

Outreach in Computing Education: a Design Toolkit

AGNESE ADDONE

Dipartimento di Informatica, Università degli Studi di Salerno

addone@unisa.it

Abstract

Nella nostra società, chi impara affronta la costante evoluzione del mondo, il suo progresso tecnologico e le grandi domande che pone la tecnologia. La domanda incrementale di competenze informatiche riflette la diffusione di una formazione mirata, che possa ampliare l'istruzione culturale e tecnica. L'Educazione Informatica (COMPUTING EDUCATION) fornisce il fondamento teorico a questi obiettivi formativi, al fine di garantire il raggiungimento di una specifica conoscenza, delle competenze e della cittadinanza digitale. Parallelamente alle istituzioni formali e ai loro programmi di studio, esiste anche un mondo complementare di programmi di sensibilizzazione (OUTREACH PROGRAMS) sull'insegnamento dell'informatica, svolti al di fuori della scuola, in ambienti di apprendimento non formale e informale. Il contributo della mia tesi si inserisce nel perimetro di queste iniziative, soprattutto quelle rivolte agli studenti che vanno dalla primaria fino alla fine delle superiori e prima di accedere al percorso post-secondario o universitario, detti anche K-12. Fornirò una proposta di TOOLKIT di progettazione per supportare i ricercatori sugli OUTREACH PROGRAMS IN K-12 COMPUTING EDUCATION, sia nella fase di progettazione didattica che nell'analisi e risoluzione delle criticità più comuni.

La tecnologia sta plasmando le nostre vite a un ritmo crescente e sta modellando il modo in cui viviamo nel nostro mondo. Comunicazioni, *social media*, automazioni, transazioni, videogiochi, sono solo alcuni degli scopi principali con cui le tecnologie prendono forma. In questo contesto cresce la domanda di formazione specifica nel campo dell'Informatica, che consente un atteggiamento consapevole e competente nell'affrontare i contenuti e la varietà di queste tecnologie.

L'insegnamento dell'Informatica (*Computing Education*) è la disciplina e l'ambito di ricerca che si occupa di queste esigenze formative, dal Pensiero Computazionale (*Computational Thinking*) all'alfabetizzazione informatica avanzata, dalle competenze professionali al bilanciamento dei problemi di divario digitale (*digital divide*) al fine di garantire la partecipazione civica alla società.

Il sistema scolastico ha iniziato a rispondere a queste esigenze educative e culturali da qualche decennio, ma manca ancora una didattica diffusa a livello globale e il divario

si riflette nello sviluppo culturale, economico e sociale. Sebbene alcuni paesi abbiano iniziato a elaborare curricula scolastici in Informatica, molti altri non hanno ancora istituito programmi formali di Informatica.

Per sopperire alla mancanza di percorsi specifici, però, sono nate e sono tuttora diffuse molte iniziative extrascolastiche, che spesso fungono anche da volano di cambiamento nella scuola: i programmi di arricchimento (*Outreach Programs*) sono sessioni non formali e informali in cui i ragazzi possono accedere all'alfabetizzazione informatica in un ambiente progettato per soddisfare le loro motivazioni verso l'Informatica e le principali istanze educative della nostra società.

Anche se ricercatori ed educatori, in veste di progettisti di programmi *Outreach*, fanno riferimento ai contenuti del curriculum scolastico nella scelta di argomenti e contenuti, tuttavia queste sessioni rimangono complementari al sistema di istruzione formale in termini di caratteristiche peculiari (motivazione, organizzazione, principi didattici, obiettivi e

strategie) che li differenziano dall'istruzione e dall'organizzazione della scuola.

Contributi della tesi e metodo. Il tema della mia tesi si inserisce nel contesto dei *Computing Education Outreach Programs* per gli studenti delle scuole primarie e secondarie (detti anche *K-12* con notazione statunitense), con l'obiettivo di descrivere il processo di progettazione e di implementare un *toolkit* a supporto di ricercatori, educatori e volontari per superare i problemi principali. Molte preoccupazioni possono infatti sorgere e anche suggerire ai progettisti una serie di opzioni diverse durante l'elaborazione di una proposta di programma o durante la riprogettazione degli *Outreach* durante la fase di *follow-up*.

Pertanto la mia tesi si basa su due domande di ricerca (RQs):

- **RQ1** - Quali sono i principali problemi di progettazione di un *Outreach Program in K-12 Computing Education*?
- **RQ2** - Quali sono le possibili soluzioni da perseguire per garantire efficacia agli *Outreach Programs*?

In risposta alla RQ1, intendo classificare il processo e le preoccupazioni dei progettisti creando alcune categorie di questioni e relative aree di influenza; rispondendo alla RQ2 intendo invece dare un supporto tangibile ai ricercatori nella progettazione o riprogettazione dei programmi fornendo loro un pratico *toolkit*.

Oltre al contributo specifico nell'elaborazione del processo progettuale e del *toolkit*, la tesi affronta a livello teorico i problemi epistemologici della materia [20]. La *Computing Education* è infatti una materia relativamente recente che si colloca tra l'Informatica (*Computer Science*, CS) e la Pedagogia, i cui rispettivi esperti provengono da entrambi i campi disciplinari ibridando e fondendo le loro conoscenze e competenze [20].

Nella tesi ho tenuto conto di un metodo di ricerca che posso così riassumere:

- un *excursus* sulla storia dei *Computing Education Outreach Programs* e sui principali

contributi scientifici al legame inter- e trans-disciplinare tra Informatica, Pedagogia e Progettazione Educativa (*Instructional Design*);

- un'analisi e una tassonomia che ho condotto sull'argomento della mia ricerca, sia sugli *Outreach Programs* [2], che sui Linguaggi, Strumenti e Ambienti (*Languages, Tools and Environments*, TLE) [3];
- una descrizione del processo di progettazione degli *Outreach* e del metodo per rilevare le principali preoccupazioni;
- un *toolkit* di strumenti di progettazione, con linee guida per i progettisti sulle possibili soluzioni alle preoccupazioni di cui sopra.

Contesto e argomenti correlati. Il primo linguaggio informatico espressamente pensato per scopi didattici, LOGO, risale al 1967 [19] e, nello stesso periodo, troviamo le prime esperienze con i bambini nell'insegnamento e nell'apprendimento dell'informatica [16].

L'ulteriore definizione di *K-12 Computing Education* ha avuto uno sviluppo negli ultimi vent'anni, con studi che vanno dalla scuola all'università, allo sviluppo professionale degli insegnanti [7]. Attualmente, interessanti panoramiche che delineano la diversità dei programmi extrascolastici *K-12* sono attualmente offerte da alcuni lavori [4, 13], anche nella forma di revisioni sistematiche della letteratura (*Systematic Literature Reviews*) sulle iniziative [9, 2].

Lavori correlati sui programmi di *K-12 Computing Education* evidenziano il ruolo del design nella pianificazione di iniziative che promuovono abilità cruciali, come la risoluzione dei problemi, o che sottolineano la valutazione degli artefatti [15, 17]. In molti studi, la pedagogia principale è centrata sullo studente e sottolinea il ruolo attivo dello studente nel processo. L'efficacia degli *Outreach Programs* è anche legata al ruolo di incentivi o gratificazioni positive e al contesto sociale. In una delle sue note riflessioni, Seymour Papert ha osservato che "per garantire una reale acquisizione di conoscenza e aumentare la motivazione fornendo un *feedback* positivo, la progettazione

dell'ambiente e del contesto dovrebbe essere significativa per gli studenti e vicina alle loro esperienze e alle circostanze di vita significative" [16].

Outreach Programs. Sebbene in letteratura non esista una definizione univoca globale di cosa siano i *K-12 Outreach Programs*, possiamo definirli come le iniziative educative non formali e informali che scuole e università, ma anche aziende e associazioni, promuovono: realizzano o arricchiscono il curriculum scolastico degli studenti o lo sviluppo professionale degli insegnanti (*Teachers Professional Development*, TPD).

Gli *Outreach Programs* promuovono un punto di vista e una metodologia diversi o una nuova tecnologia che influisce sulle scelte future degli studenti verso le discipline informatiche o STEM (*Science, Technology, Engineering and Mathematics*). Gli approcci pedagogici adottati stimolano l'apprendimento autodiretto, la creatività, la collaborazione e il tutoraggio tra pari degli studenti, rimuovendo le barriere sociali ed emotive [1].

A volte possiamo trovare *Outreach* in iniziative regionali o nazionali, progettate dai governi attraverso le loro agenzie educative, che si rivolgono in maniera massiva a un intero segmento della popolazione scolastica del paese, ad esempio le scuole primarie o secondarie [14, 8].

La programmazione è un argomento molto popolare negli *Outreach*, poiché motiva i bambini a partecipare alle attività nelle discipline STEM [21, 12, 10, 23] e, tenendo il *Computational Thinking* e l'apprendimento creativo (*Creative Learning*) come *framework* teorici [5, 22], può anche incoraggiare gli studenti a perseverare e poi perseguire una carriera in Informatica durante il passaggio dalla scuola all'università [6].

Gli *Outreach Programs* attenuano i ruoli o le gerarchie formali tra insegnanti e studenti e creano un ambiente in cui i bambini possono conservare e guidare le proprie conoscenze con un approccio all'apprendimento più intenzionale piuttosto che incentrato sull'insegnante. Pertanto, gli *Outreach* non si basano direttamente

su un curriculum o un quadro istituzionale e di solito non comportano le tradizionali forme di valutazione, optando per l'autovalutazione, la risoluzione degli errori, la discussione comunitaria.

Gli *Outreach Programs* sono decisamente complementari all'istruzione formale, promuovono una sperimentazione costante di strumenti, ambienti, ambiti e metodologie [4], e si sintonizzano con le istanze di costruzione sociale del sapere della nostra epoca.

Il processo di progettazione. La progettazione degli *Outreach* di didattica dell'Informatica per i *K-12* assume forme diverse a seconda del contesto in cui si verifica. Progettare, identificare i problemi e le loro soluzioni, valutare i risultati, rappresentano i vari aspetti di una disciplina a supporto della progettazione e dell'attuazione di programmi educativi.

Ricercatori, educatori o professionisti che concepiscono e pianificano questi percorsi affrontano molteplici sfide e escogitano alcune soluzioni. A volte invece vedono le difficoltà incontrate solo dopo la realizzazione del programma, durante il *follow-up*. Per questi motivi, la nostra ricerca si concentra sui problemi di progettazione che i programmi di informatica per i *K-12* possono presentare ai progettisti di *Outreach* durante l'ideazione o anche dopo l'implementazione di una sessione.

Sulla base delle evidenze della letteratura e di ulteriori elementi derivati dalle scienze dell'apprendimento e dalle scienze sociali, ho elaborato le seguenti domande di ricerca (RQ) per modellare il processo:

- **RQ1** - Quali sono le categorie e le caratteristiche del processo di progettazione degli *Outreach Programs*?
- **RQ2** - Quali sono le fasi del processo di progettazione degli *Outreach Programs*?

La progettazione dei percorsi educativi si basa su alcuni pilastri che guidano i progettisti nella scelta di un aspetto educativo o sociale, un tipo di approccio pedagogico, la selezione di un target e le possibili aree di sviluppo personale da ampliare. Dall'analisi della letter-

atura, ho evidenziato questi pilastri:

1. Principi

- Competenze
- Sé
- Interazione
- Cittadinanza
- Apertura

2. Obiettivi

- **Competenze** - Aumentare la creatività, il pensiero critico, l'atteggiamento positivo, la tenacia, le prestazioni, l'autoregolazione; promuovere le discipline STEM; potenziare il TPD (sviluppo professionale dei docenti);
- **Sé** - Colmare i divari; promuovere l'inclusione, l'equità, l'impegno, l'*empowerment*, l'autoefficacia e la regolamentazione; adottare l'accessibilità e il design universale per l'apprendimento (UDL); superare la sottorappresentazione, i pregiudizi, la sindrome dell'impostore, il carico cognitivo, la procrastinazione;
- **Interazione** - Promuovere l'interazione sociale e l'apprendimento collaborativo; aumentare la collaborazione e il *team building*;
- **Cittadinanza** - Promuovere la consapevolezza digitale e l'uso sicuro di Internet;
- **Apertura** - Promuovere la generalizzazione dei programmi, scalabilità, riproducibilità; adottare l'iterazione, il riutilizzo di risorse e pratiche, l'accessibilità.

3. Strategie

- Pedagogia (approcci)
- TLE (*Tools, Languages, Environments*)
- Tutoraggio
- *Team/Community building*
- Artefatti
- Co-design/personalizzazione
- *Role-models*

- Valutazione/lezioni apprese
- Disseminazione
- Durata

In risposta alla RQ1 ho adottato una tassonomia lungo tre risorse, che incorporano le diverse domande da considerare durante la progettazione di un programma: preoccupazioni che influenzano la partecipazione di specifiche categorie di studenti e l'incremento del loro accesso all'Informatica; casi pratici da risolvere in vista della realizzazione di un programma; fase di progettazione ed elaborazione dell'iniziativa. Alcuni di essi pongono vincoli reali, in qualche modo impossibili da superare; alcuni sono solo sfaccettature flessibili che possono essere adattate all'obiettivo o ai target affrontati. Da un lato, abbiamo requisiti pratici da considerare e che definiscono un gruppo concreto di elementi; dall'altro si cerca di definire alcuni concetti progettuali generali e più teorici da sfruttare durante la progettazione.

In risposta alla RQ2, ho delineato le fasi del processo progettuale che tengono conto dei suddetti pilastri, adattandoli nei diversi contesti educativi:

- **Obiettivi** - Identificare i bisogni educativi e il target/pubblico, l'approccio pedagogico e il metodo, indagando lo stato dell'arte;
- **Ambientazione** - TLE adottati, reclutamento delle risorse umane, predisposizione di sede e ambiente formativo;
- **Motivazione** - Rimozione degli ostacoli (emotivi, sociali, infrastrutturali, logistici), promozione dei *Role-models*;
- **Insegnamento/Apprendimento** - Tutoraggio, *peer-mentoring*, collaborazione, sperimentazione pratica e *tinkering*, autoregolazione, autovalutazione;
- **Artefatto** - Oggetti fisici o virtuali, presentazione e disseminazione;
- **Feedback** - Raccolta pre- o post-sessione, miglioramento della progettazione degli *Outreach*, valutazione dell'accessibilità, co-progettazione;
- **Valutazione** - Analisi dell'impatto sull'apprendimento e sulle interazioni,

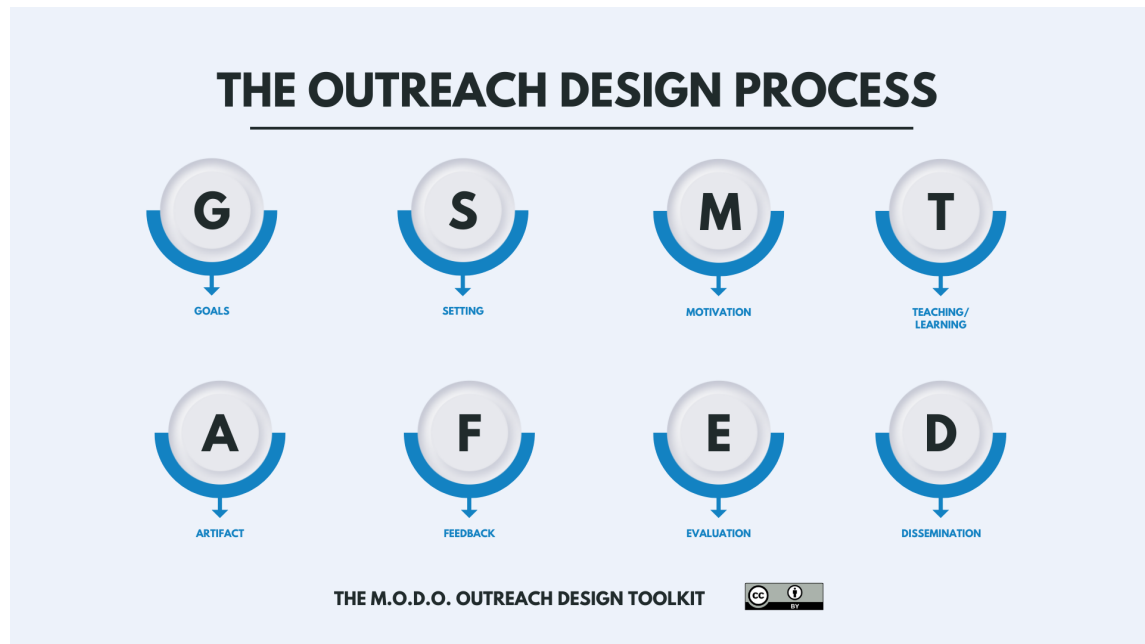


Figure 1: Il processo di progettazione degli Outreach Programs.

utilizzo dei feedback;

- **Disseminazione** - Condivisione dei risultati (pubblicazioni, campagne sui social media, antologia di artefatti degli studenti), valutazione della scalabilità e della riproducibilità.

Nel processo di progettazione è necessario analizzare attentamente e abbinare l'obiettivo dell'iniziativa al target o al pubblico a cui si rivolge, valutando anche la disponibilità di luoghi e strumenti fisici. È fondamentale intercettare i bisogni formativi emergenti e la reale motivazione dei partecipanti verso l'Informatica e la tecnologia, ma anche garantire una corretta organizzazione e ambientazione per svolgere i programmi. Il processo è riassunto nella Figura 1.

L'Outreach Toolkit. La definizione delle principali preoccupazioni progettuali genera una sorta di tassonomia delle intenzioni dei progettisti durante l'elaborazione di un programma. Le problematiche più comuni possono solitamente essere affrontate in loco con soluzioni temporanee di buon senso, ma è fondamentale anche suggerire risposte pratiche da

adottare preventivamente in fase progettuale che possano supportare l'efficacia delle iniziative.

Il modello che fornisco nella mia tesi non vuole essere definitivo o inclusivo di ogni aspetto della progettazione, ma piuttosto un passo verso la comunità scientifica per favorire lo scambio e la collaborazione reciproci. A causa dell'ampiezza del mio scopo, ho fiducia e sarò grata se la mia analisi dovesse generare ulteriori domande e risposte.

Le seguenti domande di ricerca (RQ) mirano a determinare l'importanza di un *toolkit* di progettazione:

- **RQ1** - Quali sono i principali problemi di progettazione negli Outreach?
- **RQ2** - Quali sono i criteri principali su cui tracciare le linee guida di un *toolkit* per i progettisti degli Outreach?

In risposta alle mie domande di ricerca ho esaminato la principale preoccupazione relativa a ogni pilastro di cui sopra e successivamente ho delineato una tassonomia degli Outreach lungo quattro criteri, che incorporano le diverse aree di interesse da considerare du-

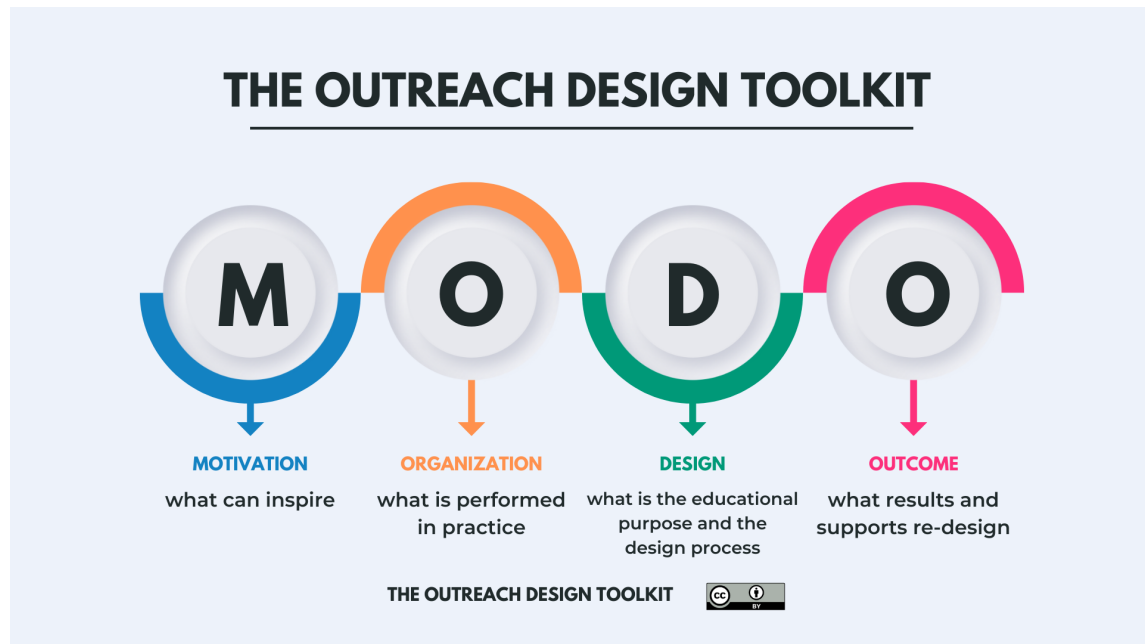


Figure 2: L'Outreach Design Toolkit per la Computing Education.

rante la progettazione di un programma o la riprogettazione dopo l'implementazione e nella fase di *follow-up*:

- **Motivazione** - *Ciò che ispira*. Include gli incentivi previsti dai progettisti che possono influenzare o agevolare la partecipazione di specifiche categorie di studenti, aumentando il loro accesso all'alfabetizzazione informatica. È anche l'insieme delle aspettative, dei desideri e delle percezioni personali che gli studenti hanno rispetto alle discipline STEM e all'Informatica in particolare;
- **Organizzazione** - *Ciò che viene realizzato nella pratica*. Comprende le istanze operative per predisporre l'ambiente educativo e sociale in vista dell'attuazione di un programma;
- **Design** - *Ciò che rappresenta lo scopo educativo e il processo progettuale*. Riguarda la progettazione, la fase di elaborazione e le soluzioni adottate per l'iniziativa, con riguardo sia agli aspetti didattici che a quelli computazionali;
- **Esito** - *Ciò che è il risultato e può supportare*

la riprogettazione. Riguarda le forme e la qualità dei risultati, tangibili o meno, dei programmi e supporta la fase di riprogettazione.

Le quattro caratteristiche dei criteri discendono dai pilastri degli *Outreach* e le principali tra esse sono riassunte nella Figura 2.

Un problema ancora difficile da gestire negli *Outreach* e non incluso nel *toolkit* è la valutazione dei risultati dell'apprendimento. Difficile, infatti, valutarne scientificamente l'efficacia e l'impatto a causa della loro natura sporadica (vincoli di durata) e perché le modalità di attuazione e fruizione sono estremamente variabili. Per quanto riguarda la valutazione e l'eventuale certificazione o convalida dei risultati, tra l'altro, ci sono alcuni quadri istituzionali interessanti [18] e contributi scientifici che possono ispirare i designer [6, 11].

Potenziali applicazioni. Il *toolkit* mira principalmente a supportare ricercatori, educatori e professionisti nella pianificazione degli *Outreach Programs*. Ad esempio, molti corsi si svolgono in contesti come campi estivi o *coding club*, dove capita che i volontari siano impegnati

nella risoluzione di problemi più strettamente pedagogici.

Il *toolkit* può supportare sia la fase progettuale che quella di *follow-up* per la risoluzione delle problematiche più comuni e frequenti, fornendo linee guida nelle singole aree progettuali evidenziate con i criteri di cui sopra.

Direzioni future. Un possibile sviluppo del *toolkit* di progettazione potrebbe riguardare l'applicazione nell'aggiornamento professionale dei docenti (TPD), in cui le relazioni e l'autoregolazione pesano sui risultati. In questo caso, la difficoltà degli adulti a mettersi in gioco e a condividere senza esitazioni il proprio percorso di apprendimento, invertendo sostanzialmente il proprio ruolo da docente a discente, ha un effetto non indifferente. Un altro importante sviluppo futuro che potrebbe essere preso in considerazione e che potrebbe valere la pena di approfondire è la valutazione dell'impatto degli *Outreach Programs*, soprattutto individuando un sistema qualitativamente efficace per implementarlo. Infine, un utile contributo che può rappresentare una direzione futura per questo lavoro di ricerca è una definizione più completa di una tassonomia dei TLE. Nel nostro laboratorio di ricerca stiamo già lavorando all'ulteriore implementazione della nostra piattaforma collaborativa di tassonomia degli *Outreach* in Computing Education [3], aperto al contributo dei ricercatori da tutto il mondo.

Conclusione. L'educazione informatica o *Computing Education* è un campo relativamente giovane e in rapido sviluppo che risponde alla domanda di educazione tecnologica e alla costruzione di conoscenze e competenze informatiche. Risponde in modo inter- e cross-disciplinare, con ricercatori provenienti sia dal perimetro dell'Informatica che della Pedagogia. L'inevitabile e necessaria contaminazione tra i due gruppi genera alcune riflessioni sempre più cruciali sia sul piano teorico ed epistemologico che sulla pratica degli esperimenti messi in atto.

I programmi di *Outreach in K-12 Computing Education* sono una realtà consolidata che viene costantemente realizzata oltre il curriculum for-

male e ha un impatto formativo e sociale sui partecipanti. Sono ambienti di apprendimento che integrano elementi del curriculum scolastico di Informatica con la motivazione e le esigenze sociali dei partecipanti e che basano la loro azione sull'incentivo a partecipare, collaborare, sperimentare linguaggi e strumenti, colmando *gap* e *digital divide*.

I ricercatori devono spesso affrontare alcuni problemi di progettazione e sfide specifiche durante l'attuazione dei programmi. Il processo progettuale, descritto nelle sue fasi, in questa tesi si arricchisce anche di una proposta di *toolkit* operativo non ancora riscontrabile in letteratura. Le soluzioni suggerite soddisfano la maggior parte delle preoccupazioni progettuali, ma abbiamo anche evidenziato alcuni vincoli didattici che non vengono risolti al momento, come la valutazione dell'impatto delle iniziative. Sarebbe interessante, come direzione futura, prevedere una convalida del quadro educativo e del *toolkit* con una serie di esperimenti volti a risolvere le preoccupazioni evidenziate.

Il *toolkit* può anche avere un'utilità per gli operatori non direttamente coinvolti nella ricerca o nel campo dell'istruzione, come i volontari o i lavoratori aziendali che svolgono sessioni informali di *Outreach*. In molti paesi in cui non esiste un curriculum scolastico di Informatica, i loro campi e club offrono effettivamente l'unico accesso significativo all'educazione informatica per molti studenti.

REFERENCES

- [1] Agnese Addone, Renato De Donato, Giuseppina Palmieri, Maria Angela Pellegrino, Andrea Petta, Vittorio Scarano, and Luigi Serra. *Novelette*, a usable visual storytelling digital learning environment. *IEEE Access*, 9:168850–168868, 2021.
- [2] Agnese Addone and Vittorio Scarano. Outreach in K-12 programming: A systematic literature review on audience and purpose. In *Proceedings of the 14th International Conference on Computer Supported*

- Education, CSEDU*, volume 2, pages 276–283. SCITEPRESS, 2022.
- [3] Agnese Addone, Vittorio Scarano, and Carmine Spagnuolo. Towards a collaborative taxonomy of tools, languages and environments in K-12 computing education. In *IEEE Frontiers in Education Conference (FIE)*, pages 1–5, 2022.
- [4] Andrew Begel and Amy J. Ko. *Learning Outside the Classroom*, page 749–772. The Cambridge Handbook of Computing Education Research. Cambridge University Press, 2019.
- [5] Karen Brennan. A case for why: Society, school, self. In *Computational thinking education in K–12: Artificial intelligence literacy and physical computing*. The MIT Press, 2022.
- [6] Adrienne Decker and Monica M. McGill. Pre-college computing outreach research: Towards improving the practice. In *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education, SIGCSE '17*, page 153–158, New York, NY, USA, 2017. Association for Computing Machinery.
- [7] Sally A Fincher and Anthony V Robins. *The Cambridge handbook of computing education research*. Cambridge University Press, 2019.
- [8] Adam Fletcher, Raina Mason, and Graham Cooper. Helping students get it: Investigating the longitudinal impacts of IT school outreach in australia. In *Proceedings of the 23rd Australasian Computing Education Conference, ACE '21*, page 115–124. Association for Computing Machinery, 2021.
- [9] Tracy Gardner, Hayley C. Leonard, Jane Waite, and Sue Sentance. What do we know about Computing Education for K-12 in non-formal settings? A Systematic Literature Review of recent research. In *Proceedings of the 2022 ACM Conference on International Computing Education Research - Volume 1, ICER '22*, page 264–281. Association for Computing Machinery, 2022.
- [10] Yasmin B Kafai and Quinn Burke. *Connected code: Why children need to learn programming*. MIT Press, 2014.
- [11] Jay Lemke, Robert Locusay, Michael Cole, and Vera Michalchik. *Documenting and assessing learning in informal and media-rich environments*. The MIT Press, 2015.
- [12] Andrew Luxton-Reilly, Simon, Ibrahim Albluwi, Brett A. Becker, Michail Giannakos, Amruth N. Kumar, Linda Ott, James Paterson, Michael James Scott, Judy Sheard, and Claudia Szabo. Introductory programming: A systematic literature review. In *Proceedings Companion of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education, ITiCSE Companion*, page 55–106. Association for Computing Machinery, 2018.
- [13] Victoria J Marsick and Karen E Watkins. Informal and incidental learning. *New directions for adult and continuing education*, 2001(89):25–34, 2001.
- [14] Karen Nolan, Roisin Faherty, Keith Quille, Brett A. Becker, and Susan Bergin. Developing an inclusive K-12 outreach model. In *Proceedings of the 2020 ACM Conference on Innovation and Technology in Computer Science Education, ITiCSE '20*, page 145–151. Association for Computing Machinery, 2020.
- [15] Alannah Oleson, Brett Wortzman, and Amy J. Ko. On the role of design in K-12 computing education. *ACM Trans. Comput. Educ.*, 21(1), dec 2021.
- [16] Seymour A Papert. *Mindstorms: Children, computers, and powerful ideas*. Basic Books, New York, 1980.
- [17] Joslenne Pena. Seeding the computational skills of diverse non-programmers through non-formal workshops. In *Proceedings of the 2019 ACM Conference on International Computing Education Research*,

- ICER '19, page 347–348, New York, NY, USA, 2019. Association for Computing Machinery.
- [18] Madhu Singh. UNESCO guidelines for the recognition, validation and accreditation of the outcomes of non-formal and informal learning. *PLA Inside Out: An International Journal on Theory, Research and Practice in Prior Learning Assessment*, 1(2), 2012.
- [19] Cynthia Solomon, Brian Harvey, Ken Kahn, Henry Lieberman, Mark L. Miller, Margaret Minsky, Artemis Papert, and Brian Silverman. History of LOGO. *Proc. ACM Program. Lang.*, 4(HOPL), 2020.
- [20] Claudia Szabo, Nickolas Falkner, Andrew Petersen, Heather Bort, Cornelia Connolly, Kathryn Cunningham, Peter Donaldson, Arto Hellas, James Robinson, and Judy Sheard. A periodic table of computing education learning theories. In *Proceedings of the 2019 ACM Conference on Innovation and Technology in Computer Science Education, ITiCSE '19*, page 269–270. Association for Computing Machinery, 2019.
- [21] Claudia Szabo, Judy Sheard, Andrew Luxton-Reilly, Simon, Brett A. Becker, and Linda Ott. Fifteen years of introductory programming in schools: A global overview of K-12 initiatives. In *Proceedings of the 19th Koli Calling International Conference on Computing Education Research, Koli Calling '19*. Association for Computing Machinery, 2019.
- [22] David Weintrop, Elham Beheshti, Michael Horn, Kai Orton, Kemi Jona, Laura Trouille, and Uri Wilensky. Defining computational thinking for mathematics and science classrooms. *Journal of science education and technology*, 25(1):127–147, 2016.
- [23] David Weintrop and Uri Wilensky. To block or not to block, that is the question: students' perceptions of blocks-based programming. In *Proceedings of the 14th inter-*
- national conference on interaction design and children*, pages 199–208, 2015.