Iyer, S. et al. (Eds.) (2022). Proceedings of the 30th International Conference on Computers in Education. Asia-Pacific Society for Computers in Education

Multiple Solution Pathways of Learners' Embodied Problem-solving Processes in Designing Authentic Computational Tasks

Spruha SATAVLEKAR[a*], Shitanshu MISHRA[b], Sridhar IYER[a]
[a]*Indian Institute of Technology Bombay, India* [b]*UNESCO MGIEP*
*spruhasumukh@iitb.ac.in

Abstract: When undergraduate students engage with computational thinking (CT) activities that are authentic to them, it adds not only meaning to their problem-solving actions but also a variation to their strategies and mechanisms applied during problem-solving, termed here as learners' embodied processes. Through the perspective of designing for embodied cognition, maintaining such possibilities for variation in solution pathways could be the key to making problem-solving authentic to the learners. Using the 4E cognition narratives of two undergraduate Arts learner's pathways in solving computational tasks in an authentic setting, we speculate that such multiple solution pathways need to be evaluated in pilot studies for density of significant actions during problemsolving to prioritize the actions that show spaces which require the design of embodied scaffolds.

Keywords: Computational Thinking, IoT, Embodied design, 4E cognition

1. Introduction

Initiating novice learners into the world of problem-solving using computational thinking (CT) is challenging because the factors of identity and meaningfulness of computation remain distant from the learners' real-life context (Liesaputra, V, 2020). One of the major challenges in learning computational thinking is its associated abstractness of the concepts and procedures. Visual programming environments, such as scratch, NetLogo, Greenfoot, etc., attempted to reduce the abstractness by replacing the abstract syntax of the constructs with visual block-like elements. These visual programming environments have been proven to be significantly effective in training students' CT skills (Brennan & Resnick, 2012). A similar approach has been seen in using visual programming by Bers, Flannery, Kazakoff & Sullivan, (2014) for working with robots. Programmable robots-based learning activities reduce syntax's abstractness by employing visual programming environments similar to NetLogo and Scratch (Kim & Jeon, 2007) to program their robots. The fact that the robots are physical and tangible, and students can systematically manipulate them through coding, further bridges the gap between abstract CT constructs and reality. However, programmable robots have limitations in terms of the problem-solving contexts they offer, i.e., most programmable robots-based activities employ abstract and imaginary scenarios and lack authentic real-life problem contexts (Bers, Bers, Flannery, Kazakoff & Sullivan, 2014). Programming the robot itself when tested as a way of teaching CT, is designed through a technocentric lens (Sengupta, Dickes, & Farris, 2018) where considerable time needs to be spent in the learning curve of the technology.

This leads to some of the definitions of CT in CT education that focus on not only the skills applied in problem-solving, but also on the need to identify computational aspects and computational potential in the real world (Royal Society, 2012). Constructionist approach (Harel & Papert, (1991) and Situated learning (Lave & Wenger, 1991) based methods designed to overcome this challenge created a remarkable solution for engaging such learners with support towards connecting computational aspects in their context.

Learning while being situated in authentic real-life scenarios has been argued as a better learning practice by such literature in cognitive sciences.

Traditionally, the processes of thinking and reasoning are predominantly understood using the lens of information processing theories of cognition (Pande & Chandrasekharan, 2017; Reynders et al., 2020). These approaches assert that a problem solver first engages in the extraction of information from the content embedded in the learning or task environment and then the learner performs thinking or reasoning about ‘using’ this extracted information. However, the new approaches to cognition (e.g. 4E cognition; Menary, 2010; Newen et al., 2018) and situated learning (Sentance & Humphreys, 2018)) insist that cognition and knowing cannot be separated from bodily actions and context. The embedded, embodied and enactive cognition approaches regard one's thinking and reasoning processes, actions, and the environmental elements being interacted with, as *entangled* together (Pande, 2021).

In summary, we derive from our previous work that embodied narratives and analysis of actions in context intertwined with cognition would lead to the exploration of learners’ problem-solving processes (Satavlekar, et al., 2021). Extending the previous work, we now look into the question of ‘how’ such embodied narratives can be useful from the perspective of design. We hypothesize that such exploration would be useful in identifying sites where embodied scaffolds can be designed for the novice CT learner. The situated learning and embodied cognition desirable for acquiring CT skills would also make room for reflection spots leading towards discoveries and multiple real-time problem-solving pathways, which add to the knowledge of either the technology or the CT practices for the learners.

In order to test this hypothesis, we conducted a pilot study with two undergraduate participants and provided them with IoT devices to engage in real-life computational problem-solving scenarios. Our broad research question for this study is ‘What can be the implications of novice adult learners’ problem-solving processes in embodied activity design rooted in real-life CT-based context?’ We propose the use of IoT devices and associated utility platforms to design learning activities that can help students practice CT constructs. Utility platforms such as IFTTT, Google Home, Alexa, etc. help configure the various IoT devices to work together and allow customizations based on the user's authentic problem-solving needs. Examples of possible student tasks are given below.

> ***Example:*** *Students are given a task to configure a smart light bulb such that it automatically switches on at the start of the evening and switches off before a predefined bedtime.*

In the above example, the task of configuring the IoT devices, similar to the programming activities, will require students to apply CT skills to accomplish the needed behaviors from the IoT objects. In this paper, we propose to exploit such affordances of the IoT objects and the utility platforms to make novice CT learners practice various CT constructs by programming Real-life smart devices. These objects are slowly becoming ubiquitous in many parts of the world and society. Application of such platforms that are situated in the real-world context (Lave, J., & Wenger, E., 1991) and have an impact on everyday life activities makes them more powerful as tools to think with to nurture CT skills. This paper reports the various different pathways that were found to contain significant action-cognition sequences. While doing so, we also analyze the density of such sequences and the agent or the actors involved, to speculate the implications of designing embodied scaffolds in the future iteration.

2. Methodology

The study is an open-ended qualitative pilot investigation of novice learners’ actions and cognition while solving computational thinking tasks grounded in their real-life context. The participants were chosen using convenience sampling. There were two participants, one male, and one female, (age 20 and 19 years respectively). They were second and third-year language (Sanskrit) undergraduates having no prior exposure to Computer Science in academics. These participants were chosen in order to know how adult learners who are novices to the CT domain interact with the smart programmable computational objects in their daily life and what could be challenges they face. In-person sessions were conducted over a duration of approximately 3 hours. The session included a familiarization phase and an unguided problem-solving phase as shown in figure 1. The tasks in both these phases were aligned according to the increasing order of complexity of the problem as described in the figure.

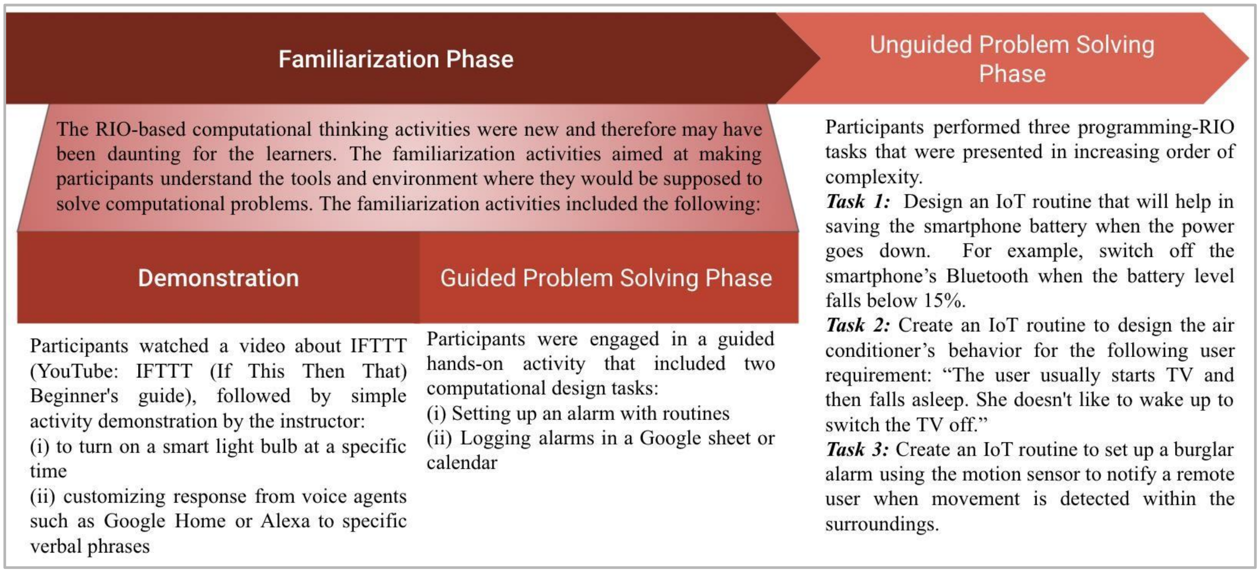

Figure 1. Phases of activities in the study (Satavlekar, et al., 2021)

Data collection and Analysis

Semi-structured interviews were administered with the objective to make the participants retrospectively reflect on their problem-solving processes. The interviews attempted to be as non–leading as possible: with questions like *"can you narrate one of the activities, how you went about solving the problem in your own words"*. However, encouraging the students to be as detailed as possible about their experience (*"Did you get bored or tired during the activities? Were there any instances which you felt interesting or could have been interesting?"*) tended to open up more information. Retrospective interviews were conducted towards the end of the study. The researcher also asked questions between the two phases, to confirm the participant's willingness to continue or abandon the next activity. In addition to the interview data, we also collected the video recording as observation data. Both the interview and video transcripts were analyzed to get a clearer picture of how the participants got familiar with the embodied environment and how they performed the given CT tasks. Using the observation and interview data from our two cases we tried to understand how the participant's engagement with the tasks in the respective sessions may have triggered CT in the participants.

Coding of the data

We performed open and focused coding of the video transcripts and identified multiple significant roles and actions corresponding to those roles associated with learners' problem-solving occurring in the unguided problem-solving phase of the session.

As seen in table 1, the roles performing as actors or agents during the problem-solving are- (i) the mentor (M), (ii) the learner (L), and (iii) the systems or application interfaces communicating through voice and visuals (S) and the combinations of two or more actors among these three. For the corresponding combination of actors, we use an abbreviated symbol as a prefix to the respective action code. The actions are inferred from the actors' problem-solving processes keeping in mind two things- (i) the list of CT skills and practices (CSTA, & ISTE., 2011) such as algorithmic thinking, logical thinking, testing, and debugging, and (ii) the response categories from the system such as showing the success of the problem being solved (RS) or expression of a constraint (C). For example, ML-DBUG specifies that the action DBUG is performed by both, the mentor and the learner, the mentor being the larger contributor. The contribution is categorized based on which actor performs as the driving anchor for that specific action. An example for clarification would be - if the learner attempts to solve a particular problem in a stepwise manner by taking the smartphone in his/her hand, we term it as L-ALGO, whereas if a mentor suggests what the learner should do next and if the learner follows or discusses upon such suggestion, we would term it as MLALGO.

Table 1. *Codes generated in data analysis showing the actors and their cognitive actions*

Encoding	Code	Meaning
Actors	M	Mentor
	L	Learner
	S	System
	LM	Learner-mentor verbal interaction
	ML	Mentor-learner verbal interaction
	LS	Learner and system verbal interaction
Actions	DSCS	Discussion
	DCMP	Problem formulation, decomposition
	XPLR	Exploring action
	ALGO	Algorithmic thinking
	LOG	Logical thinking and/ or decision making
	DBUG	Debugging actions
	ABS	Abstraction
	C	System Constraint identified or reported
	RS	Success response
	RNS	Unsuccessful response

After the encoding, we analyze the action sequences in terms of pivotal points in problem-solving - such as the problem assignment, problem decomposition, first solution, difficulty scenario faced or constraint identification, a revision (or multiple revisions) of the solution, successful final completion of the solution, and the conclusion. We identify the time spent to reach such pivot points and compare the significant actions occurring during this timespan to calculate the density of significant actions per minute. We compare the two learners' problem-solving processes in terms of the density and the learner's involvement as an actor between the particular pivot points.

3. Results

3.1 Revision and Reflection

Figure 2 shows the significant actions in context along with the learners' cognition of CT and the system in two stages: the upper part describes unguided activity 1 where the learners have to save the smartphone battery by switching off applications such as Bluetooth or reducing the smartphone brightness, etc. The first learner L1 begins with initial unguided activity and quickly designs the solution. When the mentor confirms the solution and asks the questions, *"how will you test this activity, can we check this right now"* and *"is this going to achieve what was expected in the problem statement"*, the learner faces a ***REFLECTION SPOT***, pivotal to the debugging actions ahead carried out with system application to revise the initial solution and make it more cohesive to the given problem statement. This also leads to learning about the application interfaces used for problem-solving (Satavlekar, et al., 2021) which may be useful for the learner in solving more complex problems. On the other hand. Learner 2 is motivated to ask clarification questions to come up with problem decomposition at the initial stage of problem-solving. Thus, the intrinsic motivation towards decomposing problem leads to quick revisions of own solution.

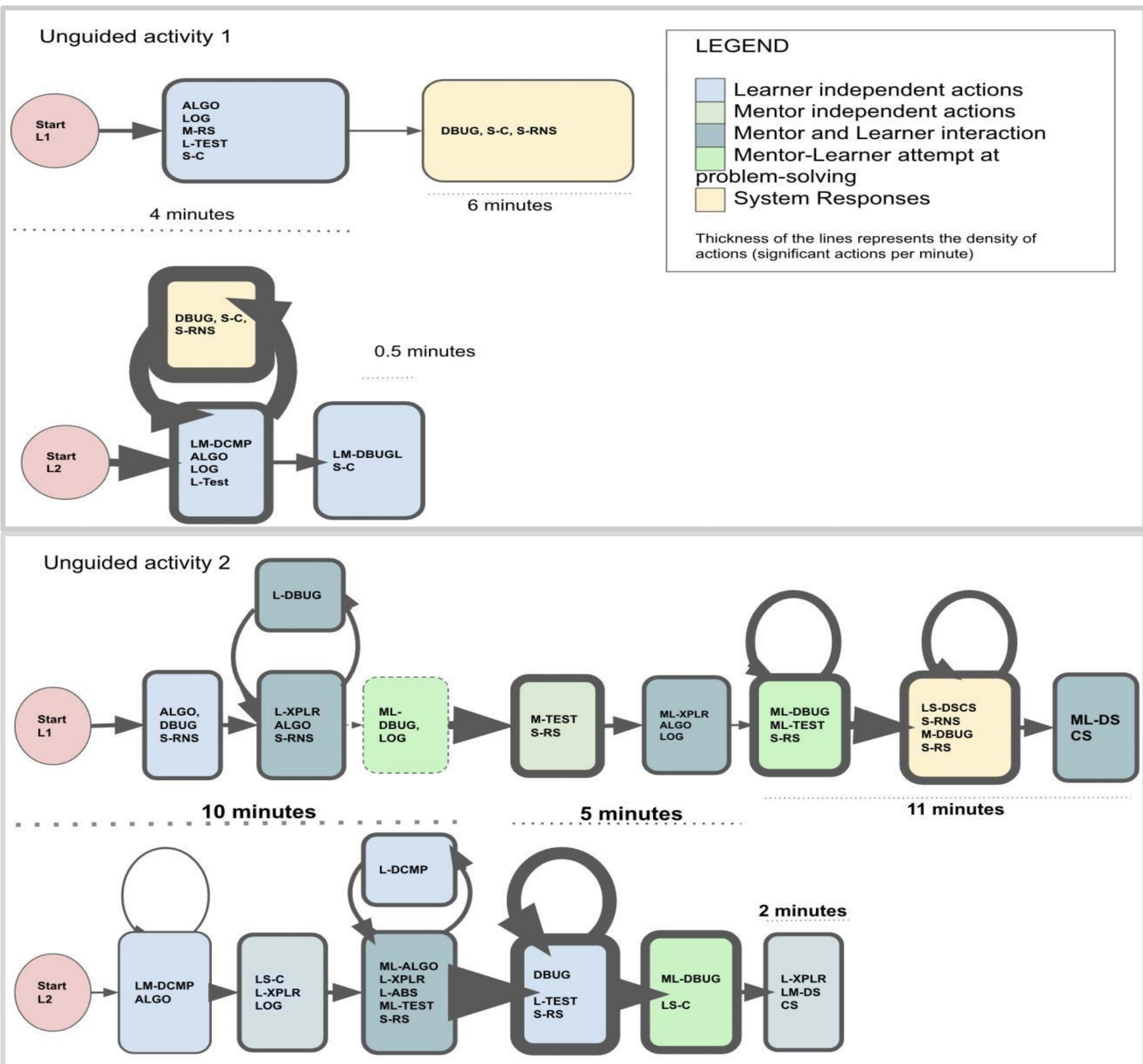

Figure 2. Learner 1 and 2's multiple pathways and density of significant actions in embodied problem solving processes of two unguided activities

3.2 Detailed Narratives: Learner 1 and 2

Analysis of a more complex activity is seen in unguided activity 2 part of figure 2, where the learner is trying to connect a motion sensor to one of the voice agents in order to program it to send an email alert when any motion is detected nearby. The learner L1 first begins with an exploration of the various smartphone applications associated with the motion sensor operation and takes approval from the mentor to try out the different applications available in the smartphone from time to time. Figuring out the proprietary application's functionality to operate the Motion sensor is relatively easy for the learner but the learner is concerned about how to integrate it with an application to set up the required routine. The second learner L2, on the other hand, begins with proactively asking clarification questions to decompose the problem and articulate the solution.

Here onwards, the learner L1 seeks a mentor's support, and together, they engage in multiple attempts to figure out which application would be suitable for the purpose, but the system shows an issue with connecting the two applications. The learner checks the manual of the motion sensor and finds Amazon Alexa can be integrated. Although the application is identified now, the learner struggles with connecting the two applications multiple times. During this entire process of thinking, following the stepwise procedure as per the procedure manual and Alexa exploration, the learner is the prominent actor operating the system application and proactively communicating with the mentor. However, as it does not yield the desirable outcome, the mentor attempts to solve the connection issue more prominently than the learner by taking control of the system application for more instances than before. Eventually, they figure out the issue and connect the two applications and the mentor shows the learner how to test and confirm that the two applications are communicating with each other and are able to detect the motion sensor. For the second learner, this link between Amazon Alexa and the Motion sensor has been established by the mentor. This allows the learner L2 to figure things out on their own without the mentor's intervention. As we can observe from the figure 2, the learner 2 spends considerably less time struggling and asking for the mentor's help in the first ten minutes. Learner 2 also stumbles upon a solution thread to the previous activity in figuring out this interface on her own, which is an important cognitive indicator.

From around 10 minutes in solving the activity, we see that the mentor is acting as the prominent support for the learner L1 in every action such as debugging, forming the revised solution by applying logical and algorithmic thinking, and helping the learner actively with the testing tasks. Whereas, for learner L2, the mentor only interferes with giving scaffolds but does not takeover the charge of the system at any point. The system application Alexa gives out a notification of motion detection sensed from the motion sensor, but the task of triggering an email remains unsolved at first for both the learners at different instances in time. The mentor assists the learners now with debugging and they finally get closer to the desired outcome. In concluding the activities, the mentor creates ***REFLECTION SPOTS*** for the learners which leads to a reflective evaluation of the solution designed by the learners as compared to the requirement of the initial problem and a discussion about the limitations of the system application. Time spent on this reflective discussion is similar (1-2 minutes) for both the learners.

Learner experience report from interview transcript excerpts

Q1. Interviewer: *"what was your experience and can you tell me say, any thing interesting you may have learnt today?"*

L1: *"Like.. exploring the apps for the first time.. Then, learning the interface or learning the features of the app.. Then third is the operating those.. With help, with guidance.. Fourth will be discovering some new operations for that. Like we did for the the motion sensor and stuff. We.. have to discover how we should link it to to Alexa.. first one. I think it would be comprehension. Like how much I would be able to comprehend with future.. This will be first test.."*

L2: "I had to find everything, right.. So it was interesting. The previous alarm and all were known like regular alarms.. These new things were completely unknown.. [Contd.] And I found one thing after another.. So it was interesting."

Q2. Interviewer: *if everything would have been already connected, do you think it would have been better to complete the tasks faster?"*

L1: *"Uhh.. for the time, when we were doing the amazon part in first half. That time I thought that ready thing would have been better.. because we have to sensor two or more times.. so that was time-consuming. But at the time we connect the motion sensor, that was new part. I mean, we have to link it through three parts.. [Contd.] So, that was just interesting part."*

Q3. Interviewer: *"Now we spent 3 hrs today. Did you feel bored? Or did you feel like 3 hrs period was appropriate?"*

L1: *"No.. I mean, first of all, this computer related stuff is interesting to me. I mean, I could perform all these tasks. So I was not bored as such. But considering time period, if we had not spent some time in the middle, we could have done more activities."*

4. Discussion

Presentation of learners' embodied processes in the context of CT-based problem-solving help us understand that it is a time-consuming process that does not follow a single straight path. The multiple pathways and struggles of the learner's cognitive processes may have been frustrating for the learner, but the answer to the interview questions Q1 and Q2 inform us that the learner was engaging in the activities with retained interest.

Takeaways from narrative

1. The prominent actors or combinations of actors vary across the whole computational thinkingbased problem-solving process. In the current analysis, this prominence is only attributed to the proactive actor in verbal communications, cognition, and actions among the three (learner, mentor, and system). No other measurement to quantify the contribution of the actors is applied here.
2. A single problem given in this context does not have a single solution, and every solution designed by the learners may not always be accurate. ***REFLECTION SPOTS*** created by the mentor for the learners play an important role in initiating the evaluation of the designed solution, attempts of identifying the limitations of the system, and also optimizing the current solution.
3. There is a need to design mentor protocol and scaffolds in such embodied problem-solving approaches involving multiple pathways and iterations such that, the learners do not become dependent upon the mentor to solve the problem. The ultimate goal of the problem-solving exercises may not be to achieve the best solution but to engage with activities that require cognitive experiences leading toward the practice of CT skills during the process. Even so, the learner's autonomy and in turn, authentic engagement may be hampered in absence of the protocol.

Speculating a takeaway from the density of significant actions

The analysis of the density of significant actions could be an important aspect while designing embodied problem-solving activities. We base this speculation on the analysis that shows how the learner L1 and the mentor had to spend significant time on activities that involved less number of computational thinking-oriented tasks in the majority part of the complex unguided activity 2 of problem-solving. We can support this speculation with the learner's answer in the first half of the Q2, that it may have been better to exclude the connection in the initial task. The learner may not desire to spend a significant amount of time on actions that are not cognitively productive for the learner. The learner may also want to reserve this time for solving more such cognitively intense tasks, as he expressed in the answer to Q3. In case of L2, having offloaded the task of connection allowed the learner to practice more productive activity of problem solving which we plan to confirm with a larger number of students in future studies.

The presented analysis of cognitive processes can also be effective in informing *WHO* is the prominent actor while dense (more significant cognitive actions in less time) are being performed. This will be useful in designing scaffolds and mentor protocol as a scaffold, such that the problem-solving activities remain authentic to the learner and the learner does not become dependent upon the mentor, as stated earlier. We end this discussion with the issue that in this study we have only analyzed two learners' actions through embodied narrative and we do not intend to make any claims about the design or learner experience because of the small sample. Our analysis of the second learner shows that the pathways followed are even more different in terms of actions of problem-solving, cognitive processes, the time taken to uncover each point of success, limitations, and revision. However, both the learners completed the activities with enthusiasm and expressed interest in solving more such complex computational problems in their semistructured interviews. Presenting an experience report and starting the discussion about a novel angle of analysis with importance to the density of significant actions in these multiple pathways is our sole intention. We need to iterate this analysis over our next study participants to provide strong support to the speculated embodied design implication. An analysis similar to or inspired by the one presented in this paper, exploring the multiple pathways of learners' embodied problem-solving processes in a real-life CTbased context may be a rich source of information to tap into to design more embodied activities for the future.

References

Bers, M. U., Flannery, L., Kazakoff, E. R., & Sullivan, A. (2014). Computational thinking and tinkering: Exploration of an early childhood robotics curriculum. *Computers & Education*, *72*, 145-157.

Brennan, K., & Resnick, M. (2012, April). New frameworks for studying and assessing the development of computational thinking. In *Proceedings of the 2012 annual meeting of the American educational research association, Vancouver, Canada* (Vol. 1, p. 25).

CSTA, & ISTE. (2011). Operational definition of computational thinking. Retrieved from https://www.iste.org/explore/Solutions/Computational- thinking-for-all Harel, I. E., & Papert, S. E. (1991). *Constructionism*. Ablex Publishing.

Kim, S. H., & Jeon, J. W. (2007, October). Programming LEGO Mindstorms NXT with visual programming. In *2007 International Conference on Control, Automation and Systems* (pp. 2468-2472). IEEE.

Lave, J., & Wenger, E. (1991). *Situated learning: Legitimate peripheral participation*. Cambridge university press.

Liesaputra, V., Ramirez-Prado, G., Barmada, B., & Song, L. (2020, February). Future-proofing Kiwi kids through the use of digital technology. In Proceedings of the 51st ACM Technical Symposium on Computer Science Education (pp. 507-513).

Menary, R. (2010). Introduction to the special issue on 4E cognition. Phenomenology and the Cognitive Sciences, 9(4), 459-463.

Newen, A., De Bruin, L., & Gallagher, S. (Eds.). (2018). *The Oxford handbook of 4E cognition*. Oxford University Press.

Pande, P. (2021). Learning and expertise with scientific external representations: an embodied and extended cognition model. *Phenomenology and the Cognitive Sciences*, *20*(3), 463-482.

Pande, P., & Chandrasekharan, S. (2017). Representational competence: towards a distributed and embodied cognition account, Studies in Science Education, 53(1), 1-43. UK: Routledge. doi: https://doi.org/10.1080/03057267.2017.1248627

Royal Society . (2012). Shut down or restart: The way forward for computing in UK schools.

Reynders, G., Lantz, J., Ruder, S. M., Stanford, C. L., & Cole, R. S. (2020). Rubrics to assess critical thinking and information processing in undergraduate STEM courses. International Journal of STEM Education, 7(1), 1-15.

Satavlekar, S., Mishra, S., Raina, A., & Iyer, S. (2021). Programming-RIO: Initiating Individuals into Computational Thinking using Real-world IoT Objects. In Proceedings of the 29th International Conference on Computers in Education. Asia-Pacific Society for Computers in Education. ICCE 2021.

Sengupta, P., Dickes, A., & Farris, A. (2018). Toward a phenomenology of computational thinking in STEM education. *Computational thinking in the STEM disciplines*, 49-72.

Sentance, S., & Humphreys, S. (2018). Understanding professional learning for computing teachers from the perspective of situated learning. Computer Science Education, 28(4), 345-370.