University students learning programming-based practices for mathematical inquiry: Contributions of an institutional approach

Laura Broley[1], Chantal Buteau[1] and Joyce Mgombelo[1]

[1]Brock University, Canada; lbroley@brocku.ca

In mathematics education research, the use of different theoretical lenses can lead to a deeper understanding of phenomena such as the teaching and learning of programming for mathematical inquiry at university. In light of our past work leveraging the instrumental approach, this paper seeks to explore the potential contributions of a different lens: the Anthropological Theory of the Didactic (ATD). Through a comparative analysis of "practices to be learned" in a mathematical inquiry project and "practices actually learned" by one student, we demonstrate the usefulness of the ATD's notion of praxeology. The complementarity of the analysis with our past work calls for further reflection on the networking of institutional and instrumental approaches.

Keywords: Institutional approach, praxeology, mathematical inquiry, computer programming.

Introduction

Researchers working on the networking of theories in mathematics education claim that using different theoretical lenses can lead to a deeper and more complex understanding of a phenomenon of interest (Bikner-Ahsbahs & Prediger, 2014). In our work, we are interested in the teaching and learning of programming for conducting mathematical inquiry at the university level. So far, our research team has utilized the instrumental approach and demonstrated the usefulness of several of its tools: e.g., the notion of scheme for understanding individual students' learning over time (e.g., Buteau et al., 2019) and the notion of instrumental orchestration for exploring how instructors create a learning environment to support students' learning (e.g., Buteau et al., 2020). In this paper, our aim is to see what additional understandings could be gained about our phenomenon of interest when using a different theoretical lens: namely, the Anthropological Theory of the Didactic (ATD). More specifically, the research question guiding our work is: How can the theoretical tools offered by the ATD contribute to our understanding of students learning to use programming for mathematical inquiry at university?

The selection of the ATD was inspired in part by the fact that it has been used by the first author in several past works (e.g., Broley et al., 2018, which reports on mathematicians' use of programming in research and teaching). The ATD is also becoming increasingly used by researchers of university mathematics education; however, as far as we know, it has not yet been used in research about students learning to use programming for mathematical inquiry. Moreover, overviews of the use of different theoretical lenses to investigate university mathematics education have highlighted the potential complementarity of the ATD and the instrumental approach (e.g., Gueudet et al., 2014; Winsløw et al., 2014). As such, we see our work as providing both an empirical and a theoretical contribution. In particular, this paper is a natural first step towards a deeper reflection on the networking of the ATD and the instrumental approach, for the purposes of better understanding how university students learn to use programming for mathematical inquiry.

Theoretical framework

The ATD was initiated by Yves Chevallard in the 1980s and has since grown to include a significant collection of theoretical tools for investigating mathematical and didactic activities. In this paper – the beginning of our work on the question posed above – we will consider only a subset of these tools.

Central to the collection of tools offered by the ATD is the notion of *praxeology* (Chevallard, 1999), which provides a way of describing the practices (i.e., regularized and purposeful human actions) that are involved in any human activity. According to the notion, every practice is composed of four interrelated, essential components: a type of task (generating the need for the practice), techniques (ways of doing the types of task), technologies (discourses producing, justifying and explaining the techniques) and theories (the rational discourses underlying the technologies). Hence, practices necessarily comprise both a practical part (the types of tasks and techniques, called the *praxis*) and a theoretical part (the technologies and theories, called the *logos*).

Another critical tool in the ATD is the notion of *institution* (Chevallard, 1991): a relatively stable structural element of a society that frames and promotes certain kinds of human actions (towards the achievement of certain aims). Indeed, a fundamental idea behind the ATD is that the praxeologies of an individual (e.g., a student) do not exist in a vacuum, but are shaped by the social institutions (e.g., universities) where they are developed. Artigue (2016) explains that with the ATD:

> The lens is no longer directed towards the student and her cognitive functioning or development, but towards the institutional practices that condition and constrain, both explicitly and implicitly, what she has the possibility to learn or not. (p. 17)

Bosch and Gascón (2014) add: "an ATD analysis therefore starts by approaching *institutional praxeologies* and then referring individual behavior to them, talking in terms of the 'praxeological equipment' of a given person" (p. 69).

A third critical tool in the ATD is the notion of *didactic transposition* (Chevallard, 1991), which highlights the institutional relativity of praxeologies with respect to three institutions that are pertinent to thinking about university mathematics education: professional communities (which produce and use "professional practices"), an education system (which, through programs, curricula, course outlines, textbooks, etc., determines "practices to be taught") and a classroom (where interactions between teachers and students determine "practices to be learned" – e.g., through the assessments given in the course, and "practices actually taught and learned").

In this paper, we propose to use the notion of praxeology to describe and analyze practices that are involved in the activity of "using programming for conducing mathematical inquiry". Since we are interested in "students learning" to engage in this activity, we start by working at the level of a classroom institution, embedded in a particular education system (described in the next section). Following an institutional approach, we start by examining institutional praxeologies (in particular, practices to be learned, inferred from assignment guidelines and anticipated solution approaches), which may shape the practices that students have the possibility to learn. We then use this as a base to investigate the potential praxeological equipment of (or practices actually learned by) students.

Context and methods

The study we present is part of a larger 5-year (2017-21) iterative design, non-interventional research that uses as a context three programming-based math courses at Brock University: Mathematics Integrated with Computers and Applications (MICA) I, II and III. In these courses, students design, program and use computer environments to investigate mathematical concepts, conjectures and real-world applications (Buteau & Muller, 2010) – i.e., to engage in mathematical inquiry.

We focus on MICA II, the second course in the sequence. Certain features of the education system (Brock's MICA concentration) ensure that the structure and operation of MICA II is relatively stable. With MICA I as a prerequisite, MICA II students are expected to have developed some practices for using a programming language to solve some foundational types of tasks (e.g., produce the graph of a given function); in MICA II, the aim is for students to use and build on these practices to engage in mathematics inquiry projects. Course outlines specify the evaluations: 4 mini-projects (worth 12% each), 1 final project (worth 22%) and 2 midterms (worth 15% each). Each week during a semester, students participate in 2-hour lectures (where the professor mainly introduces math content related to the projects) and 2-hour labs (where the students can work on their projects among their peers and with access to teaching assistants and/or the professor). In a certain semester, the professor giving the MICA II course determines the topics of the projects and hence the particular practices to be learned.

The current study used data collected from one MICA II classroom, when the course was given in 2019. The data is of two types: (1) guidelines for the 4 mini-projects and the final project (i.e., 5 assignments), which can be found in Ralph (2020); and (2) semi-structured interviews that were conducted with volunteer MICA II students shortly after they completed each of their assignments, which aimed to guide the students in reliving and describing their actions. In alignment with these two data types, our study proceeded in two stages. First, we constructed a reference epistemological model (Bosch & Gascón, 2014) of practices to be learned in the first MICA II assignment: i.e., practices that students may (be expected to) learn when engaging in the assignment. We modelled types of tasks and techniques by looking at the formulation of the assignment questions, considering the kinds of objects involved, and thinking about anticipated solution approaches of students, based on our understanding of the MICA courses (as researchers, instructors, and/or past students) and the mathematics involved in the assignment. We modelled technologies and theories by thinking about mathematical justifications for the modelled techniques. Note that our model does not necessarily reflect the exact material presented in lectures (we did not have access to that data for the current study) or the intentions of the professor (they were not interviewed for our study). Moreover, we do not claim that our model is absolute or comprehensive: It contains elements that helped us as researchers begin to explore what students may learn when engaging in the assignment. Second, we explored the praxeological equipment exhibited by one MICA II student, Mark, during the interview that followed assignment 1. To accomplish this, Mark's interview was coded to find evidence of his perceptions in relation to the different components of our reference model. These perceptions were recorded in "praxeology tables", with evidence sorted in rows according to whether it corresponded to types of tasks, techniques or technologies (as is typical of analyses of students' praxeologies, we did not find evidence specific to the level of theory). It is important to note that the interviews were not designed for the purpose of probing into Mark's perceptions in relation to the praxeologies in our reference model; and yet, Mark spontaneously provided evidence of his perceptions. This is part of the reason he was selected for the current study, to allow us to explore potential new directions for our work in terms of theoretical lenses and data analysis. It is also important to note that Mark may not be representative of all MICA students. He is a computer science and mathematics co-major who, unlike many of his peers, had significant programming experience prior to the MICA I course.

Results

A reference epistemological model of the practices to be learned in assignment 1

In this section, we present our reference model of the practices to be learned in assignment 1, using the notion of praxeology. Assignment 1 contains 5 questions (Q1, Q2, Q3, Q4, Q5), which invite students to work on and explore the scope of a statistical computational technique ("Monte Carlo") for "estimating numerical values" (we note that this is a *genre* of task, which is more general than a type of task; Chevallard, 1999). All students are required to do Q1-3 and then can choose either Q4 or Q5. For the complete assignment, including complete question statements, see Ralph (2020). In this paper, we focus on Q1, Q3 and Q5, due to space constraints, and since these were the ones for which Mark presented explicit evidence of his related praxeological equipment in his interview.

In lectures prior to assignment 1, students are introduced to the Buffon needle problem, including an analytical solution (a derivation of a formula) and a computational solution (the writing of a code) for the task: find/estimate the probability that a needle touches a line if it has length $l = 1$ and it is dropped onto a plane of parallel lines that are $d = 1$ unit apart. Crucial to these solutions is an initial modelling of the situation, which transforms the original task into a new one: find/estimate the area in $[0,\pi]$ x $[0,0.5]$ such that $y \leq (1/2)\sin(x)$. In Q1, students modify the code (and model) given in class to create a program (Figure 1) to "find" the probability if the length of the needle is changed to 0.5. Generally speaking, the task solved in Q1 (like the task solved in class) belongs to the type of task T_1: Estimate the probability of an event in a random experiment. However, the modelling of the random experiment leads to a task of another type: T_2, Estimate the volume of a bounded k-dimensional subset A. In Q3, students create a program to solve another task of this type: i.e., estimate the hypervolume of the unit hypersphere in R^4 ($k = 4$, $A = \{(x,y,z,w) |\ x^2+y^2+z^2+w^2 \leq 1\}$).

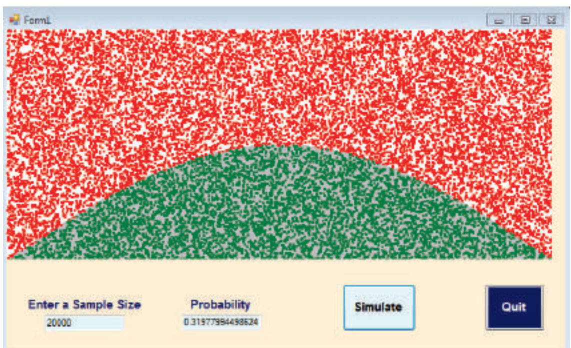

Figure 1: A student's program for solving Q1

We could model the Monte Carlo technique that can be used to solve T_2 as: τ_2, put A in a set B whose volume is known (e.g., B is the hypercube in Q3), choose n points at random in B (with n sufficiently large), keep track of the number of points m that are in A and calculate m/n*Area(B) (the estimate). One main technology underlying τ_2 is: θ_2, as the number of points increases, the estimation becomes more accurate. This is a particular instance of the "law of large numbers", which is supported by the theory: Θ, Probability and Statistics.

For MICA II students, a computational version of τ_2 is implemented in the vb.net programming language typically through modifying the original code given in class: studying the code and making relevant modifications may support students in abstracting the technique. In the tasks faced by students, B is provided (e.g., Q3 specifies to carry out the estimation by "choosing n points at random inside $[-1,1]^4$"). Moreover, the program students create enables them to vary n until it is judged to be sufficiently large to produce an accurate estimate; and in Q1 and Q3, this judgement can be made by comparing the output (an estimation) with a *known* actual value. Such a comparison may support students in developing θ_2 (though it would not be possible when trying to estimate an *unknown* value).

An additional part of Q3 introduces another way to judge the accuracy of an estimate. Students are invited to estimate the hypervolume accurate to one decimal place, which leads to a modification of $[T_2, \tau_2, \theta_2]$. To estimate the volume to a certain accuracy (T_3), students build on τ_2 to implement a new technique, τ_3: they create a program that carries out the estimation process described by τ_2 a specified number of times (w) and calculates the mean and standard deviation of the estimates (the mean is the new potential estimate), and then they vary n and w (increasing them) until a sufficiently small standard deviation is obtained (e.g., taking into consideration that 99.7% of all means would lie within three steps of the standard deviation from the mean). In this case, one main technology underlying the technique is: θ_3, a specified version of the "central limit theorem", which ensures an approximate normal distribution for a large number of independent, identically distributed variables.

Finally, Q5 introduces students to another praxeology based on a widened theory, including Calculus and Analysis. In particular, Q5 asks students to: suppose that two numbers a and b are chosen at random from $\{1, 2, \ldots, n\}$; let P_n be the probability that a and b are relatively prime; and answer: As n goes to infinity, does the limit of P_n exist? Can you guess the exact limit? The type of task explored here could be modelled as: T_4, find the limit of a sequence $(P_n)_n$, where the sequence values are the probability of an event of a random experiment. The technique τ_4 has two parts: students go back to engaging in a task of type T_1 to estimate a sequence value P_n for a given n using Monte Carlo techniques; then they vary the value of n, for larger and larger values, to see if P_n appears to approach a certain value. Underlying this technique are technologies, θ_4, related to the estimation of limits: e.g., if the values seem to get closer to a specific value as n increases, this may be the limit of the sequence.

Mark's praxeological equipment related to assignment 1

In this section, we use perceptions shared by Mark when describing his actions in completing assignment 1 to think about his praxeological equipment with respect to the reference model outlined above. Tables 1, 2 and 3 provide some selected quotes from Mark's interviews, which serve to exemplify his perceptions in relation to the three main praxeologies in our model.

Although there are some imprecisions in Mark's descriptions, they suggest a praxeological equipment that reflects well the techniques and technologies in our reference model. Consider, for example, the way Mark describes the technique for estimating the volume of the hypersphere (Table 1): he explains the existence of two embedded spaces, the dropping of random points and the required check for points that hit (or miss) the smaller space. He also justifies the technique by referring to the "large amount of points" he dropped, suggesting an awareness that the accuracy improves as the number of points increases. In a similar vein, Mark's perceptions of the practice for estimating the volume to a certain accuracy (Table 2) includes some key elements of the technique (the replication of the process of dropping points and the aim of "a really small" standard deviation) and the technology (the idea that the desired accuracy should be maintained several standard deviations from the mean since "over 99%" of means lie in that range). Finally, Mark's perceptions of finding the limit in Q5 (Table 3) reflect a technique using "trial and error of inputs" with bigger and bigger numbers, supported by a technology of the sort: if the outputs get "closer" to a value, the limit exists (and that's the limit).

Table 1: Mark's perceptions in relation to [T_2, τ_2, θ_2]

<table>
<tr><td>T_2</td><td rowspan="2">"the code was similar to question one in the fact that you have um, a space, uh contained within a, a larger space, and you're dropping uh, random points and you're … simply checking uh if you're hitting or missing this smaller space, uh to estimate the volume of it"</td></tr>
<tr><td>τ_2</td></tr>
<tr><td>θ_2</td><td>"because I was able to … run a simulation where a large amount of points … are dropped, uh that gets a fairly high accuracy obviously because there's so many data points"
"because it's on computers you're able to do it so many times"</td></tr>
</table>

Table 2: Mark's perceptions in relation to [T_3, τ_3, θ_3]

<table>
<tr><td>T_3</td><td>"once the program was done I had to switch over into math mode to … prove that the 4.9 was accurate"</td></tr>
<tr><td>τ_3</td><td rowspan="2">"I was able to run a simulation where a large amount of points … are dropped … and then that's also replicated … I was able to do I believe almost 100,000 or so points, um, nearly 200 times. Uh so because of this you're able to get a, um, fairly accurate mean value of uh 4.9 um, … the main thing was that you're able to get a really small standard deviation of 0.002 or whatever, um, so even three or four standard deviation points off you're still accurate to that 4.9 decimal place which um, obviously means that over 99% of your trials show up that it's going to be that 4.9 accuracy"</td></tr>
<tr><td>θ_3</td></tr>
</table>

Table 3: Mark's perceptions in relation to [T_4, τ_4, θ_4]

<table>
<tr><td>T_4</td><td>"you're looking for basically a probability as Pn approaches infinity"</td></tr>
<tr><td>τ_4</td><td rowspan="2">"what I ended up … doing was … there's just a small table that kind of shows … the outcomes that um I did so that way I could show um as the numbers got bigger it did get closer to that probability."
"it was more so … trial and error of inputs to uh, try and prove that uh, that the limit did exist, so um, it was uh trying to find a match between incredibly large numbers that you know, can quote 'prove infinity', uh versus finding numbers that would make you sit there through the entire lab waiting for an output"</td></tr>
<tr><td>θ_4</td></tr>
</table>

Interestingly, Mark does not refer to specific mathematical theoretical elements such as the "law of large numbers" or "central limit theorem"; in fact, he indicates that some technologies are "obvious" (Tables 1 and 2), which could be indicative of a "non-mathematical" theory (i.e., an explanation of a technology that is not based on relevant mathematical properties). Also, Mark's perceptions highlight other kinds of technologies – specific to computational techniques – that were not included in our reference model and that relate to both the affordances of computers (e.g., they allow you to repeat mathematical processes many times; Table 1) and the constraints (e.g., even computers have a limit as to how many times they can repeat certain mathematical processes; Table 3).

Towards the end of his interview, Mark gives his view of the key idea behind assignment 1:

Mark: I would say this one the key kind of concept was finding exact, um, exact values through estimation … it was definitely a cool thing so, when you're building your program, and you're like "Alright I'm just going to throw in some really big numbers and we're going to get really close to this number that's mathematically correct".

Mark seems to perceive a genre of task that differs slightly from the one in our reference model, emphasizing the estimation of *known exact* numerical values (or numbers that are "mathematically correct"). In each of Q1, Q3 and Q5, Mark found an exact answer (e.g., using Google) to compare with the estimate produced by his computer program. It would be interesting to see how his praxeological equipment might differ when the numerical value being estimated is unknown.

Conclusions

In this paper, we sought to explore how the theoretical tools offered by the ATD can contribute to our understanding of university students learning to use programming for mathematical inquiry. We claim that our analysis demonstrates the potential usefulness of the notion of praxeology: e.g., it allows us to describe and reflect on specific mathematical practices students may have the possibility to learn while (and for) engaging in programming-based mathematical inquiry projects, and to investigate the degree to which certain students learn those practices or not.

We note that the praxeologies presented in this paper are not specific to a particular programming environment. Although we see the development of such general praxeologies as a pertinent aim of teaching students to use programming for mathematical inquiry, the question remains as to how the specificities of an environment could shape students' praxeological equipment. One limitation of our study is that it used existing data from interviews that were not framed by the ATD. Further investigation into students' praxeological equipment would require a revision of existing research tools and a reflection on the usefulness of other data sources (e.g., students' responses to midterms).

In relation to our past work using the instrumental approach, we see the analysis presented in this paper as complementary. For instance, using the notion of instrumented action schemes, we have so far focused on modeling operational knowledge that is developed and used across MICA inquiry projects, primarily when students program a computer environment for the purposes of their inquiry (e.g., Buteau et al., 2019). In comparison, using the notion of institutionalized praxeologies, we were brought to focus on modeling mathematical knowledge that is developed and used in particular MICA projects, primarily at stages outside of the programming: e.g., when students use their programmed computer environment to conduct the inquiry. The current study seems to open a window into other key parts of students' learning. Such complementarity warrants further reflection not only on the potential contributions of the ATD, but also on its networking with the instrumental approach.

Acknowledgment

This work is funded by SSHRC (#435-2017-0367) and received ethics clearance (REB #17-088).

References

Artigue, M. (2016). Mathematics education research at university level: Achievements and challenges. In E. Nardi, C. Winsløw, & T. Hausberger (Eds.), *Proceedings of the first conference* *of the International Network for Didactic Research in University Mathematics* (pp. 11–27). University of Montpellier and INDRUM.

Bikner-Ahsbahs, A., & Prediger, S. (Eds.) (2014). *Networking of theories as a research practice in mathematics education*. Springer International Publishing.

Bosch, M., & Gascón, J. (2014). Chapter 5: Introduction to the anthropological theory of the didactic (ATD). In A. Bikner-Ahsbahs, & S. Prediger (Eds.), *Networking of theories as a research practice in mathematics education* (pp. 67–83). Springer International Publishing.

Broley, L., Caron, F., & Saint-Aubin, Y. (2018). Levels of programming in mathematical research and university mathematics education. *International Journal of Research in Undergraduate Mathematics Education*, *4*(1), 38–55. https://doi.org/10.1007/s40753-017-0066-1

Buteau, C., Gueudet, G., Muller, E., Mgombelo, J., & Sacristán, A. I. (2019). University students turning computer programming into an instrument for "authentic" mathematical work. *International Journal of Mathematical Education in Science and Technology, 57*(7), 1020–1041. https://doi.org/10.1080/0020739X.2019.1648892

Buteau, C., & Muller, E. (2010). Student development process of designing and implementing exploratory and learning objects. In V. Durand-Guerrier, S. Soury-Lavergne, & F. Arzarello (Eds.), *Proceedings of the sixth congress of the European Society for research in mathematics education* (pp. 1111–1120). Services des publications, INRP.

Buteau, C., Muller, E., Santacruz Rodriguez, M., Gueudet, G., Mgombelo, J., & Sacristán, A. (2020). Instrumental orchestration of using programming for mathematics investigations. In A. Donevska-Todorova, E. Faggiano, J. Trgalova, Z. Lavicza, R. Weindhal, A. Clark-Wilson, & H-G. Weigand (Eds.), *Proceedings of the tenth ERME topic conference (ETC 10) on mathematics education in the digital age (MEDA)* (pp. 443–450). ERME.

Chevallard, Y. (1991). *La transposition didactique: Du savoir savant au savoir enseigné* (2nd ed.). La Pensée Sauvage.

Chevallard, Y. (1999). L'analyse des pratiques enseignantes en théorie anthropologique du didactique. *Recherches en didactique des mathématiques*, *19*(2), 221–265.

Gueudet, G., Buteau, C., Mesa, V., & Misfeldt, M. (2014). Instrumental and documentational approaches: From technology use to documentation systems in university mathematics education. *Research in Mathematics Education, 16*(2), 139–155. https://doi.org/10.1080/14794802.2014.918349

Ralph, B. (2020, August 10). *Mathematics integrated with computers and applications II* [Assignment guidelines]. Computational thinking in university mathematics. https://ctuniversitymath.files.wordpress.com/2020/08/mica-ii-assignment.pdf

Winsløw, C., Barquero, B., De Vleeschouwer, M., & Hardy, N. (2014). An institutional approach to university mathematics education: From dual vector spaces to questioning the world. *Research in Mathematics Education, 16*(2), 95–111. https://doi.org/10.1080/14794802.2014.918345