

LINKING COMPUTATIONAL AND MATHEMATICAL THINKING: AN INSTRUMENTAL GENESIS PERSPECTIVE IN SCRATCH-SUPPORTED LEARNING

Ayşe Nevin Türkoğlu¹, Hatice Akkoç²

¹D-Market Elektronik Hizmetler ve Tic. A.Ş.(Hepsiburada.com),

²Marmara University

This study investigates how computational thinking (CT) shapes an elementary student's instrumented techniques during Scratch-supported mathematical tasks via instrumental genesis (IG) framework. A fourth-grade student participated in task-based interviews using UCL ScratchMaths tasks. Data sources included video recordings of interviews, worksheets, and researcher field notes. Findings indicate that CT regulates emerging instrumented techniques. Analysis revealed variations in work methods, specifically resourceful and Scratch-restricted work methods. Findings also illustrate how CT-oriented tasks bridge CT and mathematical thinking (MT), enabling the construction of interconnected cognitive pathways through digital artefacts.

INTRODUCTION

In recent years, CT (Wing, 2006) has emerged as a fundamental skill that supports problem-solving processes in the curricula of many countries. CT research is based on Papert's (1980) concept of *microworlds* in which students discover mathematical ideas as designers (Hoyles et al., 2002). CT involves breaking down a complex problem into manageable parts (*decomposition*); creating a logical sequence of steps to solve these parts (*algorithmic thinking*); recognizing recurring structures and relationships (*pattern recognition*); and eliminating unnecessary details to capture the essence of a problem (*abstraction*) (Wing, 2006). Although these components overlap with mathematical reasoning and problem-solving processes (Csizmadia et al., 2015), studies in the mathematics education literature on the nature of the interaction between CT and MT are limited (e.g., Barr & Stephenson, 2011; Weintrop et al., 2016). The current studies indicate a lack of process-focused studies that explain CT–MT interaction through the tools students use and the techniques they develop (Ye et al., 2023). This gap makes the IG framework, which explains the process of transforming digital artefacts, analytically meaningful (Rabardel, 1995). This study aims to examine the *instrumented techniques* an elementary school student used in Scratch-supported mathematical tasks and how CT guided this process. Accordingly, the study focuses on the following research question: Which instrumented techniques did a fourth-grade elementary school student use when performing mathematical tasks supported by the Scratch coding artefact, and how did CT guide this process?

THEORETICAL BACKGROUND

CT is a fundamental learning skill (Wing, 2006) and the process of systematically defining variables, conditions, and loops by converting solution strategies into sequential actions (Kaufmann & Steneth, 2021). This study focuses on the CT components of *abstraction*, *algorithmic thinking*, *decomposition*, and *pattern recognition*, in the context of Scratch environment and ScratchMaths program (Benton et al., 2016). The *IG* framework provides a valuable perspective on how students demonstrate these components in a digital environment and on the relationship between their techniques and conceptual understanding (Artigue, 2002). Vygotsky's (1978) work on tool use emphasizes that learners develop their mental schemas as they interact with tools. The instrumental approach, based on this idea, centers on the distinction between an *artefact* and an *instrument* (Rabardel, 2003). An *artefact* is the object with which the user interacts, while an *instrument* consists of the cognitive structures and strategies the individual develops while working with that *artefact*. This transformative process, known as *IG*, includes two components: *instrumentation*, in which the artefact shapes the individuals' thinking, and *instrumentalization*, in which the individuals adapt the artefact to their own purposes (Trouche, 2004).

Schemes, which form the basis of *IG*, are studied through *instrumented techniques* since they cannot be directly observed (Guin & Trouche, 2002). *Instrumented techniques* are observable sequences of behaviour that reflect schemes (Lagrange, 2000). In this context, a *schema* is the integrated structure of implicit knowledge (operative invariants) that guides the user's actions, and the *instrumented techniques* are their operational counterparts in the task sequence (Vergnaud, 1996).

To examine students' ways of working with tools, Trouche (2005) defined various *work methods*, which provide insight into how students use tool resources and develop solution strategies. In the *theoretical work method*, the solution process relies on preliminary mathematical references. Reasoning is based on analogy and theoretical generalizations, while digital tools play only a limited, verifying role. In the *rational work method*, students demonstrate their mathematical mastery through inferences developed within the task context. Learners construct mathematical relationships through situation-specific reasoning rather than theoretical references, and digital tools play a secondary role in supporting this inference process. In contrast, while students' progress through memorized solutions and trial and error in the *automatistic work method*, the *Scratch-restricted work method* (originally called the *calculator-restricted work method*) relies heavily on the outputs provided by the digital tool, and mathematical references take a backseat. In the *resourceful work method*, students compare different sources using digital tools, paper-and-pencil methods, and mathematical knowledge, thereby develop flexible and creative solution strategies.

METHODOLOGY

This study is a single-case study (Yin, 2003) that examined in depth the task-related actions and strategies exhibited by a fourth-grade elementary school student in Türkiye while working on Scratch-supported mathematical tasks. The dataset consists of video recordings of task-based interviews protocol, worksheets completed by students, and field notes taken by the researcher-teacher. To ensure depth of analysis, we selected Yiğit (pseudonym) from the class in which the first author taught Scratch-supported lessons. The selection criteria were Yiğit’s experience with Scratch and achievement in mathematics. We used the following tasks in the interviews:

Table 1: Scratch tasks used in task-based interviews

Task	Instructions and explanations
Task 1	The task is to calculate the smallest integer value that must be entered into the repeat block for the sequence “pen down → move 40 steps → turn 45 degrees (counter-clockwise)” to form a regular polygon. This task has been used to examine the block selections and algorithmic steps involved in establishing the angle-repeat relationship.
Task 2	This task involves creating a tree by defining special Scratch blocks for the “Trunk” and “Top” components, then duplicating this tree at different points on the stage using the ‘repeat’ and “random position” blocks.

We analyzed the video recordings and interview transcripts using content analysis. We examined in detail how Yiğit used the Scratch artefact to complete each task (unit of analysis) and identified the instrumented techniques he demonstrated during the tasks. Instrumented techniques were considered purpose-oriented, consistent sequences of actions in which the student used the digital tool to achieve a specific mathematical goal, as defined by Trouche (2005). Accordingly, the video was divided into meaningful sections, primarily focusing on the changes observed in the student’s goals, the strategies he used, and the mathematical references he cited. We analyzed the processes the student struggled with in the identified sections, the Scratch commands he used, the visual outputs on the Scratch stage, and his verbal explanations to determine the instrumented techniques he used. The diagrams shown in Figure 1 and Figure 2 illustrate the instrumented techniques. The next stage of the analysis involved associating the identified instrumented techniques with CT components.

CT components have been used as an analytical tool to explain the role of each instrumented technique in the production of mathematical meaning. Finally, the work methods (Trouche, 2005) were interpreted by considering the instrumented techniques demonstrated by the student during the task. Work methods, including the tools used by the student during the task, the strategies developed, and the problem-solving approaches, were analyzed according to Trouche’s (2005) framework.

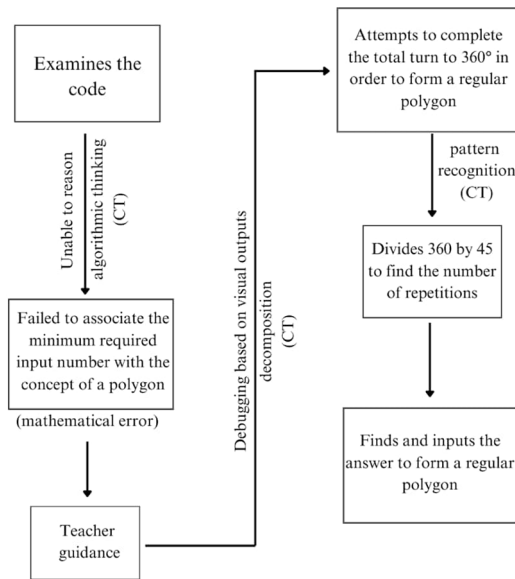
FINDINGS

This section presents the findings under two themes based on the IG approach: the instruments developed and the work methods used. Both themes explain how the CT components (abstraction, decomposition, algorithmic thinking, pattern recognition) shaped the instrumented techniques demonstrated in the tasks. Each subheading presents the two instruments that emerged from the data analysis, along with the methods Yiğit used to develop them. Findings will also show how CT components shaped the student's instrumented techniques during the task implementation process.

Findings regarding Task 1: The *Polygon* Instrument

Figure 1 below presents Yiğit's instrumented techniques for Task 1 (see Table 1) on creating regular polygons. The boxes in the diagrams represent the actions performed by the student and the associated mental steps, while the arrows represent the transitions between these steps. The explanations above the arrows indicate the CT components that stand out in the relevant transition, the researcher's interventions, or situations where the student struggled.

Figure 1. Instrumented techniques used by Yiğit (Task 1)



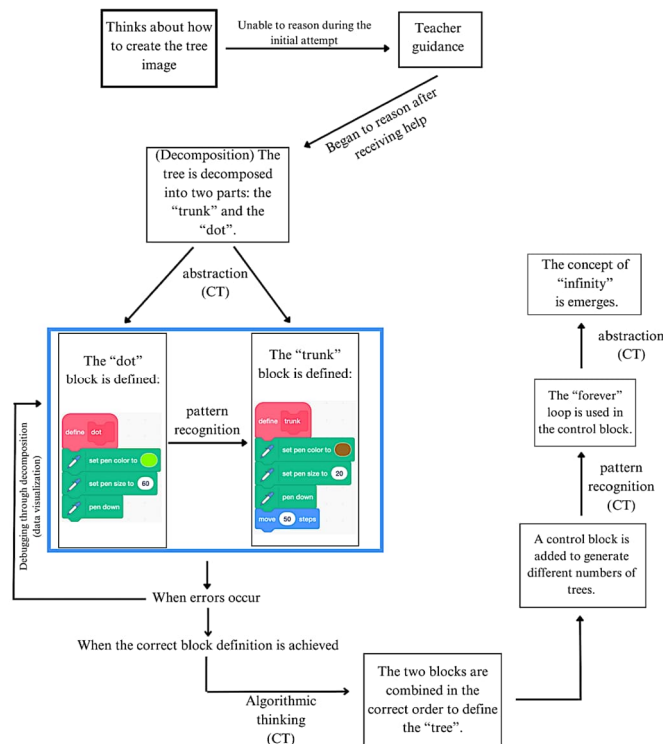
Yiğit struggled to find the number of repetitions from the algorithmic structure at the beginning of the task and therefore started the solution process by experimenting with the smallest value of the parameter. The researcher ensured he observed the output on the stage together by entering 1, then seven repetitions. Noticing that one side was missing for the value 7, Yiğit said, “We should make it 8,” and updated the number of repetitions accordingly. He observed that the number of sides was not appropriate and, using this information as a clue, he stated that a regular octagon would form after eight repetitions (Figure 1). This situation shows that Yiğit drew the polygon by identifying the error and considering the visual output, rather than by establishing the mathematical relationship between the angle and the number of loop repetitions. He realized his mathematical error by decomposing the problem and attempting different trials (Click

[here](#) to see Scratch Project). Yiğit's decision to use the number of edges generated in the Scratch output indicates that he used the *Scratch-restricted work method*.

Findings regarding Task 2: The *Infinite* Instrument

When Yiğit began creating a tree using Scratch as part of Task 2 (see Table 1), he struggled first. With the teacher's guidance, he began developing a solution by breaking the task down into smaller steps (*decomposition*). Yiğit combined special code blocks while creating the tree figure in Scratch, monitored the output, adjusted the blocks accordingly, and verified the solution with visual validation. *Pattern recognition* emerged during experimentation with the correct ratio as he entered different numbers into the "repeat" loop block (Figure 2). The most striking finding concerns the way of thinking related to the "forever" infinite loop block. Guided by the teacher, the student established the relationship "infinite repetition = countless production" and developed what he called the "infinite tree forest" instrument. (Click [here](#) to see the project).

Figure 2. Instrumented techniques used by Yiğit for the tree creation task (Task 2)



The following dialogue clearly shows the student's thoughts on this relationship:

Researcher: But can we count that number?

Yiğit: We can't count them. That's why we should do less than that.

Researcher: How will I know how many there are if I keep repeating it?

Yiğit: Well, a lot! We don't know. Then we shouldn't keep repeating it.

Finally, Yiğit identified the pattern that 10 repetitions produced 10 trees. He concluded that using an infinite loop would create an uncountable number of trees and, therefore,

should not be used. His conclusion shows that he established a generalizable relationship through interaction with the tool and identified its limitations.

Yiğit's approach aligns with the *resourceful working method* because he makes decisions based on all available resources (teacher guidance, data visualization, and debugging), rather than relying solely on mathematical references.

DISCUSSION AND CONCLUSION

This study presents evidence on how CT components guided the construction of instrumented techniques during the implementation process of two Scratch-supported tasks. The findings show that the student's work methods have a dynamic, context-dependent structure, shaped by teacher intervention and task context. Context sensitivity becomes apparent through the instrumented techniques Yiğit applied across different task contexts. Yiğit's *Scratch-restricted work method* adopted while creating polygons in Task 1 was based on Scratch's visual feedback and questioned the mathematical structure to a limited extent. On the other hand, in the context of Task 2, while creating his own tree, he adopted a *resourceful work method* by associating loops and debugging structures with numerical repetition and decomposition. These findings clearly demonstrate the context-sensitive nature of CT-MT interaction.

On the other hand, Yiğit's shift from a visually focused approach at the beginning of Task 2 to an instrumented technique based on mathematical reasoning shows that working methods can change within a single task. This situation indicates that CT components play a transitional role in the construction of mathematical concepts. Indeed, after *recognizing patterns*, Yiğit transitioned from the *Scratch-restricted work method* to the *resourceful work method* and began to produce mathematical meanings. This finding aligns with studies that point to a reciprocal relationship between CT and MT (Zeynivandnezhad et al., 2025).

As Kynigos and Grizioti (2018) point out, the process begins with the student's meaning production from digital outputs. Still, with the introduction of CT components, this meaning gradually transforms into a more rational, mathematically justifiable structure. This transformation was particularly evident when Yiğit performs Task 2. Yiğit's limited approach to the vehicle at the beginning of the task, which was a superficial trial-and-error based on visual output, guided the debugging process through decomposition. Subsequently, with the realization of *pattern recognition* and *abstraction*, his interaction with the tool became effective, and he began to use resources efficiently.

Additionally, Yiğit's reference to an "endless forest of trees" provides a concrete example of the epistemic bridge established between the loop as a CT concept and the mathematical concept of infinity. However, the emergence of such epistemic gains does not imply that the process proceeds without difficulty. Such learning processes, where CT intersects with mathematical knowledge, can create cognitive load for

students (Fang et al., 2025). This situation makes the role of teacher scaffolding critical in initiating the process and deepening conceptual understanding (Shute et al., 2017).

The literature emphasizes that the programming approach alone does not guarantee mathematical learning (e.g., Psycharis & Kallia, 2017). The findings of this study also align with this view, showing that CT does not automatically support mathematical learning. Instead, the interaction between CT and MT is closely related to the task context, the student's mathematical background, and teacher intervention. In this regard, we recommend that future research examine how teachers could promote MT in different CT-oriented task contexts and conduct an in-depth analysis of how and under what conditions students transition between different work methods.

In conclusion, this study sheds light on the issue of integrating CT into school mathematics, as raised by Weintrop et al. (2016), by making visible the processes through which students transform digital artefacts into instruments that support MT, within the IG framework. It also proposes an analytical reference point that can help teachers structure their orchestrations in CT-oriented learning environments, taking into account students' work methods and the guiding role of CT components.

Acknowledgement

This paper is part of the first author's MSc thesis entitled "An Investigation of Elementary School Students' Computational Thinking Developed in Mathematics Learning Environments Supported by Scratch From the Framework of Instrumental Genesis", completed at Marmara University, Institution of Education.

References

- Artigue, M. (2002). Learning mathematics in a CAS environment: The genesis of a reflection about instrumentation and the dialectics between technical and conceptual work. *International Journal of Computers for Mathematical Learning*, 7(3), 245-274.
- Barr, V., & Stephenson, C. (2011). Bringing computational thinking to K-12: What is involved and what is the role of the computer science education community? *ACM Inroads*, 2(1), 48-54.
- Benton, L., Hoyles, C., Kalas, I., & Noss, R. (2016, February). *Building mathematical knowledge with programming: Insights from the ScratchMaths project*. Suksapattana F.
- Csizmadia, A., Curzon, P., Dorling, M., Humphreys, S., Ng, T., Selby, C., & Woollard, J. (2015). *Computational thinking: A guide for teachers*. <https://eprints.soton.ac.uk/424545/>
- Fang, X., Ng, D. T. K., & Yuen, M. (2025). Effects of Geogebra-enhanced Scratch computational thinking instruction on fifth-grade students' motivation, anxiety, cognitive load. *Education and Information Technologies*, 30(1), 377-402.
- Guin, D., & Trouche, L. (2002). Mastering by the teacher of the instrumental genesis in CAS environments: Necessity of instrumental orchestrations. *Zentralblatt für Didaktik der Mathematik*, 34(5), 204-211.

- Hoyles, C., Noss, R., & Adamson, R. (2002). Rethinking the microworld idea. *Journal of Educational Computing Research*, 27(1), 29-53.
- Kaufmann, O. T., & Stenseth, B. (2021). Programming in mathematics education. *Int. Journal of Mathematical Education in Science and Technology*, 52(7), 1029-1048.
- Kynigos, C., & Grizioti, M. (2018). Programming approaches to computational thinking: Integrating turtle geometry, dynamic manipulation, and 3D space. *Informatics in Education*, 17(2), 321-340.
- Lagrange, J. B. (2000). L'intégration d'instruments informatiques dans l'enseignement: une approche par les techniques. *Educational Studies in Mathematics*, 43(1), 1-30.
- Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. Basic Books.
- Psycharis, S., & Kallia, M. (2017). The effects of computer programming on high school students' reasoning skills and mathematical self-efficacy and problem solving. *Instructional Science*, 45(5), 583-602.
- Rabardel, P. (1995). *Les hommes et les technologies : Approche cognitive des instruments contemporains*. Armand Colin.
- Rabardel, P. (2003). From artefact to instrument. *Interacting with Computers*, 15(5), 641-645.
- Shute, V. J., Sun, C., & Asbell-Clarke, J. (2017). Demystifying computational thinking. *Educational Research Review*, 22, 142-158.
- Trouche, L. (2004). Managing the complexity of human/machine interactions in computerized learning environments: Guiding students' command process through instrumental orchestrations. *Int. J. of Computers for Math. Learning*, 9(3), 281-307.
- Trouche, L. (2005). Instrumental genesis, individual and social aspects. In D. Guin, K. Ruthven, & L. Trouche (Eds.), *The didactical challenge of symbolic calculators: Turning a computational device into a mathematical instrument* (pp. 197-230). Springer.
- Vergnaud, G. (1996). Au fond de l'apprentissage, la conceptualisation. In *Actes de l'école d'été de didactique des mathématiques* (pp. 174-185).
- Vygotsky, L. (1978). Interaction between learning and development. In M. Gauvain & M. Cole (Eds.), *Readings on the development of children* (pp. 34-40).
- Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2016). Defining computational thinking for mathematics and science classrooms. *Journal of Science Education and Technology*, 25(1), 127-147.
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33-35.
- Ye, H., Liang, B., Ng, O. L., & Chai, C. S. (2023). Integration of computational thinking in K-12 mathematics education: A systematic review on CT-based mathematics instruction and student learning. *International Journal of STEM Education*, 10(1), Article 3.
- Yin, R. K. (2003). *Case study research: Design and methods* (3rd ed.). Sage.
- Zeynivandnezhad, F., Fernández, R. E., & Binti Ismail, Z. (2025). Fostering mathematical thinking through a computer algebra system in a differential equation course. *International Electronic Journal of Mathematics Education*, 20(3), Article em0826.