Applying the Mathematical Task Framework to K-8 Computing

Michael Deutsch
Department of Integrated Studies in Education
McGill University
Montreal, Quebec, Canada
michael.deutsch@mail.mcgill.ca

ABSTRACT

This theoretical work in progress introduces the Mathematical Task Framework (MTF) to computing educators and makes the case for its application to primary and middle grade computing (CSK8). Early work is described in which the MTF is synthesized with prior computing work, and some familiar programming tasks are evaluated through it. Extensions of this work are then discussed.

CCS CONCEPTS

• **Social and professional topics → K-12 education**.

KEYWORDS

K-12, computing pedagogy, programming, teaching methods, task design, cognition, framework

ACM Reference Format:
Michael Deutsch. 2023. Applying the Mathematical Task Framework to K-8 Computing. In *The 18th WiPSCE Conference on Primary and Secondary Computing Education Research (WiPSCE '23), September 27–29, 2023, Cambridge, United Kingdom.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3605468.3609753

1 INTRODUCTION

As primary and middle grade computing (CSK8) grows around the world, instructional tasks continue to proliferate. A CSK8 educator today has access to tasks published by national, state, or local bodies; tasks created by educational companies and not-for-profit organizations; and tasks they create or adapt themselves. Many types of CSK8 tasks are now familiar to educators: unplugged activities, code reading and prediction exercises, Parsons problems, faded examples, step-by-step programming tutorials, and open-ended projects, among others. CSK8 teachers have many options.

Dengel and Gehrlein [4] have found, though, that it can be unclear, or at least a matter of opinion, which task type is apt for the conceptual or procedural learning goals at hand. We might expect that programming-education taxonomies and frameworks such as SOLO [1, 7], Matrix [6], and CCCP [5] would help. Unfortunately such frameworks have been found difficult to use or impractical for the job of instructional design or teaching [6, 8, 9].

WiPSCE '23, September 27–29, 2023, Cambridge, United Kingdom

ACM ISBN 979-8-4007-0851-0/23/09.
https://doi.org/10.1145/3605468.3609753

1.1 A task framework for CSK8

This raises questions for us in CSK8 as we design and select learning tasks. Through our tasks are we in fact prompting the thinking and learning we describe in our objectives? Are we demanding more than students can handle, or perhaps not enough? Are there task types or specific task elements that, by their design, dependably generate certain demand? The present study (in progress) aims to answer these and related questions, by capturing the cognitive demand of CSK8 tasks in concrete, recognizable classroom terms.

2 MATHEMATICAL TASK FRAMEWORK

Mathematics education provides a useful starting point: the Mathematical Task Framework (MTF) from Stein and Lane [10]. Part of the MTF [10, p. 58] is presented here in condensed form: four kinds of task demand, from least to most demanding.

Memorization. "Committing facts, rules, or definitions to memory," or reproducing those that were previously learned. Such tasks contain little ambiguity and have "little or no connection to concepts or meaning."

Procedures *Without* Connections to Underlying Concepts. Using a "well-rehearsed algorithm" to complete a problem "with no attention to why or how the algorithm works" and with "limited, if any, connection to underlying mathematical ideas." Such tasks require "little cognitive effort" because the algorithm is either "specifically modeled" prior to being given or "its use is evident" thanks to explicit instruction or the context in which it appears.

Procedures *With* Connections to Underlying Concepts. Using a procedure in a way that "maintains and/or develops deep levels of understanding of mathematical concepts and ideas." Students may "follow a suggested pathway through the problem," but that pathway is "broad" and contains "close connections to underlying conceptual ideas." This is contrasted with following a "narrow algorithm that is opaque with respect to underlying concepts."

Doing Mathematics. Applying "complex, non-algorithmic thinking" to a task where "there is not a predictable, well-rehearsed approach or pathway explicitly suggested by the task, task instructions, or a worked out example." Such tasks can be "likened to the processes in which mathematicians engage when solving problems."

3 APPLICATION TO CSK8

From this we can envision a CSK8 Task Framework, adapted to primary computing. Allowing for words that mathematics and computing use differently (e.g., algorithm and procedure), the task characteristics described in the MTF are familiar in CSK8.

A *Memorization* task comprises simple knowledge; it might be recalling syntax or interface elements.

A *Procedures Without Connections* task requires work of some kind, but contains little ambiguity about what is to be done. A student may be able to complete it by mimicking what they have been shown, without creative or conceptual thought. Tasks like this might include a code puzzle with a well-rehearsed or obvious answer, or copying code from a teacher, classmate, or tutorial.

A *Procedures With Connections* task requires the student to apply conceptual programming knowledge. The general approach may be clear, but it takes effort and understanding to complete and may call for creativity, adaptation, or extension. "Modify" and "remix" tasks fit here; reading and comprehending code too, suggesting that "passive" tasks like code tracing, predicting, explaining, and debugging are as cognitively demanding as writing new code. Note that this level does not entail working code; pseudocode counts.

Finally, a *Doing Computing* task is the most demanding, requiring the student to work out their own strategy from what they understand of the concepts, procedures, and constraints at hand.

Importantly, these categories are posed without judgment or inherent pedagogical value. Each type of demand has its appropriate use. The key is the ability to distinguish them. With a framework such as this, an educator is better equipped to accurately assess the learning opportunity that any given task affords [2, 11] and weigh its suitability for the context at hand.

3.1 Assessing a sample task

As a first application, the task shown here is from the Hour of Code catalog at Code.org [3]. This is step 5 in a sequence of 20 exercises of increasing complexity. The learner is given code that produces a geometric pattern. The challenge is to correctly fill in two missing parameters. It is visually engaging; the instructions are clear. What kind of thinking does it demand?

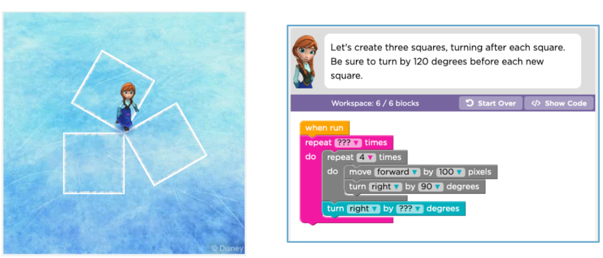

Figure 1: Block coding task: what level of demand?

When done in sequence, where each puzzle closely resembles the ones that came before it—and the correct values appear in the instructions—a learner can complete this from visual cues or habit. Only shallow tracing may be needed. This is type (2), Procedures Without Connections. But if a teacher deploys this same task alone or out of sequence, tracing is very much required: type (3), Procedures With Connections. If the completed pattern were supplied with no code, only instructions to "Create this pattern," then this task might be (3) or even (4) Doing Computing. The same program may be scaffolded in different ways, resulting in different levels and types of demand.

4 FUTURE WORK AND DISCUSSION

The present paper outlines a starting point for a CSK8 task framework. To fully realize the framework I will carry out a qualitative synthesis of the MTF with prior work in computing education. After drafting the framework I will validate it using case studies of published CSK8 tasks, expanding on the example used here. Educators will see the variety of forms a "demanding" task may take, and the levels of demand at which a single task may be implemented.

It can be expected that the CSK8 task framework will contribute to CSK8 curriculum design and professional development, as the MTF has done for mathematics [2, 11]. It may recast our understanding of what constitutes "rich" or "real" computing, and open up new ways of implementing familiar tasks. It may also provide structure for instructional coaching, helping CSK8 teachers recognize and maintain rigorous engagement.

ACKNOWLEDGMENTS

Annie Savard and Michelle Friend, for valuable feedback.

REFERENCES

[1] John Biggs and Kevin Collis. 1989. Towards a Model of School-based Curriculum Development and Assessment Using the SOLO Taxonomy. *Australian Journal of Education* 33, 2 (Aug 1989), 151–163. https://doi.org/10.1177/168781408903300205

[2] Melissa D. Boston and Anne Garrison Wilhelm. 2017. Middle School Mathematics Instruction in Instructionally Focused Urban Districts. *Urban Education* 52, 7 (2017), 829–861. https://doi.org/10.1177/0042085915574528

[3] Code.org. 2015. Code with Anna and Elsa. https://studio.code.org/s/frozen/lessons/1/levels/5. Accessed: (20 June 2023).

[4] Andreas Dengel and Rupert Gehrlein. 2022. Comparing Teachers' and Preservice Teachers' Opinions on Teaching Methods in Computer Science Education. In *Proceedings of the 17th Workshop in Primary and Secondary Computing Education* (Morschach, Switzerland) *(WiPSCE '22)*. Association for Computing Machinery, New York, NY, USA, Article 13, 4 pages. https://doi.org/10.1145/3556787.3556866

[5] Rodrigo Duran, Juha Sorva, and Sofia Leite. 2018. Towards an Analysis of Program Complexity From a Cognitive Perspective. In *Proceedings of the 2018 ACM Conference on International Computing Education Research* (Espoo, Finland) *(ICER '18)*. Association for Computing Machinery, New York, NY, USA, 21–30. https://doi.org/10.1145/3230977.3230986

[6] Ursula Fuller, Colin G. Johnson, Tuukka Ahoniemi, Diana Cukierman, Isidoro Hernán-Losada, Jana Jackova, Essi Lahtinen, Tracy L. Lewis, Donna McGee Thompson, Charles Riedesel, and Errol Thompson. 2007. Developing a Computer Science-Specific Learning Taxonomy. In *Working Group Reports on ITiCSE on Innovation and Technology in Computer Science Education* (Dundee, Scotland) *(ITiCSE-WGR '07)*. Association for Computing Machinery, New York, NY, USA, 152–170. https://doi.org/10.1145/1345443.1345438

[7] Raymond Lister, Beth Simon, Errol Thompson, Jacqueline L. Whalley, and Christine Prasad. 2006. Not Seeing the Forest for the Trees: Novice Programmers and the SOLO Taxonomy. *SIGCSE Bull.* 38, 3 (jun 2006), 118–122. https://doi.org/10.1145/1140123.1140157

[8] Susana Masapanta-Carrion and J. Angel Velazquez-Iturbide. 2022. Replication of an Evaluation of Teacher Training in the Classification of Programming Exercises Using Bloom's Taxonomy. In *2022 IEEE Global Engineering Education Conference (EDUCON)*. IEEE, Tunis, Tunisia, 1839–1848. https://doi.org/10.1109/EDUCON52537.2022.9766707

[9] Karolína Miková and Jakub Krcho. 2023. Cognitive Taxonomy and Task Gradation in Educational Robotics – Preliminary Results. In *ROBOT2022: Fifth Iberian Robotics Conference*, Danilo Tardioli, Vicente Matellán, Guillermo Heredia, Manuel F. Silva, and Lino Marques (Eds.), Vol. 589. Springer International Publishing, Cham, 575–585. https://doi.org/10.1007/978-3-031-21065-5_47

[10] Mary Kay Stein and Suzanne Lane. 1996. Instructional Tasks and the Development of Student Capacity to Think and Reason: An Analysis of the Relationship between Teaching and Learning in a Reform Mathematics Project. *Educational Research and Evaluation* 2, 1 (Jan 1996), 50–80. https://doi.org/10.1080/1380361960020103

[11] Mary Kay Stein, Margaret Schwan Smith, Marjorie A. Henningsen, and Edward A. Silver. 2009. *Implementing Standards-Based Math Instruction: A Casebook for Professional Development* (2nd ed.). Teachers College Press, New York.