Features of 'Authentic' Programming-based Mathematical Tasks

Chantal Buteau[1], Joyce Mgombelo [1], Eric Muller[1], Abolfazl Rafiepour[2], Ana Isabel Sacristán[3]

[1]Brock University, Canada, cbuteau@brocku.ca; [2]Shahid Bahonar University of Kerman, Iran; [3]Cinvestav, Mexico

Policy makers, teachers, and scholars have expressed widespread interest in the integration of programming into school curricula, or more broadly "computational thinking", now considered a 21st century skill. In this poster we analyse a sequence of programming-based mathematics tasks found in an undergraduate course series taught since 2001. We identify 4 features of 'authentic' programming-based tasks, i.e., of tasks that aim at empowering students to engage in programming-based mathematical work 'as mathematicians would do'.

Keywords: programming, task design, computational thinking.

INTRODUCTION

There is a resurgence of interest in integrating programming—more broadly, computational thinking (CT)—in education (e.g. in the UK: Benton et al., 2016), which many argue reflects the number of scientific fields that have developed a computational counterpart (Weintrop et al., 2016), and the rise of a 21st century skill and need for proficiency in computational practices. Our interest is in CT curriculum development and task design that would equip students with skills and competencies to address this need, particularly in relation to programming for math learning. Weintrop et al. (2016) argue for 'authentic' computational tasks in math and science classrooms and provide a taxonomy of computational practices that students would use in such tasks.

This poster discusses design features of tasks, implemented annually for more than 15 years, in a sequence of three undergraduate programming-based mathematics courses named *Mathematics Integrated with Computers and Applications* (MICA) offered at Brock University (Canada). Based on a case study examining a student's learning experience through her 14 project tasks during the sequence of the three MICA courses (Buteau et al., 2016), we argue that such courses develop students' proficiency in CT engagement for mathematics. Furthermore, these 14 tasks (accounting for 70-80% of students' final grades) afford students to engage in 'authentic' computational practices for mathematics (Broley et al., 2017).

CONCEPTUAL FRAMEWORK & METHODOLOGY

The conceptual framework draws from various interrelated concepts reflected in the literature on CT, CT in mathematics, and CT in mathematics education (Buteau et al., 2018). Wing (2014) defines CT as "the thought processes involved in formulating a problem and expressing its solution(s) in such a way that a computer —human or machine—can effectively carry out" (p. 5), whereby computer programming is an underlying activity. In the field of mathematics education, CT has a 45-year legacy

that started with LOGO programming language (Papert, 1980). In our work we view students' learning of mathematics through CT-based mathematical activities with Lave and Wenger's (1991) concept of "legitimate peripheral participation," which describes how learners enter into a community of practice and gradually take up its practices. The focus of this proposal is on the features of 'authentic' tasks in which students (newcomers) engage peripherally in CT for mathematics practices - as mathematicians (elders) would do. We use affordances of CT for mathematics learning (Gadanidis et al., 2017) to guide our analysis by exploring the relevance of such affordances in a mathematician's work, and then examine the 14 programming-based mathematics tasks from the MICA courses used in Buteau et al.'s (2016) study in order to identify common task features as 'authentic' CT-based mathematics tasks.

FOUR 'AUTHENTIC' PROGRAMMING-BASED MATH TASK FEATURES

The resulting 4 task features identified in our analysis are: i) involves **mathematics that cannot be done by hand**; ii) involves **(dynamic) visualization**; iii) **should lead to conjecturing/exploring of unknown mathematics** (to the student) or to interpreting mathematics applications; and iv) should be **meaningful to students**.

REFERENCES

Benton, L., Hoyles, C., Kalas, I., & Noss, R. (2017). Bridging primary programming and mathematics: Some findings of design research in England. *DEME*, *3*(2), 115-138.

Broley, L., Buteau, C., & Muller, E. (2017). (Legitimate peripheral) computational thinking in mathematics. *Proceedings of CERME* (pp. 2515-23), Dublin, Ireland.

Buteau, C., Muller, E., Marshall, N., Sacristán, A. I., & Mgombelo, J. (2016). Undergraduate mathematics students appropriating programming as a tool for modelling, simulation, and visualization: A case study. *DEME*, *2*(2), 142-156.

Buteau, C., Muller, E., Mgombelo, J., & Sacristán, A. (2018). Computational thinking in university mathematics education: A theoretical framework. *Proceedings of RUME*, San Diego, CA.

Gadanidis, G., Hughes, J. M., Minniti, L., & White, B. J. G. (2017). Computational thinking, Grade 1 students and the Binomial Theorem. *DEME*, *3*(2), 77-96.

Lave, J., & Wenger, E. (1991). *Situated learning: Legitimate peripheral participation*. New York, NY: Cambridge University Press.

Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. New York, NY: Basic Books.

Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2016). Defining computational thinking for mathematics and science classrooms. *Journal for Science Education and Technology*, *25*, 127-147.

Wing, J. M. (2014). Computational thinking benefits society. *Social Issues in Computing*. Retrieved from http://socialissues.cs.toronto.edu/index.html%3Fp=279.html