Chapter

Coding and Creativity: Reflections and Design Proposals

Simona Ferrari and Federica Pelizzari

Abstract

The aim of the chapter is to reflect on and guide the design of coding from the perspective of creativity and the development of critical thinking. The assumption is that coding is seen from a functionalist perspective: it is used to know and practice languages that allow and force a culture of market-driven schooling. Starting from presenting and discussing four different paradigms for viewing code, we will show why emancipatory and interpretive paradigms could introduce coding to develop creativity and give students the capacity to be true democratic citizens of the world. We will describe design elements of these two paradigms and the connections with a media educative point of view. Therefore, this chapter examines coding from an emancipatory perspective and uses critical thinking to reduce the risk of being controlled by the informational society.

Keywords: coding, creativity, paradigms, media education, critical thinking

1. Introduction

The term coding, although now widely and variously mind used, “suffers” from a certain underlying ambiguity that conditions a consistent and homogeneous application in different educational contexts.

This ambiguity derives primarily from the fact that the literal translation of the term is that of “making code”, the sense of which is not unambiguous and can be understood as “assigning a code”, “translating into a code,” and “writing code” with the purpose of providing a machine or other entity with the instructions necessary to make them operate according to our intentions.

Some confusion of perspectives also results from this in the definition of the concept of “computational thinking” [1–3] which has always been connected to the concept of coding.

Definitions include that of Wing [4], who considers computational thinking to be the ability to solve problems, including those related to understanding human behavior, using systems and approaches specific to the computer sciences, such as abstraction, debugging, and remixing.

Aho [5] takes up this perspective in part and considers computational thinking as the set of thought processes involved in formulating and solving problems through solutions that can be represented as computational steps and algorithms.

The equivocal of the coding concept strongly conditions its development and the possibility of grasping its real opportunities to "teach thinking," proliferating visions of its application: the possibility of supporting the development of logical thinking, the development of specific problem-solving skills [6], and the opportunity to further engage students in the study of science subjects and computer science.

Against this background, this chapter investigates the relationship that coding can have with creativity and how both can be developed within instructional designs in preschool and elementary school.

2. Conceptual framework of creativity

Creativity starts from the idea of contrasting divergent and convergent thinking.

According to Guilford [7], convergent thinking operates within established patterns, approaches the problem with a certain method and, through the latter, and finds the only possible solution. However, using divergent thinking outside the established patterns allows one to approach the problem with a fresh perspective, arriving at original solutions and identifying the creative process with the typical dynamics of problem-solving. Thus, divergent thinking is expressed in not only the search for exact solutions but also in the multiplicity and originality of the answers given, the richness of ideas, and the restructuring of the subject matter.

Various models have been developed to explain the mechanism that regulates or from which creativity originates, including the factorials (cognitivist-oriented) models, which consider creative thinking to be an articulated unit that can be broken down into parts called factors and identified through surveys and statistical analysis [8–10].

Sternberg and Lubart [11] carried out a comprehensive survey of the landscape of creativity studies and observed that historically, this line of research has faced several obstacles, probably due to a broad cultural legacy that regarded creativity as something "mystical" and unexplainable.

According to these studies, creativity consists of the "ability to produce something new (original, unexpected) and appropriate (useful, adaptable to the set task)," thus elaborating the investment theory of creativity [12].

Finally, Resnick [13], analyzing the ways in which children learn, seeks to identify and enhance the creative dimension as the key to meeting and overcoming the challenge facing today's children to become tomorrow's adults.

As Resnick argues in the TED Talk "Let's teach kids to code" (https://www.ted.com/talks/mitch_resnick_let_s_teach_kids_to_code), when children create a coding project, they also learn to program; however, more importantly, they also program to learn. Because by learning to program, they learn a thousand other things, thus opening up new learning opportunities.

From these considerations, Resnick opens up to the view of the learning process represented as a "spiral of creative learning" (**Figure 1**): exploration of the world (and consequent knowledge) occurs through manipulating objects and experimenting, building things and testing their functionality, reasoning by prototypes and identifying errors…all ways in which children learn and through which they develop knowledge of the fundamental laws of the environment in which they live.

This should be the training ground for exercising creative thinking throughout life, during which each individual must continue to learn to exist in the world. Imagining, creating, experimenting, sharing, and reflecting should be the stages of

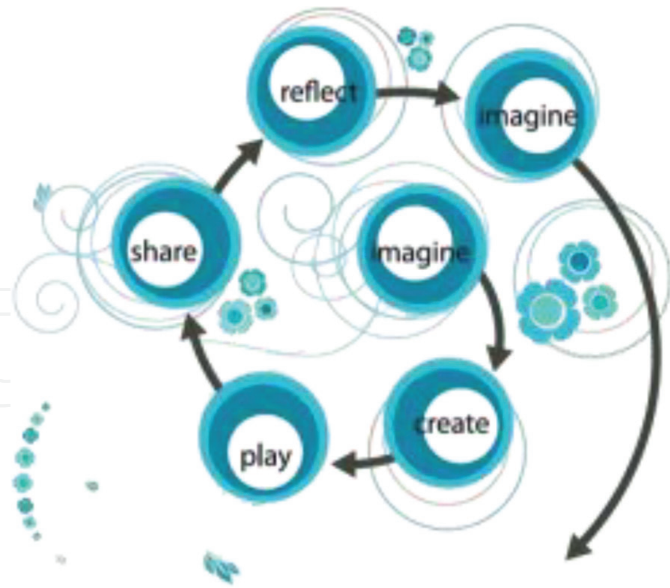

Figure 1.
Resnick's spiral of creativity. Source: https://www.flickr.com/photos/wfryer/37920982305. Author: Wesley Fryer. License: CC-BY 2.0.

a process to be reproduced continuously, the cyclical repetition of a sequence (to use an appropriate metaphor, in terms of programming) to be applied and cultivated throughout life, as the inexhaustible engine of one's learning process.

Resnick [14] goes further to think about how design that works with coding and creativity can work. This led to the 4P model, which consists of designing teaching around four key words: Project, Peer, Passion, and Play. The four P indicate:

- Projects: people learn best when they are actively working on meaningful projects, generating new ideas, designing prototypes, and iteratively improving them;
- Peers: when learning becomes a social activity, people can exchange ideas, collaborate on projects, and grow together;
- Passion: when people work on projects they care about, they work longer, harder, overcome challenges, and learn more along the way;
- Play: learning involves fun experimentation. Try new things, tinker with materials, test your limits, take risks, and do it repeatedly.

From this perspective, coding should be introduced in school as a cross-curricular activity precisely because cross-curricular is the skill it enables. Computational thinking does not require technology, and it precedes technology.

The adoption of coding as an activity to exemplify concepts, describe procedures, solve problems, and find solutions can be entrusted to teachers of any discipline; in fact, this activity does not require specific computer skills, as it provides an interdisciplinary perspective, combining creativity and imagination with logic and mathematics.

Learning to be effective must be meaningful [15], which means it must motivate and engage pupils actively, bringing both logical and creative competence to bear.

3. Paradigms of coding

Creativity and coding do not always stand together. It depends how teachers perceive them. To reflect on coding from the perspective of creativity's conceptual framework presented, it is necessary to introduce four possible paradigms of coding [16]: postmodernist, functionalist, interpretivist, and emancipatory.

The first understands coding as a creative activity oriented to the think-make-improve process; it finds its natural application in informal settings, within spaces such as FabLabs or communities such as CoderDojo. In addition to being oriented by what the media returns on the topic, it focuses attention on coding as a tool that—in line with digital media—calls for the revision of teaching practices, the return of the laboratory, the flipped lesson, and interrelationships between informal and formal learning.

In contrast, the functionalist paradigm approaches coding as a language useful for better understanding school subjects, on which programming activities can be grafted. Strongly recalling an idea of school as a space of instruction for profit [17], it approaches coding from a disciplinary perspective [18–20].

The third, interpretive, uses coding to develop critical analysis; coding is a device to develop critical thinking. The actions of disassembling to understand and reassembling to create [21] are the basis of the creative approach based on problems and solutions [22] that well activates the use-modify-create transition [23].

Finally, the emancipatory paradigm resorts to coding to overcome the dictatorship of the script [24]. In a political-social context, it starts from self-awareness and empowerment and goes beyond the digital into the outside world, trying to unhinge its logic.

In order to better understand the implications of these four paradigms for constructing implementation paths, it is possible to think of an organization that works on a dual axis.

On the one hand, the first axis is what we might call functional enrollment: from this perspective, coding can play a facilitating function with respect to the adaptation of subjects to a society like ours marked by the cultural and productive prominence of information technology, or a critical function of soliciting suspicion with respect to the risk of homogenization and the renunciation of thought.

On the other hand, the second axis is what we might refer to as the axis of educational enrollment: from this perspective, coding can be thought of both as a pedagogical logic through which to build the citizen of tomorrow and as a social logic aimed at releasing energies and activating resources. In the former case, we move within formal contexts (such as school), while in the latter, we occupy nonformal contexts. In the first case, coding is an education; in the second, it is a form of expression, a way of being, even an experience of media-activism.

Constructing the two axes in the form of a Cartesian plane, four quadrants are identified to examine as many ways of thinking about coding (**Figure 2**).

The hypothesis emerging from discussing with teachers and working in the classroom is that the adaptation perspective prevails in the teachers' representations. Either coding is an activity that serves primarily to prepare future professionals in school by getting them accustomed to interacting with the languages of computer science (functionalist paradigm), or to unleash the creative possibilities of children who are finally allowed to express themselves in their most natural ways (postmodernist paradigm).

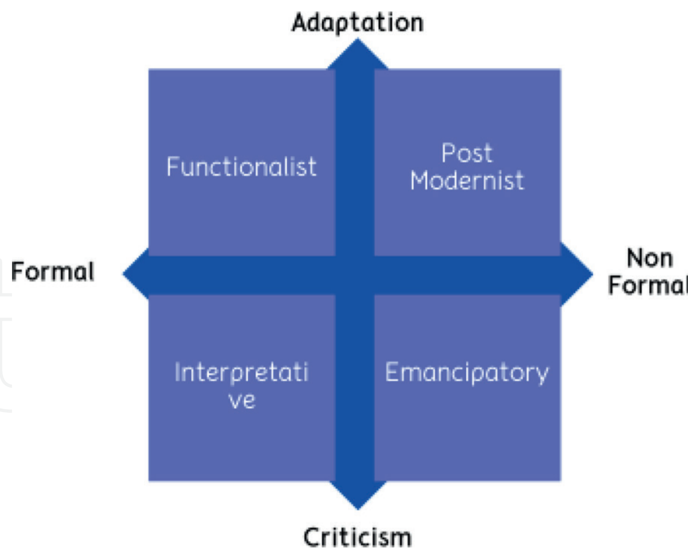

Figure 2.
Matrix of the four paradigms with respect to coding education.

Education's interest is on the lower end of the quadrant. The protagonist here is critical thinking, the call for deconstructive reflection, the systematic teaching of suspicion as a means of gaining meaning at a deeper level. From this perspective, coding ceases to be thought of as a gymnasium of future computing and is seen in its more specific pedagogical valence, namely that of being a media education activity, moving beyond the dictatorship of the script. This expression refers to the inherent ambiguity of the 2.0 logic; in fact, while merely filling in the format presents itself as a winning aspect of these applications, by virtue of the ease and navigability this entails, it also results in standardization. The script encourages the dissemination and polarity of computer applications but robs the user of the ability to modify formats. Owning the code, in this perspective, means knowing what is "behind the script" and being able to modify it if one wants to.

Therefore, the problem is not the new technology itself but how it is used; one can opt for games and activities that make one passive in this interaction or one can devise tools and programs that open up a thousand possibilities and engage children in the process of creative learning and playing.

To understand how teachers see coding, it is important to collect their representations of it.

Representations are cognitive systems, processes adopted by subjects to control the natural fear of the unknown, "to understand and act upon society, serve them as a reference frame for their thoughts and decisions, and color their imagination" ([25], p. 952). Moscovici had studied anchoring and objectification as systems that allow making familiar what is unfamiliar or novel. In particular, Moscovici had emphasized the role of social representations as a guide of behavior.

To investigate what is the most popular representation of coding, we recalled a set of 12 images (three images for each model) (**Figure 3**) employed in a previous study designed by Center for Research in Media, Innovation and Technology Education (CREMIT) and completed by 989 subjects [26].

The images metaphorically represent the various characteristics of the four models: the functionalist model and the postmodernist model are visualized through think-make-improve activity and more logical-mathematical research, while the

Figure 3.
Four coding paradigms and images.

interpretive model is visualized as critical analysis and the emancipatory model as overcoming the script.

Images provide an alternative to word-based surveys and are growing interest in social research methods [27]. The two surveys in this study involved 24 kindergarten educators and 23 primary school teachers in 2021 and 2022.

Educators and teachers were asked to choose the image that, in their opinion, represented the vision of coding and how it works in education. They were not told either that these images depicted approaches to coding on a symbolic level or what paradigm each image was associated with. In **Figures 4** and **5**, you can see the results. It is evident that postmodernist representation is the most common one.

Merging the data from the individual images, it can be seen that in kindergarten educators' representations of coding, the postmodernist model emerges overwhelmingly (76%), followed by the functionalist (18%) and the emancipatory models (6%).

Furthermore, in primary school teachers, the representations are even dichotomous: 74% of them represent themselves in the postmodernist model and 26% in the functionalist model (**Figures 6** and 7). These data bring with them a reflection: it is precisely educators and teachers who need to change their perspective, leading students to be able to develop divergent thinking and meaningful learning.

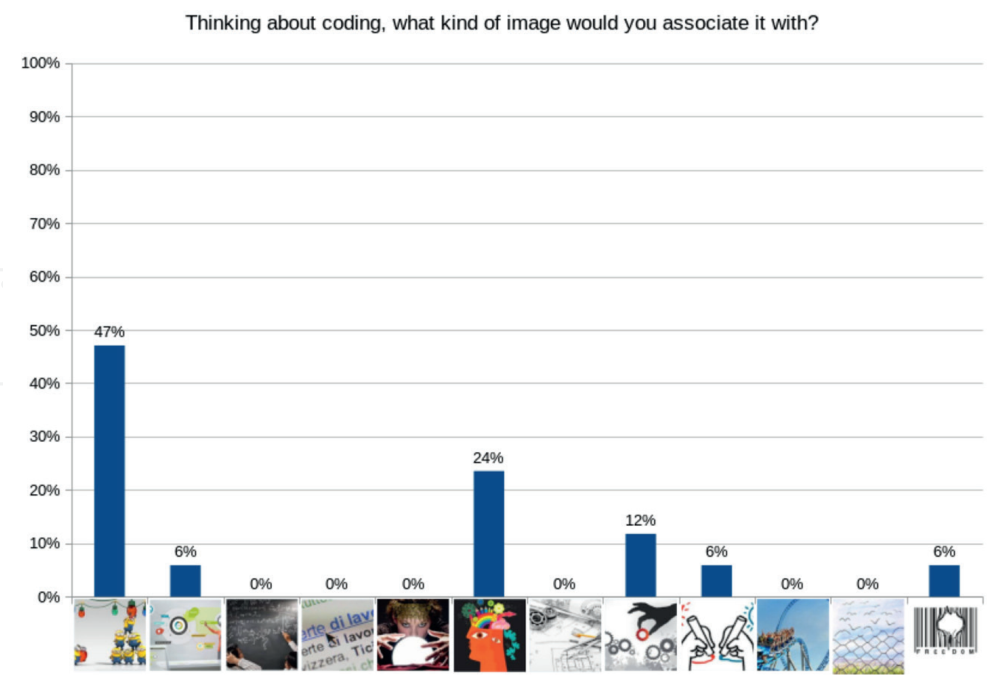

Figure 4.
Images of coding for preschool educators.

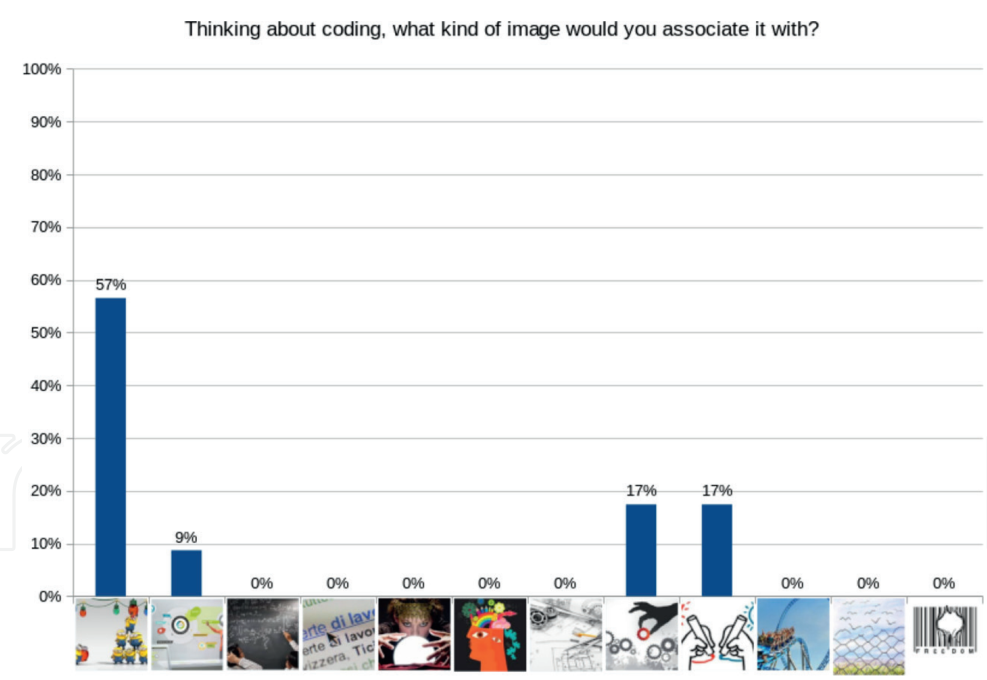

Figure 5.
Images of coding for primary teachers.

Theoretical framework on creativity and on coding with these initial data was discussed with educators and teachers to reflect about teaching and learning. This phase has been the starting point to change the design of coding, moving it from a framework related to the simple application of code to one open to creativity and to monitor and to evaluate the outcomes of learning.

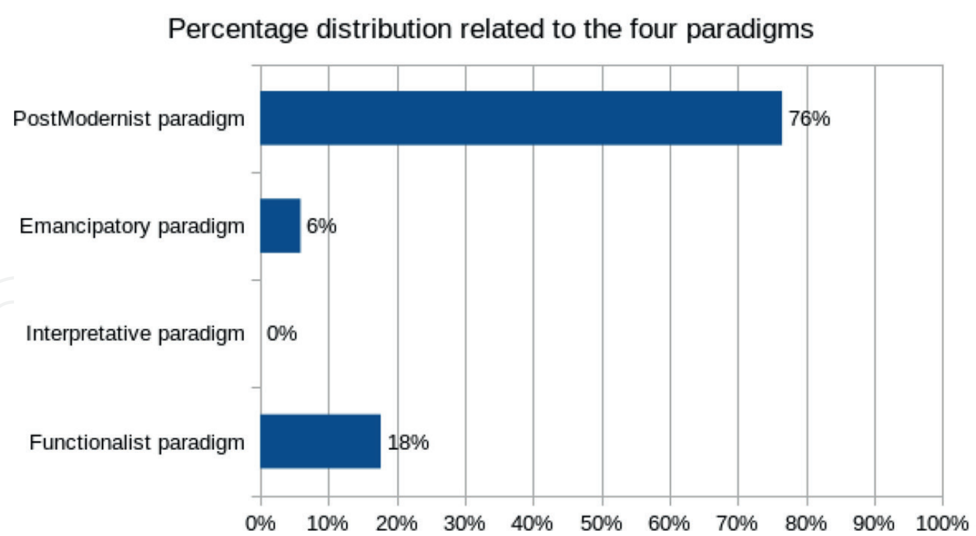

Figure 6.
Distribution of paradigms for preschool educators.

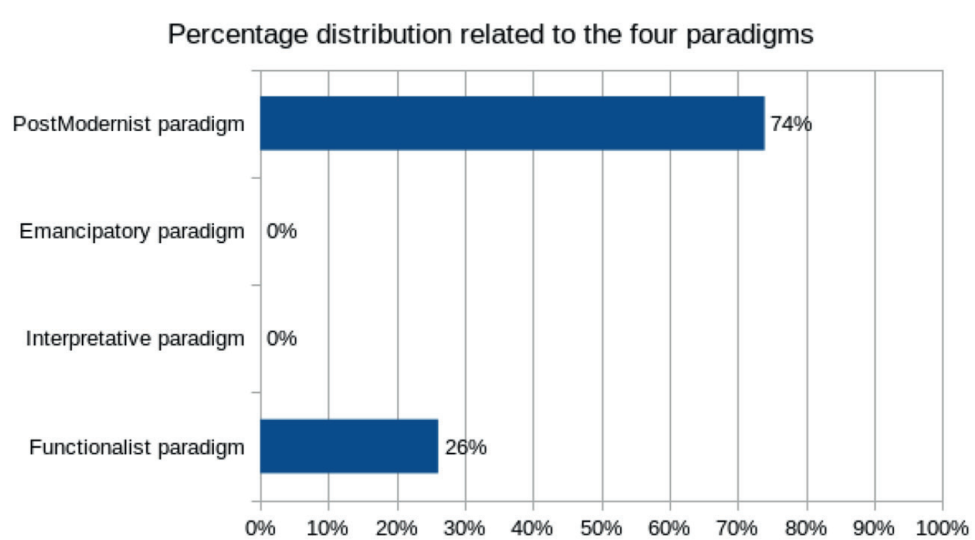

Figure 7.
Distribution of paradigms for primary teachers.

4. Good design practice

If coding can admit such a broad framework of application possibilities in education, it is possible to reason about it in terms of skills and problem-solving strategies [28]. Specifically, in response to the emerging need to shift the design of coding from an alphabetical-informational paradigm to a creative paradigm, we have moved on three sliders:

- of competence;
- of problem-solving;
- of Media Education.

Let us start with the first one. As research focus, coding is mainly developed within specific hours, referred to as code hours [29, 30], and how it is used to explore STEM disciplines or logical-mathematical skills with computational thinking [31, 32]. However, this does not show how coding allows one to go beyond skills and get to developing competencies.

According to Le Boterf [33], competence is not reducible to a set of atomic and separate performances, but rather tends to be thought of as an integration of the resources possessed by the individual, involving the activation of knowledge, skills, and personal dispositions relating to both the cognitive and the socio-emotional and volitional planes. Its expression requires bringing into play and mobilizing the wholeness of the person in its multiple dimensions [34]. It also requires going beyond behaviors observable and to pay attention to the internal dispositions of the subject.

According to this understanding, the construct of competence turns out to be inclusive of the different dimensions involved in the learning process [35], which can be traced to the following three planes:

- knowledge, understood as representations of the world that the subject constructs for himself through the prompts that come to him from the external environment and codified knowledge;

- skills, understood as operational schemes that enable the subject to act in physical and mental form on material or symbolic objects;

- dispositions to act, understood as the subject's attitudes to relate to the context in which he or she operates.

In this perspective, competences reside in the mobilization of the individual's resources (knowledge, skills, attitudes ...), not in the resources themselves. Thus, they take the form of knowing how to act (or react) in a given situation or context to achieve a performance.

Moreover, developing situated competence means working not only on the subject's resources but also on the conditions that lead him or her to effectively mobilize his or her resources (knowledge, skills and personal, and social and/or methodological abilities) in relation to a situation-problem, with the aim of proposing effective responses that express their full responsibility and autonomy [36]. To "act competently," a person must be able to "read" the situation-problem [37] according to "competent" patterns, leading him or her to interpret it, assign meaning to it and, consequently, make relevant decisions. Based on such decisions, the person will take effective actions in response to the situation itself, choosing from a set of strategies available to him or her [38]. Finally, the person will have to evaluate in progress the quality of his or her interpretations and actions, revising and changing them should they prove inadequate in the course of events.

Coming back to coding, it develops skills of a significant and transversal type, leading to a model of "competent action" that continues to review, deepen, and bring into play the learning system of the individual student and the class group.

From this perspective, in line with Le Boterf [39], we can place the coding actions designed by educators and teachers on the slider of competence (**Figure 8**).

The competence's slider makes it visible and questions the design of student's requests: complex situations require multidimensional understanding of needs; mobilize resources (knowledge, skills, and attitudes) at individual level or social one;

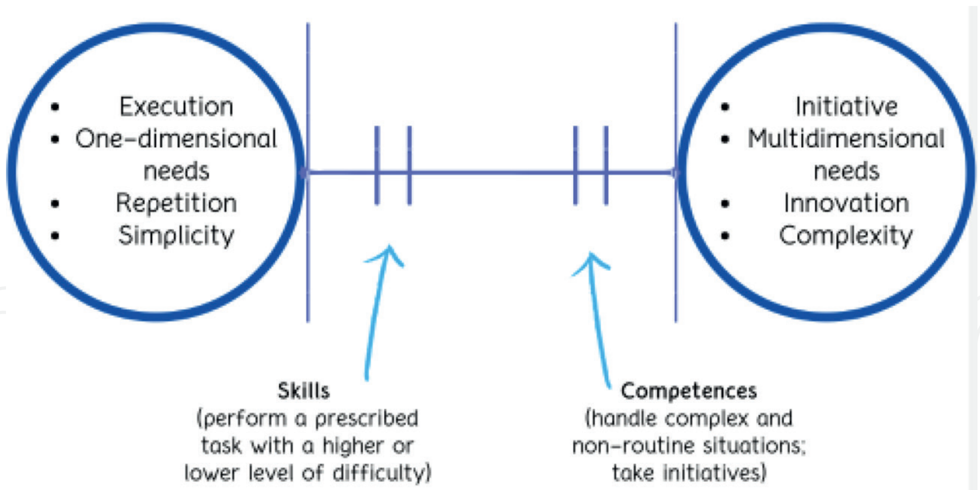

Figure 8.
Le Boterf's slider of competence.

broad visions in terms of resolving possibilities; request to take initiatives and are open to innovations; and set up a setting that allows the possibility to act safely, to take responsibility and risks without fear of negative feedback or failure. Complex situations involve students in:

- acting:
- wanting to act on an internal and external personal motivation [40];
- being able to act.

From this perspective, we can insert ourselves with a reflection on the type of task and situation-problem that it is necessary to propose within the teaching of coding. The goal of the activities must be to give meta-reflective momentum to the students [41], allowing them to re-evaluate the error as a way to learn and return to their own steps by analyzing how to adjust, improve, and change their practice.

Second slider is related to the problem-solving process and student's involvement. The scientific literature reports that problem-solving in coding is essential, and it is the first skill developed and observed [42–45].

On this basis, it is essential to propose activities that move from Veridical Decision-Making (VDM) to Adaptive Decision-Making (ADM) [46], two strategies studied within the neuroscientific understanding about how our brain makes decisions.

VDM occurs when the problem situation provides only one correct solution, and the task is to find it. Conversely, ADM occurs when the problem situation provides more than one possibility for effective solution and the task consists of finding the most functional one.

If you try to combine the slider of competence with these two decision-making strategies, you can notice how VDM is typical of the development of skills, while ADM can be inserted within the competences.

Moreover, you can then think about the design of teaching with coding from the perspective of ADM and competences, going to work on resources, setting and motivation just proposing situations-wide problem, with different possible solutions and with a collaborative and comparative expendability that allows students to be meta-reflective among themselves, thus enhancing the continuous communication-relational exchange.

If these two factors of competence and strategy work on the problem, it is possible to graft the four coding paradigms explored earlier.

Considering the representations of educators and teachers, it is possible to place the functionalist and postmodernist paradigms within the development of skills and work on VDM. This is because programming language decision-making and the think-make-improve system leads students to focus on a linear problem situation with a single solution that can be taken apart and reassembled and that has right or wrong feedback within it, without the need for meta-reflection work. In these two paradigms, the work is possible to do individually and without continuous comparison with peers.

In contrast, the interpretive and emancipatory paradigms lead students to work from the perspective of competences and AMD logic. The reason for this is precisely implicit in action design: working on critical analysis and overcoming the script require having complex problem situations, which require broad solutions and may be different depending on the type of resolution perspective. Moreover, applied decision-making must adapt to the context and be flexible with respect to possible changes and relaunches given by peers, with whom it is necessary to work to arrive at functional resolutions.

In **Table 1**, let us try to summarize the design features that coding can have, working on three types of skills: decision-making, problem reading, and its resultant and transversal competence.

The last slider refers to the media education approach. If the intention of coding is to go to work on media activism as well, from the perspective presented here, it is possible to place it within new media literacy education [47], considering critical development in terms of rethinking media and their algorithms, not only on a technical level.

The goal of the new media literacy education is not adaptation, the mere acquisition of skills to interact with technological tools that are dropped on us from above [48]. The goal of media education for democracy is to train for citizenship, for the acquisition of the skills to use and think about technologies in a critical and empowering sense [49]. In addition, in this model, coding comes full circle only if it is creative and participatory.

In addition, this can only be done through an instructional design of coding that moves away from the logic of the Hour of Code and closer to an active strategy that

	Design for skills—VDM	Design by competence—ADM
"Reading" the problem	"Closed" problems: only one way of interpreting the situation	Problems "open" to multiple interpretations
Way of dealing the problem	One single solution	Multiple solution strategies
Process work system	Individual	Collaborative/cooperative
Evaluation of activities performed	Right/wrong feedback	Meta-reflection on one's strategies

Table 1.
Summary table of approaches to coding versus instructional design sliders.

fits cross-curricular with the teaching of disciplines and fields of experience, with a relaxed and vertical time.

Of course, the design shift is not easy to apply. However, if it is true that coding aims to develop complex and therefore flexible thinking, it will be increasingly necessary to move forward in this direction.

The sliders are the three important designing questions for the development of creativity through coding: put in place by teachers, it is then possible to strive and destroy simple univocal thinking to open up toward creative thinking of a thousand opportunities.

5. Conclusion

The perspective that has been outlined leads us in conclusion to look for the relationship between coding and creativity in three essential movements in education: the first is related to microlearning [50], which increasingly requires a student-centered design [51, 52] that sees the teacher as the one who not only leads but also guides the activities and thinking. The second is related to the way of reviewing and reevaluating convergent and divergent thinking; if the former leads the student to learn in a closed way, which does not allow him/her to see knowledge and skills from his/her own perspective, it is necessary to deepen and develop the latter, as forma mentis and as a modus operandi, not only of the student but also of the teacher. Finally, the third movement is that of moving from product assessment and observation to process assessment and observation [53]; it is in this process that meaningful learning takes place and enactment becomes possible, allowing one to develop one's own innovative thinking.

These three elements come together precisely in the coding methodology, not only bringing the student to the center but allowing him or her to see different solutions to real problems and looking at the work process as the focus of creation and metacognition activities [54, 55]. Moreover, if these three elements are developed in a circular sense, creativity will be developed consistently and fluently.

The final value of coding lies precisely in its educational potential: doing coding means developing critical thinking, it means doing Media Education [56].

Additional information

The chapter was jointly designed by the authors. In particular, Simona Ferrari drafted paragraphs Paradigms of coding and Good design practice, and Federica Pelizzari drafted paragraphs Introduction, Conceptual framework of creativity, and Conclusion.

Coding and Creativity: Reflections and Design Proposals
DOI: http://dx.doi.org/10.5772/intechopen.109971

Author details

Simona Ferrari* and Federica Pelizzari
Catholic University of the Sacred Heart, Milan, Italy

*Address all correspondence to: simona.ferrari@unicatt.it

References

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

[2] Brennan K, Resnick M. New frameworks for studying and assessing the development of computational thinking. In: Proceedings of the 2012 Annual Meeting of the American Educational Research Association. Vol. 1. Vancouver, Canada: MIT Media Lab; 2012. p. 25

[3] Grover S, Pea R. Computational thinking in K–12: A review of the state of the field. Educational Researcher. 2013;**42**(1):38-43

[4] Wing JM. Computational thinking. Communications of the ACM. 2006;**49**(3):33-35

[5] Aho AV. Computation and computational thinking. The Computer Journal. 2012;**55**(7):832-835

[6] Norman GRH. Problem-solving skills, solving problems and problem-based learning. Medical Education. 1988;**22**(4):279-286

[7] Guilford JP. Creativity: Yesterday, today and tomorrow. The Journal of Creative Behavior. 1967;**1**(1):3-14

[8] Drevdahl JE. Factors of importance for creativity. Journal of Clinical Psychology. 1956;**1956**:21-26

[9] Torrance EP. The nature of creativity as manifest in its testing. In: The Nature of Creativity. Cambridge University Press; 1988. pp. 43-75

[10] De Bono E. Serious Creativity: How to be Creative under Pressure and Turn Ideas into Action. New York: Random House; 2015

[11] Sternberg RJ, Lubart TI. The concept of creativity: Prospects and paradigms. In: Handbook of Creativity. Vol. 1. Cambridge University Press; 1999. pp. 3-15

[12] Sternberg RJ. The nature of creativity. In: The Essential Sternberg: Essays on Intelligence, Psychology and Education. Springer Publishing Company; 2009. pp. 103-118

[13] Resnick M. Come i bambini: immagina, crea, gioca e condividi. Coltivare la creatività con il Lifelong Kindergarten del MIT. Trento: Edizioni Centro Studi Erickson; 2018

[14] Resnick M. Give P'sa chance: Projects, peers, passion, play. In: Constructionism and Creativity: Proceedings of the Third International Constructionism Conference. Vienna: Austrian Computer Society; 2014. pp. 13-20

[15] Barron B, Darling-Hammond L. Teaching for Meaningful Learning: A Review of Research on Inquiry-Based and Cooperative Learning. George Lucas Educational Foundation; 2008

[16] Dufva T, Dufva M. Metaphors of code—Structuring and broadening the discussion on teaching children to code. Thinking Skills and Creativity. 2016;**22**:97-110

[17] Nussbaum MC. Non per profitto. Perché le democrazie hanno bisogno della cultura umanistica. Bologna: Il Mulino; 2011

[18] Fessakis G, Gouli E, Mavroudi E. Problem solving by 5-6 years old kindergarten children in a computer

programming environment: A case study. Computers & Education. 2013;**63**:87-97

[19] Hui TH, Umar IN. Does a combination of metaphor and pairing activity help programming performance of students with different self-regulated learning levels? Turkish Online Journal of Educational Technology-TOJET. 2011;**10**(4):121-129

[20] Kazakoff ER, Bers MU. Put your robot in, put your robot out: Sequencing through programming robots in early childhood. Journal of Educational Computing Research. 2014;**50**(4):553-573

[21] Cope B, Kalantzis M. Multiliteracies: Literacy Learning and the Design of Social Futures. Oxford: Routledge, Psychology Press; 2000

[22] Perkovi L, Settle A, Hwang S, Jones J. A framework for computational thinking across the curriculum. ACM. 2010;**10**:123-127

[23] Lee L, Martin F, Denner J, Coulter B, Allan W, Erickson J, et al. Computational thinking for youth in practice. ACM Inroads. 2011;**2**(1):32-37

[24] Lanier J. Tu non sei un gadget. Milano: Arnoldo Mondadori Editore; 2010

[25] Moscovici S. The myth of the lonely paradigm: A rejoinder. Social Research. 1984;**1984**:939-967

[26] Ferrari S, Rivoltella PC, Rosa A, Mangione G. Fare coding per emanciparsi. In: Limone P, Parmigiani D, editors. Modelli pedagogici e pratiche didattiche per la formazione iniziale e in servizio degli insegnanti. Bari: Progedit; 2017

[27] Mamali C. The value of images for exploring the functions of social representations: Toward self-generated pictorial social representations. In: A Comment on "History, Emotions and Hetero-referential Representations" by Sen and Wagner. Papers on Social Representations. 2005. p. 15

[28] Su J, Yang W, Li H. A scoping review of studies on coding curriculum in early childhood: Investigating its design, implementation, and evaluation. Journal of Research in Childhood Education. 2022;**2022**:1-21

[29] Yauney J, Bartholomew SR, Rich P. A Systematic Review of "Hour of Code" Research. Computer Science Education; 2022. pp. 1-33

[30] Du J, Wimmer H, Rada R. "Hour of code": A case study. Information Systems Education Journal. 2018;**16**(1):51

[31] Sengupta P, Dickes A, Farris A. Toward a phenomenology of computational thinking in STEM education. In: Computational Thinking in the STEM Disciplines. Springer; 2018. pp. 49-72

[32] Martín-Páez T, Aguilera D, Perales-Palacios FJ, Vílchez-González JM. What are we talking about when we talk about STEM education? A review of literature. Science Education. 2019;**103**(4):799-822

[33] Le Boterf G. De la compétence. Essai sur un attracteur étrange. 1994;**1994**:16-18

[34] Bloom BS, Krathwohl DR, Masia BB. Bloom Taxonomy of Educational Objectives. London: Pearson Education; 1984

[35] Raemer D, Anderson M, Cheng A, Fanning R, Nadkarni V, Savoldelli G. Research regarding debriefing as part of the learning process. Simulation in Healthcare. 2011;**6**(7):S52-S57

[36] Trinchero R. Costruire e certificare competenze con il curricolo verticale nel primo ciclo. Torino: Rizzoli Education; 2017. pp. 1-143

[37] Bot L, Gossiaux PB, Rauch CP, Tabiou S. 'Learning by doing': A teaching method for active learning in scientific graduate education. European Journal of Engineering Education. 2005;**30**(1):105-119

[38] Trinchero R. Costruire, valutare, certificare competenze. Proposte di attività per la scuola. Milano: Franco Angeli; 2012

[39] Le Boterf G. Repenser la compétence. Pour dépasser les idées reçues. 2008;**15**:978-1000

[40] Deci EL, Ryan RM. Autonomy and need satisfaction in close relationships: Relationships motivation theory. In: Weinstein N, editor. Human Motivation and Interpersonal Relationships: Theory, research, and applications. Springer Science + Business Media; 2014. pp. 53-73. DOI: 10.1007/978-94-017-8542-6_3

[41] Duangnamol T, Supnithi T, Srijuntongsiri G, Ikeda M. Computer-supported Meta-reflective learning model via mathematical word problem learning for training metacognition. Research and Practice in Technology Enhanced Learning. 2018;**13**(1):1-29

[42] Charles R. How To Evaluate Progress in Problem Solving. National Council of Teachers of Mathematics. Resto, VA; 1987. p. 22901

[43] Care E, Scoular C, Griffin P. Assessment of collaborative problem solving in education environments. Applied Measurement in Education. 2016;**29**(4):250-264

[44] Çiftci S, Bildiren A. The effect of coding courses on the cognitive abilities and problem-solving skills of preschool children. Computer Science Education. 2020;**30**(1):3-21

[45] Hufad A, Faturrohman M, Rusdiyani I. Unplugged coding activities for early childhood problem-solving skills. Jurnal Pendidikan Usia Dini. 2021;**15**(1):121-140

[46] Goldberg E, Podell K. Adaptive versus veridical decision making and the frontal lobes. Consciousness and Cognition. 1999;**8**(3):364-377

[47] Rivoltella PC. Nuovi alfabeti. Educazione e culture nella società post-mediale. Vol. 124. Brescia: Scholé-Morcelliana; 2020. pp. 5-220

[48] Aroldi P. Mediatizzazione della società, cittadinanza digitale e necessità della Media Education. In: Sala M, editor. Libro bianco Media e Minori: L'educazione ai nuovi media ai tempi del coronavirus. Soveria Mannelli: Rubbettino Editore; 2021. pp. 39-43

[49] Buckingham D. The Media Education Manifesto. New York: John Wiley & Sons; 2019

[50] Hug T. Didactics of Microlearning. Münster: Waxmann Verlag; 2007

[51] Lee E, Hannafin MJ. A design framework for enhancing engagement in student-centered learning: Own it, learn it, and share it. Educational Technology Research and Development. 2016;**64**(4):707-734

[52] Jonassen DH. Revisiting activity theory as a framework for designing student-centered learning environments. Theoretical Foundations of Learning Environments. 2000;**89**:121

[53] O'Sullivan M. Lesson observation and quality in primary education as

contextual teaching and learning processes. International Journal of Educational Development. 2006;**26**(3):246-260

[54] Pang K. Creating stimulating learning and thinking using new models of activity-based learning and metacognitive-based activities. Journal of College Teaching & Learning (TLC). 2010;**7**(4):29-38

[55] Lin X. Designing metacognitive activities. Educational Technology Research and Development. 2001;**49**(2):23-40

[56] Rivoltella PC. Media Education: Idea, metodo, ricerca. Brescia: LS La Scuola; 2019