Instrumental orchestration of using programming for mathematics investigations

Chantal Buteau[1], Eric Muller[1], Marisol Santacruz Rodriguez[2], Ghislaine Gueudet[3], Joyce Mgombelo[4], and Ana Isabel Sacristán[5]

[1]Brock University, Department of Mathematics and Statistics, Canada, cbuteau@brocku.ca; [2]Universidad del Valle, Colombia; [3]Université de Bretagne Occidentale, France; [4]Brock University, Canada; [5]Cinvestav, Mexico

We investigate how the instrumental orchestration can contribute to our understanding of the teaching to university students of using programming technology for mathematical investigation projects. Our case study highlights a dual role the instructors play, as policy maker and as teacher, to orchestrate students' instrumental geneses, and the integration of projects as a key element of the exploitation mode.

Keywords: instrumental orchestration, schemes, programming, university education.

INTRODUCTION AND CONTEXT

In the field of mathematics education, programming for learning has a legacy of half a century that started with the designing of the LOGO programming language for learning (Papert, 1972). Studies working in this area have largely focussed on learning, whereas pedagogical design was mainly tangentially analysed (e.g., Noss & Hoyles, 1992). However, with the recent increased integration of programming in schools and curricula, we see a crucial need for studies about teaching and teacher education concerning programming such as that of Benton et al. (2018).

In this paper, we address this need under the fourth theme of the conference, namely *Theoretical perspectives and methodologies/approaches for researching mathematics education*, by presenting a preliminary study concerning the theoretical contribution of the instrumental orchestration (Trouche, 2004) to analyse the teaching to university students of using programming for mathematical investigation projects.

Our study is part of a five-year naturalistic research that takes place in the context of a sequence of three university mathematics courses, called *Mathematics Integrated with Computers and Applications* (MICA) I-II-III taught at Brock University since 2001. In these project-based courses, math majors and future math teachers learn to design, program, and use interactive environments to investigate mathematics concepts, conjectures, and applications (Buteau et al., 2015). The research aims at understanding how students learn to use programming for 'authentic' mathematical investigations, if and how their use is sustained over time, and how instructors support that learning.

The question guiding the study presented in this paper is: *What do we learn about the teaching of using programming for authentic mathematical investigations by using the theoretical frame of the instrumental orchestration, considering programming as an artefact?* Building on previous work on students' instrumental genesis of using programming (Buteau et al., 2019a) and on constructionist facets of a related teaching

(Buteau et al., 2019b), this study focuses on teaching aspects that aim at steering students' instrumental geneses. Next, we present the instrumental approach, and how we use it when the artefact is programming. We then present our methods, and illustrate the use of instrumental orchestration by analysing the case of the MICA II teaching. Finally, we discuss insights gained from using the instrumental approach.

INSTRUMENTAL APPROACH: SCHEMES, GENESIS, ORCHESTRATION

The instrumental genesis approach (Rabardel, 1995) provides a lens to describe how a student, in an activity with a math goal, learns to use an *artefact* (e.g. programming) and learns mathematics at the same time, through the development of schemes. It introduces a distinction between an artefact, which is produced by humans, for a goal-directed human activity, and an *instrument*, developed by a subject along his/her activity with this artefact for a given goal through a process called *instrumental genesis*.

The instrument is composed by a part of the artefact and a scheme of its use (Vergnaud 1998), either as a *usage scheme*–"oriented towards the management of the artefact"– or as an *instrumented action scheme*– "oriented to the carrying out of a specific task" (Trouche, 2004, p.287). In math education, the instrumental approach was first used to study learning processes of secondary school students using calculators (Guin et al. 2005). These studies used a detailed definition of schemes based on Vergnaud's work. Namely, a *scheme*, defined as a stable organization of the subject's activity for a given goal, comprises four components: i) the *goal* of the activity; ii) *rules-of-action* (RoA), generating the behaviour according to the features of the situation; iii) *operational invariants*: concepts-in-action and theorems-in-action (TiA), which are propositions considered as true and governing the RoAs; and iv) possibilities of *inferences*.

Trouche (2004) added that students' instrumental genesis may need to be guided by a teacher and proposed the concept of *instrumental orchestration*. He explains that the instrumental orchestration refers to the teacher's intentional organization, arrangement and didactic use of various artefacts in the class (including digital ones), with the purpose of steering the students' instrumental genesis. Drijvers et al. (2010) added the idea of didactical performance to explain the different adjustments, which are made in response to the events of the class. These authors (2010, p. 215) summarize the resulting three elements of the instrumental orchestration: i) *Didactical configuration*: "an arrangement of artefacts in the environment, or, in other words, a configuration of the teaching setting and the artefacts involved in it"; ii) *Exploitation mode*: "the way the teacher decides to exploit a didactical configuration for the benefit of his or her didactical intentions [it] includes decisions on the way a task is introduced and worked through, on the possible roles of the artefacts to be played"; and iii) *Didactical performance*: "involves the ad hoc decisions taken while teaching on how to actually perform in the chosen didactic configuration and exploitation mode." We next describe how we use the instrumental approach when considering programming as the artefact.

Programming for Mathematics Investigations: Students' Instrumental Genesis

The general goal of the students' activity is to investigate a complex situation, combining mathematical knowledge and programming. By using programming for this goal, students develop an instrument, associating some aspects of programming (artefact) and schemes of use for specific sub-goals such as those described in the development process model (see Fig.1). For example, the *scheme of articulating a mathematics process in the programming language*, as a sub-scheme of the *scheme of designing and programming an object* (Step 3 in Fig.1). Thus a student's instrumental genesis in this context means that a student develops a complex web of schemes which ramifications include, among others, those in Fig.1 (Buteau et al. 2019a).

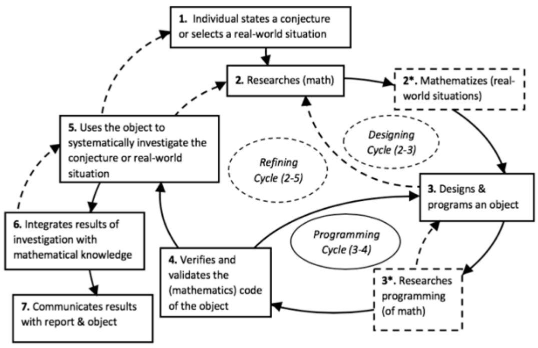

Figure 1. Development process (DP) model of a student engaging in programming for a pure or applied mathematical investigation (Buteau et al., 2019a).

METHOD

We investigate the teaching of using programming for authentic mathematical investigations using the case of the instrumental orchestration of one MICA II instructor, Bill, during Winter 2019. Bill is a mathematician, a remarkable teacher with long teaching experience (over 30 years) who played a key role in the development and teaching of MICA courses–he is not researcher in this project. Data included all course material and semi-structured task-based interviews with Bill for each of the 4 assigned and 1 original math investigation projects. Data also included a 2000 departmental program description document when the MICA courses were adopted. For this paper, we use only Bill's first project assignment for which he chose 4 short investigation problems involving Monte Carlo integration (we discuss Bill's choice further below): P1-Buffon Needle problem; P2-area between two curves; P3-hypervolume of the unit hyper-sphere in R^4; and either P4- Buffon-Laplace problem or P5-the infinite limit of the probability that two randomly selected integers smaller than n are relatively prime. Bill's interview and project guidelines were analyzed by identifying potential students' schemes that might have, implicitly or explicitly, intentionally been promoted by Bill. In this study, we did not directly observe Bill's teaching. Only a part of his teaching was accessible through the course material he produced. This is a limitation of our naturalistic research, but we also collected student data from Bill's class allowing to relate shortly their instrumental genesis with Bill's orchestration. Next, we outline each orchestration component of a MICA instructor using our understanding of the teaching of MICA courses (e.g. Buteau et al., 2019b), and illustrating them using Bill's teaching.

INSTRUMENTAL ORCHESTRATION OF PROGRAMING: A CASE STUDY

Didactical Configuration

The choice of programming technology use in MICA courses and the teaching setting configuration was established in 2000 by the mathematics department (including Bill) at Brock University, and has since remained. The teaching format involves 2 hrs of lecture (in a regular lecture room) and 2 hrs of computer lab (1 computer per student), weekly. In regards to the artefact, there is an agreement among the instructors that MICA I-II courses mainly use vb.net language with Visual Studio. There are now two MICA III courses; one for math and science majors moving on to C++ programming language with GNU IDE, and the new MICA III course for future mathematics teachers using vb.net, Scratch and Python with Jupyter Notebook. Specifically, in the case of Bill's teaching in 2019 of MICA II, he decided to also introduce Excel technology as part of his second assignment. Bill justifies: "I also think… that every math major has to be able to use Excel, because this is one of the standard tools in the outside world." (B.A2.143)

Exploitation Mode

The main didactical intention grounding how programming technology would be integrated into a sequence of three MICA courses was also established in 2000 by the mathematics department: students would learn to exploit programming for mathematical work. The 2000 departmental document stipulates:

> [Students] will confront problems from pure and applied mathematics that require experimental and heuristic approaches. In dealing with such problems, students will be expected to develop their own strategies and make their own choices about the best combination of mathematics and computing required in finding solutions.

Also, the core of each MICA course was to be pure and applied programming-based mathematics investigation projects that account for 70-80% of a student's final MICA course grade. This is a key element of the exploitation mode (here again, the 'instructor' is viewed as a 'policy maker faculty' rather than a 'teacher'): Through these projects, the department appears to thus intend that students engage in the process described in Figure 1 (Buteau et al., 2019a). During lectures, the instructor introduces students to mathematics that is needed for the assigned individual mathematics investigation projects which are worked on during the labs. We interpret such projects as aiming at developing and/or re-enforcing various students' schemes, such as the *scheme of articulating a math process in the programming language*. Since the instructor chooses the topic and direction of these mathematics investigations, and communicates it through detailed guidelines, we interpret such projects to aim at students developing their web of schemes associated mainly to steps 3 to 7 of the DP model (Fig.1). Each MICA course also involves a final original project, in lieu of a final exam where students work individually or in pair, and choose a topic of their own and the direction of the mathematics investigation. Such final projects can be viewed as an intention for students to develop further or mobilize their complete web of schemes including those associated to steps 1 and 2. For the individual MICA instructor (as a 'teacher'), the ways s/he decides to exploit the didactical configuration in order to meet the didactical intentions envisioned and decided by the department, include decisions about the mathematics content and related investigation projects. It also includes decisions about the ways the content is developed in lectures and synchronized with the investigation project work in the labs. In terms of the choice of mathematics content in MICA II course, the instructors over the years have selected various topics and areas of mathematics relevant to a computational approach for investigations, often according to their own, evolving mathematics interests and research (Buteau et al., 2019b). For example, Bill comments on the computational relevance of P5 investigation and his personal interest in this area:

> [P5] generated a lot of great discussion… I love the idea… I like analytic number theory, I love the idea that, um, there are patterns… I jump up and down about that with the [students]... I can sell this assignment to my students. (B.A1.174)

We associate Bill's choice of 'relevant topic' to Step 1 (Fig.1), and interpret it as an implicit guidance to students (to develop a scheme of) identifying when a programming approach is an added value for the work, such as for math that cannot be done by hand. Using various resources (including their own research), the MICA II instructor *designs* programming-based mathematics investigation projects and *develops guidelines aligned with both* the planned lecture content and planned guidance in lab as student work through the projects. For example, Bill *designs* MICA II project assignments by 'playing on the computer with some math' and decides on parts of an assignment (and guidelines) by thinking on the potential difficulties that MICA II students may confront e.g. when programming the mathematics (Steps 3-5 in DP model). Bill mentions:

> I actually tried many many many things before we got the formula that you have here and I would try something and I'd say "That's too hard, that comes too fast, this has to go, this has to be sequenced differently." (B.A1.43)

We also interpret Bill's expectations from students to be able to mobilize usage schemes of programming in vb.net (Step 3) developed in MICA I. He says e.g. that P1 is "to keep them calm" as "there are no new programming tricks..it's all review". In P1 he gives students a code to build from. By providing the students with a "well written piece of code", Bill says that it "helps them review …proper coding practice"; e.g. "how to change from math coordinates to graph coordinates… separately and clearly;" etc. Bill requires students to submit, for P3, a print out of their code rather than the program; he says "I'm telling them I'm going to actually read the code on the page, it sends that signal" (B.A1.154). We interpret it as Bill's intention to students' developing their scheme of coding with rule-of-action 'I write codes according to standards'. As for the *project guidelines,* they outline the topic to be investigated, within the mathematics context developed (i.e. *synchronized) in the lectures*, together with some details of the investigation design (such as input and output) sometimes complemented with some partial code (as for P1 mentioned above). We associate these respectively to Steps 1, 2, and 3 in the DP model. For example, Bill says:

> the idea that we can take a real-world situation, and we can distill from that the mathematics, and then take that mathematics and write a simulation based on that mathematics… I'm thinking about that all the time, that, that sequence. (A1.96-98)

The *guidelines* sometimes also detail how to use the program for the mathematical investigation (e.g. by suggesting range of parameter values) and/or emphasize the need to interpret output within their mathematics knowledge (e.g. by requiring to justify their conclusion from the investigation). We associate these respectively to Steps 5 and 6 in the DP model. For example, Bill's guidelines for P3 read:

> The output should show the mean and standard deviation of the samples. Estimate the hypervolume accurate to one decimal place and use your observations to explain why you are confident that your first decimal place is correct.

This suggests to the students that they must apply their statistics knowledge in order to appropriately use their program and justify their answer (Step 5-6 cycle). In fact, Bill mentions that he revised the guidelines due to his dissatisfaction from past students' poor interpretation of their program output (i.e., more guidance needed for Step 6).Bill deliberately includes a more challenging question (selection between P4 or P5) as part of assignment 1 where *he plans close to no extra guidance* beyond the statement of the problem (i.e. Step 1): "But there has to be a question on every assignment that, is something to think about… This one is solo. Um, they don't get very much help from me" (B.A1.162-4). We interpret Bill's intention that students mobilize and develop, without his help, their whole web of schemes for this particular investigation task.

Didactical Performance

The ways a MICA instructor teaches in lectures and labs involves ad hoc decisions aligned with how they have planned to support students' learning to use programming technology for pure and applied mathematical work mainly through their individual mathematical investigation projects. Based on interactions with students, individually or collectively, and on observations of interactions among students, the instructor takes decision as to how to respond. This response may take form as *individual help* addressing an identified student's difficulty *to develop/mobilize a certain scheme*, or as a class intervention aiming at steering the *collective development a certain scheme*. Bill recalls many individual interventions during the labs. E.g. we interpret Bill's expectation from students to mobilize their scheme of debugging (programming cycle in DP model) when needed. He explicitly mentions it to students: "it will be unusual for either me or the TA to debug your code, that's not our job" (B.A1.207-8). He recalls an intervention with a student, aligned with his expectations, as he sits down beside the student: "Explain the principles and the ideas… If you're desperate, we might look through your code." (B.A1.216) We interpret Bill's response as a reminder to the students of this schemes' effective rules-of-action: '*step back from the code', 'think* *through the big picture of the code design',* and *'think of the different parts of the code'.* Aligning with his planned *'close to no extra guidance'* for P5 that we associated with students' mobilizing their whole web of schemes for this task, Bill mentions helping individual students by responding with guiding questions: "Investigate: what does it mean?...How big should n be? And my answer is 'I don't know'." (B.A1.164; 170). As for a collective response in lab, Bill e.g. comments addressing the students' difficulty of explaining their output from the program in P3, which we interpret as steering the collective mobilization and development, for this task, of schemes associated to step 6:

> I'm very interactive in the lab … so when we get to working on this question I'll be talking about variability… it's an opportunity to work on the board with them… none of this sits by itself. (B.A1.140)

DISCUSSION

The research question guiding this paper concerned what we learn about the teaching of using programming for authentic math investigations by using the frame of the instrumental orchestration. Drawing on our case study of MICA II teaching, we discuss here elements of answer to this question, and indicate directions for future research.

The identified **didactical configuration, main didactical intention** and **the project element as part of the exploitation mode** turned out to be the same for all MICA courses adopted by the department and also followed by MICA instructors (i.e., can be viewed as operational invariants of the collective 'MICA instructor'). This configuration and exploitation mode element stress a 'student-centered' approach, whereby the core of the courses is on individual student projects that aligns with a constructionist approach (Papert, 1980). The 20 years of sustained MICA implementation could suggest that this didactical configuration and exploitation mode element support well the teaching of programming-based math investigations.

The instructor aligns with the collective **exploitation mode**; namely through his/her choices of 'content' through *project guidelines* and *planned guidance in lab and lectures* according to his/her intention of steering the collective students' instrumental genesis of their complex web of schemes associated with the programming-based mathematical investigation activity. This gives insights on how the institutional decisions support well the individual instructors. The exploitation mode of MICA II teaching also highlights that, unlike most technology-rich mathematics courses, the choice of integrating programming comes *before* to the choice of mathematics content, and has led to describe the math content at the *individual* level, rather than the usual *collective* level (as a 'policy maker level'). The **didactical performance** of MICA II teaching pointed to the significance of the lab setting as a key element of the didactical configuration to facilitate the MICA II instructor to steer both individual and collective students' development of schemes. Unlike the didactical configuration, these other two components of MICA instrumental orchestration seem to be evolving. For example, Bill's refining of the project guidelines to explicitly steer the students' mobilization or development of the scheme to mathematically interpret the program output.

In terms of future research, we note that the project guidelines, as a collection, appear to steer students to develop or mobilize their whole complex web of schemes associated with the math investigation activity. Studying aspects of investigation project tasks, as part of the whole task collection, that affect which and how different schemes are guided in the project guidelines (and in lectures and labs), will lead to essential recommendations for practice.

ACKNOWLEDGMENTS

Work funded by SSHRC (#435-2017-0367) with ethics clearance (REB #17-088).

REFERENCES

Benton, L., Saunders, P., Kalas, I., Hoyles, C., & Noss, R. (2018). Designing for learning mathematics through programming: A case study of pupils engaging with place value. *International journal of child-computer interaction*, *16*, 68-76.

Buteau, C., Gueudet, G., Muller, E., Mgombelo, J., & Sacristán, A. (2019a). University Students Turning Computer Programming into an Instrument for 'Authentic' Mathematical Work. *IJMEST.*

Buteau, C., Muller, E., & Ralph, B. (2015). Integration of programming in the undergraduate math program at Brock University. In *Proceedings of Math+Coding Symposium*, London, ON.

Buteau C., Sacristán A. I. & Muller E. (2019b). Roles and Demands in Constructionist Teaching of Computational Thinking in University Mathematics. *Constructivist Foundations 14*(3), 294–309.

Drijvers, P., Doorman, M., Boon, P., Reed, H., & Gravemeijer, K. (2010). The teacher and the tool: instrumental orchestrations in the technology-rich mathematics classroom. *Educational Studies in Mathematics*, *75*(2), 213-234.

Hoyles, C. & Noss, R. (1992). *Learning Mathematics & Logo*, Cambridge: MIT Press.

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

Rabardel, P. (1995). *Les hommes et les technologies; approche cognitive des instruments contemporains.* Paris, France: Armand Colin.

Trouche, L. (2004). Managing complexity of human/machine interactions in computerized learning environments: Guiding students' command process through instrumental orchestrations. *IJCML*, *9*(3), 281–307.

Vergnaud, G. (1998). Toward a cognitive theory of practice. In A. Sierpinska & J. Kilpatrick (Eds.), *Mathematics education as a research domain: A search for identity* (pp. 227–241). Dordrecht: Kluwer.