta di soluzione?
2. Quali misure concrete sono necessarie per implementare la soluzione?
3. Come può il gruppo valutare il successo dei suoi sforzi di problem solving?

A quest'ultimo approccio ci ispiriamo in modo diretto. Se utilizzato con sistematicità e con un po' di pratica, può facilmente diventare una strategia cognitiva di grande efficacia. Intorno a questo modello si possono ricomporre curricula formativi che sviluppino competenze oltre che capacità di calcolo e si può riconoscere un più giusto ruolo al CT. La caratteristica essenziale di questo approccio alla risoluzione dei problemi è il fatto che consente di formalizzare direttamente i problemi di applicazione nella loro rappresentazione dichiarativa anche ad un alto livello di semantizzazione. La risoluzione di problemi dichiarativi con una rappresentazione di alto livello del problema in questione è strettamente correlata alla rappresentazione dichiarativa con cui si descrive la conoscenza esperta sul dominio problematico piuttosto che un metodo per la sua soluzione. Spesso, il compito di risolvere i problemi in ambito computazionale consiste nel reperire informazioni mancanti da questa rappresentazione relative all'obiettivo. Propriamente, la soluzione consiste nel trovare l'estensione di alcuni predicati che non sono specificatamente indicati nella rappresentazione. La computazione di tali estensioni della teoria che rappresenta il nostro problema è un compito abduttivo e come tale dovrebbe essere insegnato.

Bibliografia di riferimento

- Allan, W., Coulter, B., Denner, J., Erickson, J., Lee, I., Malyn-Smith, J., Martin, F. (2010). *Computational thinking for youth*. White Paper for the ITEST Small Working Group on Computational Thinking (CT).
- Annacontini, G. et alii (2016). *Istituzioni di pedagogia e didattica: manuale dell'attualità educativa e sociale*. Milano: Pearson.
- Arthur W. B., Durlauf S., Lane D., and Lane D. A., (1997). Introduction: Process and Emergence in the Economy. In *The Economy as an Evolving Complex System II*, W. B. Arthur and S. Durlauf, Eds. Addison-Wesley.
- Bandura, A. (2001). Social cognitive theory: An agentic perspective. *Annual Review of Psychology*, 52, 1-26.
- 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.
- Brennan, K., Valverde, A., Prempeh, J., Roque, R. & Chung, M. (2011). More than code: The significance of social interactions in young people's development as interactive media

- creators. In T. Bastiaens & M. Ebner (Eds.), *Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications 2011* (pp. 2147-2156). Chesapeake, VA: AACE.
- Chalmers, D. (1996). *The Conscious Mind: In search of a fundamental theory*. Oxford University Press, New York.
- Chitnis, M., Sorrell, S., Druckman, A., Firth, S. K., and Jackson, T. (2013). Turning lights into flights: Estimating direct and indirect rebound effects for UK households. *Energy Policy*, vol. 55, pp. 234–250.
- Dai, L., Vorselen, D., Korolev, K. S., and Gore, J. (2012). Generic Indicators for Loss of Resilience Before a Tipping Point Leading to Population Collapse. *Science*, vol. 336, no. 6085.
- Denning, P. J. (2009). Beyond computational thinking. *Communications of the ACM*, vol. 52, no. 6, p. 28, Jun.
- Easterbrook, S. M. (1991). Negotiation and the Role of the Requirements Specification. In Quintas, P. (Ed). *Social Dimensions of Systems Engineering: People, processes, policies and software development*. London: Ellis Horwood, 1993, no. July.
- Goguen, J. A. (1992). The Dry and the Wet. In E. Falkenberg, C. Rolland, and E.-S. El-Sayed, (Eds). *Information Systems Concepts* (pp. 1–17). North Holland: Elsevier.
- Gundersson, L. and Holling, C. S. (2002). *Panarchy: Understanding Transformations In Human And Natural Systems*. Washington, D.C.: Island Press.
- Holling, C. S. (2001). Understanding the Complexity of Economic, Ecological, and Social Systems. *Ecosystems*, 4(5), pp. 390–405, Aug.
- Jones, E. (2001). The Trouble with Computational Thinking. [Online]. Available: <https://c.ymcdn.com/sites/www.csteachers.org/resource/resmgr/JonesCTOnePager.pdf> (ultima visita: 02/10/2017).
- Manson, S. M. (2001). Simplifying complexity: a review of complexity theory. *Geoforum*, 32(3), pp. 405–414, Aug.
- Maslow, A., (1966). *The psychology of science: A reconnaissance*. Chapel Hill, NC: Maurice Bassett Publishing.
- Meadows, D. H. (1999). Leverage Points: Places to Intervene in a System. *The Sustainability Institute*, Tech. Rep.
- Morozov, E. (2013). *To Save Everything, Click Here: The Folly of Technological Solutionism*. PublicAffairs.
- Nuseibeh, B. and Easterbrook, S. M. (2000). Requirements Engineering: A Roadmap. In *The Future of Software Engineering*. Companion volume to the proceedings of the 22nd IEEE/ACM International Conference on Software Engineering (ICSE2000), A. C. W. Finkelstein, Ed. Limerick, Ireland: IEEE Computer Society Press, 2000, pp. 35–46.
- Parisi, D. (2001). *Simulazioni. La realtà rifatta nel computer*. Bologna: Il Mulino.
- Rittel, W. J. and Webber, M. M. (1973). Dilemmas in a General Theory of Planning. *Policy Sciences*, 4, pp. 155–169.
- Walker, B. H. and Salt, D. (2006). *Resilience thinking: sustaining ecosystems and people in a changing world*. Washington, D.C.: Island Press.
- Weinberg, G. M. (2001), *An Introduction to General Systems Theory*. New York, NY: Dorset House.
- Wing, J. (2006). Computational Thinking. *Communications of the ACM*, 49(3), pp. 33–35.