Applying the Mathematical Task Framework to K-8 Computing

Michael Deutsch
MA thesis (WIP)

McGill | Faculty of Education

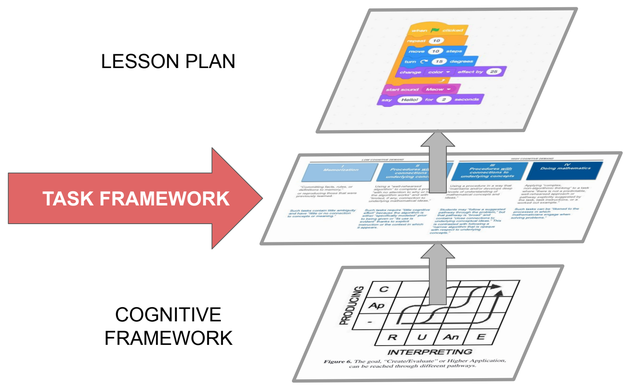
Context

Or: "How demanding is this CS task, *really?*

As K-8 CS tasks proliferate, it's often unclear – or a matter of opinion – how demanding CS tasks are. **Are we placing appropriate demands** on students, spurring the **thinking & doing** that we want to see?

Research-based cognitive taxonomies are informative, but have not proven **practical** for K-8 **curriculum development & teaching**.

The Mathematical Task Framework (Stein & Lane, 1996) provides a useful reference point for CSK8: a **cognitive framework,** expressed in terms of **descriptive features** that apply to many familiar task types.



RQ1

What elements make a K-8 CS task cognitively demanding?

A: Begin with the MTF

Condensed from the Mathematical Task Framework (Stein & Lane, 1996, pp. 58-59):

LOW COGNITIVE DEMAND		HIGH COGNITIVE DEMAND	
I ***Memorization***	***II*** ***Steps <u>without</u> connections to the underlying concepts***	***III*** ***Steps <u>with</u> connections to the underlying concepts***	***IV*** ***Doing mathematics***
"Committing facts, rules, or definitions to memory," or reproducing those that were previously learned.	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."	Using a procedure in a way that "maintains and/or develops deep levels of understanding of mathematical concepts and ideas."	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 contain little <u>ambiguity</u> and have "little or no connection to <u>concepts</u> or meaning."	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.	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."	Such tasks can be "likened to the processes in which mathematicians engage when solving problems."

B: Synthesize with CS Ed Research

MTF synthesized with relevant CSER work on cognition and programming tasks:

SOLO Taxonomy in programming
Lister et al (2006)

Matrix Taxonomy
Fuller, et al (2007)

Abstraction Transition Taxonomy
Cutts, et al (2012)

Cognitive Complexity of Programs
Duran, Sorva, Leite (2018)

Block Model for comprehension
Schulte (2008)

RQ2

Which demands do we see in common CSK8 tasks?

C: Analyze CSK8 tasks through the synthesized framework

LOW COGNITIVE DEMAND		HIGH COGNITIVE DEMAND	
I ***Memorization***	***II*** ***<u>Following</u> steps***	***III*** ***<u>Understanding</u> and <u>creating new</u> steps***	***IV*** ***Doing computing***
Place familiar CSK8 tasks on the Computing Task Framework:			

Early findings toward a CSK8 task framework centred on cognitive demand:

- Keys to cognitive demand in a task include: <u>procedures</u>; <u>concepts</u>; <u>strategy</u>; <u>explicitness</u> vs <u>ambiguity</u>/<u>latitude</u>.
- Demand lies in both design and implementation. The instructions and context matter.
- Each level/type of demand has appropriate uses. A given task can be implemented with *different* demands.
- One can reach High Cognitive Demand (III, IV) with "passive" tasks or pseudocode (e.g.).

Selected References

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.

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.

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.

Quintin Cutts, Sarah Esper, Marlena Fecho, Stephen R. Foster, and Beth Simon. 2012. The Abstraction Transition Taxonomy: Developing Desired Learning Outcomes through the Lens of Situated Cognition. In Proceedings of the Ninth Annual International Conference on International Computing Education Research (Auckland, New Zealand) (ICER '12). Association for Computing Machinery, New York, NY, USA, 63–70.

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.

Carsten Schulte. 2008. Block Model: An Educational Model of Program Comprehension as a Tool for a Scholarly Approach to Teaching. In Proceedings of the Fourth International Workshop on Computing Education Research (Sydney, Australia) (ICER '08). Association for Computing Machinery, New York, NY, USA, 149–160.

Acknowledgements

Special thanks to:

Prof Annie Savard
McGill University

Prof Michelle Friend
University of Nebraska Omaha