Constructing Computational Thinking Skills for EUD Measurement: a Challenge beyond Identification

Angela Locoro[1]

[1]Dipartimento di Scienze Teoriche e Applicate (Università degli Studi dell'Insubria), Via O. Rossi, 9, 21100 Varese, Italy

Abstract

This paper starts from the definition of Computational Thinking (CT) and Computational Thinking Skills for EUD and extends it to the problem of outlining a measurement construct to assess their level of mastery in individuals. Stemming from the literature re Computational Thinking and its assessment, the idea of this contribution is to rescue this discourse from that of the K-12 Computer Science (CS) education domain, where it mainly stands, and bring it into the adult (possibly working) life, where it crosses the concepts of Computational Literacy (mastery of background knowledge) and Computational Fluency (mastery of practical application of background knowledge). These concepts are then interleaving with those of subjective abilities and problems / solutions difficulties, forming an intricate network of elements to be systematized into an *unicum* for measurement reasons. This measurement property should be able to assess the level of subjective chances to overcome an objectively difficult problem, and having many ways to solve it. The purpose of this work is to start systematizing those concepts and their reciprocal relationships that could possibly converge into a measurement construct, and to clarify which notions should remain necessarily open and unsytematized or are simply needing further investigation.

Keywords

Computational Thinking for EUD, Computational Literacy, Computational Fluency, Measurement Construct, Computational Thinking Skills Assessment

1. Introduction and Motivations

A recent study in the EUD domain posed the challenging problem of defining Computational Thinking Skills for EUD as part of a wider framework where the construct of EUDability was defined as "the degree of concreteness, modularity, structuredness, reusability, and testability fostered by a EUD environment designed for specified end-user developers, with a specified goal to be pursued in a specified context."[1, p.10]. The derived definition of Computational Thinking (CT from now on) is the following: "the acquired capability of adopting a three-stage mental process, i.e., defining the problem, solving the problem, analyzing the solution, by knowing how to apply five basic skills, i.e., abstraction, decomposition, algorithm design, generalization, and evaluation, up to an individual level of mastery."[1, p.6]. These notions were acquired in the EUD domain as part of the problem of rescuing the concepts of Computational Literacy, Computational Fluency and the like from that of K-12 education or education in general. Most of the time, attempts were made to either define these constructs or assess them through different

IS-EUD 2023: 9th International Symposium on End-User Development, 6-8 June 2023, Cagliari, Italy
angela.locoro@uninsubria.it (A. Locoro)
https://www.uninsubria.it/hpp/angela.locoro#0 (A. Locoro)
0000-0002-6740-8620 (A. Locoro)

assessment tools. The purpose of this paper is to take some of the lessons learnt from the CS education domain regarding the study of CT and of its assessment, and to transpose it to the EUD domain, where things stand quite differently in terms of what to learn and what to measure. In this domain, whenever CT is applied to problem solving, disparate solutions exist (i.e., EUD techniques). An exact mapping of some well known EUD paradigms and the related CT Skills required to adopt such EUD solutions are again part of the above study [*ibidem*].

This paper is an attempt to go beyond the identification of concepts such as CT and CT Skills, their definitions, and their mapping with EUD elements. Stemming from the literature about defining CT in K-12 education and CT assessment, we will see that the main limitations in this domain lays in that the main issue is still ontological. Put it in other words, the main concerns are still that of defining CT and its assessment. Elements of definitions include: what is included in CT; what are the abilities to be identified and assessed and what should be excluded; what tools exist to assess the acquired CT skills; how to define a standard definition and a (possibly) CT standard assessment tool.

Going beyond these aspects is crucial if we want to advance the problem of CT towards its measurement aspects. This problem has to do with the need of defining a (possibly) unique construct (or variable), with basic and advanced levels, able to account for what among the skills and problem steps devised in the main CT frameworks are the easiest, what the mid level ones, and what are the most difficult ones, in the continuum space of such variable. This approach is intended to resemble the one of standard measurement tools such as the PISA test[1] for measuring reading literacy. PISA defines reading literacy as "understanding, using, reflecting on and engaging with written texts, in order to achieve one's goals, develop one's knowledge and potential, and participate in society". In this vein, we should be able in the end to define a CT measurement construct and the related CT measurement scale. To the best of my knowledge, whatever the domain of application, we are still far from this achievement.

That said, the idea of this paper is to begin to outline what are the current achievements and what are the further steps towards a CT measurement scale.

2. Many Definitions, No Integration

Starting from the most high-level definitions of CT in K-12 education field, we may have an idea of the main pillars constituting the CT essence. For example, a very recent study in the educational domain [2] merged three definitions from the International Society for Technology in Education (ISTE), and the Computer Science Teachers Association (CSTA), and ended up by proposing the following definition: "problem-solving process that includes formulating problems, using a computer or other tools, logically organizing, analyzing, and representing data, automating solutions through algorithmic thinking, achieving efficient and effective solutions, and generalizing and transferring to other problems". Very similar considerations about what can be said of CT in terms of its essence are available in [3, 4, 5], with an indication of more or less detailed tasks, similar to the ones listed in the above definition. For example, Hsu et al. [5] released a list of thirty-one hyper-specific or micro-tasks elements, calling them

[1]More information at https://www.oecd.org/pisa/test/.

classification of CT, ranging from “Algorithmic thinking” to “Parallelization”, from “Simulation” to “Efficiency & Performance Modeling”.

However, seen from another perspective, CT may be analyzed from the point of view of its more generic forming concepts. For example, Brennan and Resnick [6] and then Yağcı [7] identified three main dimensions of CT, i,e., “computational concepts” (roughly corresponding to the Hsu micro-tasks), “computational practices” (corresponding to more or less routinely activities such as, for example, debugging and experimenting); and “computational perspectives”, regarding the more creative activities of generalization, problem-posing, and the like. A set of thirty-two items were devised for assessing these key dimensions. However, the problem with these assessment tests are double-folded: they are mainly self-assessment tests, and they treat each dimension / aspect as a separate one.

One of the most promising approaches in the direction of building a measurement construct is that of integrating / prioritizing CT dimensions, proposed by [8]. After the identification of mental processes, including the five skills that were also adopted in and adapted to the EUD-ability framework (i.e., “Abstraction”„ “Decomposition”, “Algorithm Design”, “Generalization”, and “Evaluation”), the authors propose to see CT as “an integrated ability including algorithmic thinking (e.g., formulating a sequence of steps to solve the problems), social-cooperative capacities (e.g., solving problems collaboratively), creative thinking (e.g., creatively formulating solutions) and critical thinking (e.g., thinking multi-dimensionally while working on problems).”.

This perspective lays in the direction of going beyond a single element such as the problem-solving one, but rather considering all the elements influencing people ability to solve problems. For example: their scholarly background knowledge, which can be called their “Computational Literacy”, their workplace or context of routine creation and consolidation, their experience / exposure to problem solving of the kind proposed, and to the related solutions adopted, which can be partly overlapping with the concept of “Computational Fluency”, their individual attitudes to creativity for example, and the like. In [9], the Cattell-Horn-Carroll (CHC) model of intelligence finds a place for the so called “fluid reasoning”, which is defined as: “the use of deliberate and controlled mental operations to solve novel problems that cannot be performed automatically”. These mental operations include “drawing inferences, concept formation, classification, generating and testing hypothesis, identifying relations, comprehending implications, problem solving, extrapolating, and transforming information.*Inductive and deductive reasoning are generally considered the hallmark indicators of fluid reasoning* (italic is mine).” [*ibidem*].

To sum up, as also remarked by Tikva and Tambouris [10], CT definitions are all pertaining to generic enumeration of elements included or entailed by CT, or they develop into descriptions of capabilities and skills, also including operational definitions; moreover, they may provide models of processes and individual competencies.

What we learnt from these literature on CT education? There are many aspects that should be taken into account when considering what is the property that we want to measure when we want to assess the degree or level of mastery of an individual vs. a problem whose level of difficulty has to be defined in its turn. What is it still missing from the however rich horizon of all the CT current definitions and perspective? For example, the existence of a cognitive model able to account for how hard or easy is to “abstract, generalize, evaluate”, and the like, and to apply all of the CT skills defined, based on the level of mastery of individuals and the level of difficulty of the problem at hand.

Table 1
An excerpt of CTS items as reported in the DEA (Design for EUD-Ability) methodology (and adapted from [8]).

Skills	Items (What users perceive they are able to do)
Abstraction	see the whole point, not the details see common patterns
Decomposition	decompose a problem into pieces
Algorithmic Thinking	figure a step-by-step solution figure how to execute a step
Generalization	solve problems based on one's experience apply a given solution to a new problem
Evaluation	find a correct solution think of the best solution

3. A ladder without rungs

A Computational Thinking Scale (CTS) already exists: it is composed of the five sub-scales: that related to the five CT Skills of: Abstraction, Decomposition, Algorithmic Thinking, Evaluation and Generalization. The paper about DEA methodology for the design of EUDability, part of this year edition of the IS-EUD proceedings [11], reports a Table who is based on an adaptation of the CTS scale from [8]. We report here the same table (1) to better frame the open problem posed by this assessment scale and sub-scales.

From Table 1 it is clear that each dimension or skill of the CT frameworks currently outlined are to be kept separated. We should start from here to be aware of what is missing from this kind of conceptualization: that of giving at least a hypothetical model of what are the skills that are deemed more difficult and what are the ones deemed easier. In a few words, the open problem is that of constructing a variable, in which the skills can be put in a continuum, from the easiest to the harder, so as to model where, on average, people's Computational Thinking (Literacy and/or Fluency) may be positioned. Or, said otherwise, what is the level of mastery of an individual with respect to a specific activity/skill/task. This kind of construct may be anchored for example to the perspective that each skill is somehow related to each other, and to the fact that it is cognitively impossible that one skills is mastered by an individual without having acquired the full mastery of another one, strictly related to the first one, and which has to be acquired beforehand.

In other words, we need a model of individuals' abilities and skills difficulty, upon which a measurement scale could be validated and exploited to measure the level of Computational Thinking Skills of people with respect to an EUD paradigm / technique. Without the conceptualization of this model and / or construct, we are far from the realization of an evaluation tool able to give a more realistic and as objective as possible measure of EUDability, and to generalize our discourse to that of constructing measurement scales.

4. The proposal of a tool

After this thorough analysis of existing models for CT Skills and assessment methods, I may conclude that none of the models designed for K-12 education assessment were shown to follow any rigorous methodology in their design, nor were existing models either revisited for fitting or found to reach a validation from their assessment with a CT construct and validated items.

A cognitive model of human CT learning progression would be intended to describe how skills are developed towards the highest "literacy and fluency" achievements, i.e., the ability to master the declarative and procedural knowledge of, as well as the creativity, around the topic of interest. Besides their descriptive utility, models of this kind also serve as theoretical reference for guiding in the design of measuring instruments for assessing the learning progression of individuals with the topic of interest, along the progression steps of a model that can be built upon. Constructing a model of this kind is part of a rigorous methodology to construct standard measurement scales in the CT field to measure individuals' progression with a certain topic (e.g., EUD) [12].

Following a rigorous methodology to design and implement our CT Skill model, and items for its assessment would enrich the value of the construct and of the assessment tool. In particular, we propose to adhere to the BAS workflow[2] as an "integrated approach to developing assessments that provide meaningful interpretations of individuals' work relative to the cognitive and developmental goals" [13], for the case of measuring EUD CT Skills in the EUDability framework. This methodology is aimed at validating a model quantitatively, and taking it as the base for constructing a measurement tool of the level of CT "literacy or fluency" of individuals on a measurement scale [14].

4.1. A prioritization proposal

In [15] the *Abstraction* element is the focus of the analysis. I suggest to start from this element too, in order to build a construct able to account for EUD CT Skills measurement. Abstraction is most probably the key element of CT [16], in that it is the process by which a concrete or daily object or problem, as familiar as it could be, is subject to the mental process of being ab-stracted (extracted away from its context and its continuum with other objects) and becomes the focus of a thorough inspection [17]. The abstraction process also implies to scrutinize the object from a whole perspective into an identification and de-composition of its properties [16]. This scrutiny also calls for the decomposition of object into its main parts. A second process may regard the mapping of the properties of the object or problem at hand with steps of the solution. And a third aspect of abstraction is also related to the capability of stripping away useless details and distill only the main information to manage the object and solve the problem at hand [18]. On the other hand, the abstraction process may also include the generalization step. Having abstracted an object from its context and individual details, only the meaningful patterns may be identified and mapped with similar ones [19]. This may imply finding a solution which simply consists of reusing previous solutions or being able to adapt a previous solution to the problem at hand, and the like. For this double valence of the Abstraction activity see also [20, 21].

[2]Available at https://bearcenter.berkeley.edu/page/about-bear.

All these many aspects are concerning whether and to which extent the required level of abstraction is in the ability of the person who is in charge of managing the problem in computational terms. Furthermore, one may assume that abstraction is inherent to different levels of cognitive abilities related to EUD: the level of the problem (the goal or task to be executed), the level of the object (what has to be manipulated and managed to solve the problem), and the level of the solution (the mapping of the EUD technique and its steps to be accomplished) [22, 23].

4.2. A proposal of items

Beside self-assessment questionnaires such as the CTS presented in the previous Section, specific items need to be conceived for measuring the level of abstraction that individuals can put in place when asked to solve a problem that implies the exploitation of CT Skills. For example, for the abstraction ability some items of interest could involve the request of identifying the relevant details of an object (from the computational standpoint), of focusing only on relevant details, and the capacity of reduction to the essential details. Other items should pertain the exemplification of patterns of computation, such as for example the rule-based paradigm, by proposing tasks of the “if-this-then-that” kind [24].

Other items should ascertain the level of generalization acquired by individuals with respect to some created/modified or applied solutions (e.g., some EUD technique), together with the capacity of classifying problems and / or objects under the same solution paradigm (also part of the generalization ability) or the capacity of identifying common properties and / or solutions for different objects /problems with core parts in common [19].

In this vein, abstraction should span over different activities that has the common goal of making a simplification / reduction of the problem at hand in terms of sub-problems and combination of solution rules. Seen from this perspective, abstraction seems to also include the two properties / abilities of decomposition and generalization [16]. Also, abstraction seems to precede the application of the solution, and is also prior to the application of single steps of the problem solution [25].

5. Conclusions

This paper starts from the definition of Computational Thinking skills (CT) for EUD and extends it to the problem of outlining a measurement construct to assess its level of mastery in individuals. Stemming from the literature re Computational Thinking and its assessment, it crosses the concepts of Computational Literacy (mastery of background knowledge) and Computational Fluency (mastery of practical application of background knowledge). These concepts are interleaving with individuals’ subjective abilities and with problems objective difficulties, forming an intricate network of elements to be systematized into a measurement construct. The main contribution of this paper is to define the requirements for a method and to propose a tool to find a cognitive model of CT for EUD and a construct with related items for validating such construct, towards a measurement scale. This construct should take into account how CT Skills are intrinsically related to each other and that the Computational Thinking Literacy or Fluency in an individual is not static, but rather it is evolving with her attitude, context of work and work practices. Furthermore, individuals' evolution may also pertain to the social sphere of collaboration of users in teams and small groups. Interesting investigation may also start from the concept of Abstraction, as other authors did in the CS education domain when designing CT from a learning perspective.

References

[1] B. R. Barricelli, D. Fogli, A. Locoro, Eudability: A new construct at the intersection of end-user development and computational thinking, Journal of Systems and Software 195 (2023) 111516.

[2] C. Wang, J. Shen, J. Chao, Integrating computational thinking in stem education: A literature review, International Journal of Science and Mathematics Education (2021) 1–24.

[3] J. Wing, Computational thinking, communication of the acm, n 49 (2006).

[4] T. S. Barcelos, R. Muñoz-Soto, R. Villarroel, E. Merino, I. F. Silveira, Mathematics learning through computational thinking activities: A systematic literature review., J. Univers. Comput. Sci. 24 (2018) 815–845.

[5] T.-C. Hsu, S.-C. Chang, Y.-T. Hung, How to learn and how to teach computational thinking: Suggestions based on a review of the literature, Computers & Education 126 (2018) 296–310.

[6] K. Brennan, M. Resnick, New frameworks for studying and assessing the development of computational thinking, in: Proceedings of the 2012 annual meeting of the American educational research association, Vancouver, Canada, volume 1, 2012, p. 25.

[7] M. Yağcı, A valid and reliable tool for examining computational thinking skills, Education and Information Technologies 24 (2019) 929–951.

[8] M.-J. Tsai, J.-C. Liang, C.-Y. Hsu, The computational thinking scale for computer literacy education, Journal of Educational Computing Research 59 (2021) 579–602.

[9] W. J. Schneider, K. S. McGrew, The cattell-horn-carroll model of intelligence. (2012).

[10] C. Tikva, E. Tambouris, Mapping computational thinking through programming in k-12 education: A conceptual model based on a systematic literature review, Computers & Education 162 (2021) 104083.

[11] B. R. Barricelli, D. Fogli, A. Locoro, Designing for a sustainable digital transformation: The dea methodology, in: Proc. of the 9th International Symposium on End-User Development, IS-EUD 2023, 2023.

[12] M. Wilson, Measuring progressions: Assessment structures underlying a learning progression, Journal of Research in Science Teaching: The Official Journal of the National Association for Research in Science Teaching 46 (2009) 716–730.

[13] M. Wilson, K. Sloane, From principles to practice: An embedded assessment system, Applied measurement in education 13 (2000) 181–208.

[14] M. Wilson, Constructing measures: An item response modeling approach, Routledge, 2004.

[15] N. O. Ezeamuzie, J. S. Leung, F. S. Ting, Unleashing the potential of abstraction from cloud of computational thinking: A systematic review of literature, Journal of Educational Computing Research (2021) 07356331211055379.

[16] J. Kramer, Is abstraction the key to computing?, Communications of the ACM 50 (2007) 36–42.

[17] S. Atmatzidou, S. Demetriadis, Advancing students' computational thinking skills through educational robotics: A study on age and gender relevant differences, Robotics and Autonomous Systems 75 (2016) 661–670.
[18] J. H. Hill, B. J. Houle, S. M. Merritt, A. Stix, Applying abstraction to master complexity, in: Proceedings of the 2nd international workshop on The role of abstraction in software engineering, 2008, pp. 15–21.
[19] A. C. Calderon, D. Skillicorn, A. Watt, N. Perham, A double dissociative study into the effectiveness of computational thinking, Education and Information Technologies 25 (2020) 1181–1192.
[20] O. Lindeberg, J. Eriksson, Y. Dittrich, Using metaobject protocol to implement tailoring; possibilities and problems, in: The 6th World Conference on Integrated Design & Process Technology (IDPT, volume 2, 2002.
[21] I. Cetin, E. Dubinsky, Reflective abstraction in computational thinking, The Journal of Mathematical Behavior 47 (2017) 70–80.
[22] A. Mørch, Three levels of end-user tailoring: Customization, integration, and extension, Computers and design in context 1997 (1997) 61.
[23] J. Perrenet, J. F. Groote, E. Kaasenbrood, Exploring students' understanding of the concept of algorithm: levels of abstraction, ACM SIGCSE Bulletin 37 (2005) 64–68.
[24] B. Ur, E. McManus, M. Pak Yong Ho, M. L. Littman, Practical trigger-action programming in the smart home, in: Proc. of the SIGCHI Conference on Human Factors in Computing Systems, CHI '14, Association for Computing Machinery, New York, NY, USA, 2014, p. 803–812. doi:10.1145/2556288.2557420.
[25] K. J. Newton, J. Leonard, A. Buss, C. G. Wright, J. Barnes-Johnson, Informal stem: Learning with robotics and game design in an urban context, Journal of Research on Technology in Education 52 (2020) 129–147.