Understanding how students learn programming for mathematical inquiry at university: Schemes and social-individual dialectics

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

[1]UR EST, Université Paris-Saclay, ghislaine.gueudet@universite-paris-saclay.fr; [2]Brock University, Canada; [3]Cinvestav, Mexico; [4]Universidad del Valle, Cali, Colombia

The instrumental approach, and in particular the concept of scheme, can contribute to our understanding of how students learn programming for mathematical inquiry at university. While most studies consider only individual schemes, in this paper we propose to investigate schemes' social aspects, focusing on the scheme of "Validating the programmed mathematics". Through a questionnaire conducted with students over 2 years, and interview data, we identify shared rules-of-action developing over time. Deepening the analysis for the case of two contrasted students, we observe that common rules-of-action can be associated with different theorems-in-action.

Keywords: Social aspects of schemes, Programming, Instrumental approach, Mathematical inquiry.

1. Introduction

The study presented here is part of a larger 5-year project, whose aim is to better understand how students learn programming for mathematical inquiry at the university level. In our previous work (Buteau, Gueudet, et al., 2020; Gueudet et al., 2020) we have shown that the instrumental approach (Rabardel & Béguin, 2005; Trouche, 2003) and the concept of scheme (Vergnaud, 2009) can contribute to this aim. This concept allowed us in particular to investigate how students develop operational knowledge (knowledge that provides means to do and succeed, Vergnaud, 2009), through instrumental geneses. We focused only on the individual level: the schemes developed by a given student. Nevertheless, at university, students learn in a social context, and most probably their schemes also comprise common elements. Some aspects of the operational form of knowledge are shared; identifying these aspects is essential in particular to inform teachers' orchestrations (Trouche 2003) in this context. The question we investigate here is: Can we identify patterns in the culture of a community of students learning to use programming for mathematical inquiry?

In terms of research, we claim that studying this issue is an important theoretical and methodological contribution, because the instrumental approach has been used mostly for studying individual learning processes. In terms of teaching, we have observed in a previous study (Buteau, Muller, et al., 2020) that insights into students' schemes can be very helpful for teachers orchestrating their learning of programming for mathematical inquiry.

Our study 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 (Buteau & Muller, 2010). In these project-based courses, mathematics majors and future mathematics teachers learn to use programming for mathematical inquiry (e.g., using programming to simulate a battle between two armies; see section 5).

2. Theory and research questions

We draw from Vergnaud's (2009) theory of conceptualization, distinguishing between an operational form of knowledge (knowledge that provides means to do and succeed) and a predicative form of knowledge (knowledge that consists of means to express ideas in words or symbols). We argue that Vergnaud's theory is relevant for our study because operational knowledge is very important when learning to use programming for mathematical inquiry.

Operational knowledge's development and evolution are captured by Vergnaud (2009) through the concept of scheme, which is central in our study. A scheme is an invariant organization of activity for a certain goal and consists of four components: the goal and sub-goals; rules-of-action (RoAs), to generate action, information seeking, and control; operational invariants (concepts-in-action, which are concepts considered as relevant, and theorems-in-action (TiAs), which are propositions considered as true); and possibilities of inferences.

We also use the instrumental approach, particularly its theory of instrumental genesis, which conceptualizes the process of how users (learners) transform an artefact (a human product, designed for a goal-directed activity) into an instrument for a specific goal and situation (Rabardel & Béguin, 2005). For Rabardel and Béguin (2005), an instrument is a hybrid entity: partly an artefact and partly scheme(s).

According to Rabardel and Béguin (2005), schemes have both a private and a social dimension. The private dimension is specific to each individual, while the social dimension reflects the fact that schemes may be shared by members of social groups. The social dimension also may be a consequence of schemes developing during a process involving individuals who are not isolated; for example, classmates working on assignments in a shared space such as a computer lab. Schemes may be shared informally or prompted/promoted formally by training, such as in teaching and learning situations (e.g., through assignment guidelines, lectures, etc.).

This points to the possibility of the development of social dimensions of schemes in teaching situations. In fact, Trouche (2003) claims that through instrumental orchestration, one aim of the teacher is to reduce the variability of the individual schemes students develop, in order to strengthen the social dimensions of schemes; for instance, reflecting those shared by an intended community of practice, such as that of mathematicians (Lave & Wenger, 1991).

The above theoretical elements lead us to refine our initial question as follows: Can we identify common elements in the schemes developed by students learning programming for mathematical inquiry? For a common goal, do students share common RoAs, and if so, are these rules associated with the same operational invariants? How do these common elements develop over time, in a course, and over several years of courses?

3. Methods

In our past work, we developed a general model of students engaging in learning to use programming for mathematical inquiry (Buteau & Muller, 2010), which enabled us to identify common goals in the schemes developed by students. For this study, we focus on one such goal, namely "Validating the programmed mathematics", considered as particularly important because it associates mathematics and programming. For this goal, we further focus on RoAs because most operational invariants and inferences cannot be made explicit; RoAs are more explicitly identifiable.

We mainly draw from two types of data: a questionnaire and individual interviews. The questionnaire was given to students at the end of each MICA course over 3 years (2019–2021), and included questions on demographics, confidence levels in programming (mathematics), et cetera. Since 2020, it also included two questions seeking insights about two specific schemes. Each of these questions presented a list of RoAs in a matrix format and a 5-point Likert scale for students to answer if they used the RoA always, often, sometimes, rarely, or never. Most of these RoAs were identified in our previous work analyzing a student's activity (Buteau, Gueudet et al., 2020), and some more were added based on our experience as mathematicians researching with programming or teaching programming for mathematics investigations. The questionnaire was answered by 30 anonymous volunteer students, in addition to another 13 volunteer students whom we have been following closely throughout their MICA courses (e.g., using also individual interviews).

We used relative frequency bar graphs for responses to each of the questions and for responses regrouped into three categories (always/often, sometimes, rarely/never) in order to identify RoAs that could be considered as social. To do so, we used an arbitrary threshold of 70% of the regrouped always/often responses. We note that the sample of 43 participants is not necessarily representative of the entire MICA community but can nevertheless give insights into potential social dimensions of schemes. For triangulation purposes, we also called upon interview data. For this paper, we selected data collected in Year 2 (2019) of our research since we had the opportunity to collect data both from an instructor (Bill) and some of his students. In one of the interviews with this MICA II instructor, we showed him the list of RoAs and invited him to comment on whether, according to him, students enact the RoAs, and to elaborate about his related guidance.

Finally, we went to individual student cases. Two among eight student participants enrolled in Bill's MICA II class were selected due to their more reflective answers in interviews and for their differing profiles: Kassie is a female student enrolled in the mathematics and education program who had no programming background prior to starting her university studies and Mark is a male computer science and mathematics co-major student who had a significant programming background prior to MICA I. After each of the programming-based mathematics inquiry projects (five assignments), individual semi-structured interviews were conducted. We used guiding questions incorporating "explicitation" techniques (Vermersch, 2006) to help students relive their actions during the development of their investigation projects. Kassie and Mark's interview data were coded to identify potential elements of schemes (in particular, the "Validating" scheme) according to their RoAs and TiAs (Gueudet et al., 2020), which were then confronted with their assignment reports. The outcomes of this analysis were organized in terms of RoAs and summarized for each project in a common Excel table, to which we added Kassie and Mark's own questionnaire responses as well as the relative frequencies of the overall questionnaire responses. Using these tables, we identified individual elements of the "Validating" scheme for Kassie and Mark, respectively.

4. Identifying social aspects of a scheme

In this section, we present the results of our analysis identifying the social RoAs for the "Validating" scheme (Buteau, Gueudet, et al., 2020), as part of the process of creating a program for conducting a mathematical inquiry. We consider an aspect of a scheme to be "social" when it is shared by a social community. In our analysis, we realized that RoAs may be identified as social (or not) depending on whether the community in which they are shared comprises the students in MICA I or the students in upper MICA courses (i.e., MICA II and III). These two communities are not independent: We might even see the MICA I student community as in the process of "becoming" or "developing" into an upper year MICA community. This relates to the instrumental lens, whereby the study of students' instrumental geneses is considered as progressing over time. Hence, we will not consider as our reference community all MICA students. Note that we grouped 2nd- and 3rd-year students (into upper MICA) to have more appropriate sample sizes for comparison purposes (N=18 for upper MICA, N=25 for MICA I). This also reflects the implementation model in these courses: MICA II and III invite students to use programming skills to engage in mathematics inquiry, while MICA I is also focused on developing students' programming skills (due to their lack of background in programming).

RoA	Upper MICA	MICA I
1. I check a few times as I program by compiling with a few inputs.	94, 6, 0	72, 28, 0
2. Once I have programmed it all, I run the program with a few different inputs and compare the output with my hand calculations.	89, 0, 11	92, 8, 0
3. I compare the output of my program with that of a peer or with examples from the internet.	83, 6, 11	40, 24, 36
4. I compare my program with that of a peer.	72, 17, 11	32, 24, 44
5. I ask someone (a peer, a TA, the instructor, etc.)	61, 22, 17	52, 20, 28
6. I trust that I translate in vb.net/python what I do on paper. († This is the original wording in the questionnaire. However, we acknowledge that this formulation rather points to an operational invariant and a 'no RoA'.)	56, 22, 22	28, 32, 40
7. I use other technology (Maple, Desmos, etc.) to generate an example and compare it with the output of my program.	28, 50, 22	64, 24, 12
* I don't really know if it works. (* We label this statement differently because it is not a rule-of-action. It is more an indication that students do not have a fully developed scheme for validating the programmed mathematics.)	11, 22, 67	8, 24, 68

Figure 1: Percentages of participants who say they do the RoA always/often, sometimes, rarely/never, in response to the question "When I program a mathematics concept, I know that it works because…"

Figure 1 depicts findings from the online questionnaire, listing the RoAs in the order of highest to lowest percentage of upper MICA participants stating that they use it always or often when validating a programmed mathematics process.

While these RoAs can seem to concern programming in general, we note that the question invites the students to answer only about programming a mathematics concept. Some of the rules are directly linked with mathematics: calculating by hand (RoA 2), or with Maple (RoA 7). Moreover, when the scheme is mobilized in a specific mathematical situation, these rules are specified in relation to the mathematical contents, as we illustrate in the following section.

The top four RoAs are used by more than 70% of the upper MICA participants, suggesting that those RoAs may be social components of the validating scheme for the upper MICA community.

RoAs 1 and 2 appear to be social also in MICA I. For RoA 2, the proportions are similar: 89% and 92%, respectively. One reason for this may be that the lab and assignment guidelines explicitly encourage students to check their output with hand calculations, starting in MICA I. In contrast, for RoA 1 the proportions have a greater difference: from 72% in MICA I to 94% in upper MICA. One way to explain this is that RoA 1 goes together with the practice of coding incrementally. The MICA II instructor from our study, Bill, explains:

Bill: They will often test. … They develop programs incrementally, so they'll write … just a couple of … a loop or something and they just [say]… see: does that make sense?

Such a skill may require time to develop. Also, the mathematics problems explored in upper MICA lead to more elaborated programs, which can encourage students to code incrementally. This is in contrast with RoA 2, where students check after the program is completed.

RoAs 3 and 4 appear to become social only in the upper MICA community. For RoA 3, the proportion increases from 40% in MICA I to 83% in upper MICA. This could be related to upper MICA students having become more confident in their own skills and more comfortable with their peers; hence they may be more willing to interact and share with their peers. We note that interaction and sharing among MICA students, as well as comparing output with examples from the internet, is also encouraged by instructors. Bill confirms that students do RoA 3 and says:

Bill: There are places where the answers are on the internet and I encourage them to check that their program gives that output.

Similarly for RoA 4, the proportion increases from 32% in MICA I to 72% in upper MICA. Bill again confirms that students do this constantly and are encouraged to do so:

Bill: It's a public forum. Chat, talk, discuss. … We are collaborating and I want to foster that atmosphere.

The last four RoAs do not appear to be social in the upper MICA community. Bill confirmed this in some cases: E.g. for RoA 6, Bill noted that it is not common for students to trust their translation of what they do on paper into the programming language. Finally, we note the low proportions (in *I don't really know if it works) that may be interpreted as indicating that all students already have started to develop their "Validating" scheme to a certain degree by the end of MICA I.

5. Social and individual aspects of schemes

In this section, we further analyze the "Validating" scheme for two students, Mark and Kassie. Considering this individual level allows us to observe similarities and differences when a common RoA is mobilized by different students, and to deepen our analysis of the intertwined mathematics and programming knowledge involved in this process. For the sake of brevity, we focus on assignment 4 (question 1) from the MICA II course taken by Mark and Kassie, and we select only one of the social RoAs. We also evoke other MICA II assignments that confirm the stability of the organization of students' activity and illustrate variability depending on the mathematics.

Presentation of assignment 4

In this assignment, students investigated simulations of a battle between two opposing armies using discrete equations. The assignment contained three questions that became progressively more complex. In the first question, students were told to create a program to output the day-by-day evolution of the battle represented by the Lanchester equations: $X_{n+1} = X_n - a*Y_n$ and $Y_{n+1} = Y_n - b*X_n$, where a and b are fixed parameters and X_n and Y_n represent the number of soldiers in the two armies on day n (the battle ends when X_n or Y_n is less than 1).

RoA 2: A common mobilization, different TiAs

For question 1 of assignment 4, both Kassie and Mark mobilized RoA 2: "Once I have programmed it all, I run the program with a few different inputs and compare the output with my hand calculations", which we found to be social among both MICA I and upper MICA communities. In their individual interviews, they declare for example:

Kassie: I did like the equations myself, just after like the first day of battle and kind of like, compared them with what I got with my program.

Mark: You're able to, again, at least begin on paper to kind of understand and write it out yourself of what the expected results are going to be. So, because of that, I kind of just tried to do the first few questions by hand.

We found evidence of the mobilization of RoA 2 by Kassie and Mark for other assignments as well, confirming that it is part of a stable organization of their activity. We infer that both Kassie and Mark developed a corresponding TiA, such as: "When the result I compute by hand and the output of my program coincide, my program is correct". Both this general TiA and RoA 2 take different forms depending on the specific mathematical content. In assignment 4, once the parameters a and b and the initial values X_0 and Y_0 are chosen, simple computations using the recurrence relations can provide the successive values X_n and Y_n. Students look for an exact match between the output and their hand calculations; however, they are only able to do this for a finite number of days in the simulated battle and eventually trust that the further iterations will be calculated properly too. In comparison, in assignment 1, where students adapted a program (presented in class) simulating the Buffon needle random experiment, they could compute by hand the exact probability (applying a formula given in class). They knew (law of large numbers) that the frequency computed by their program should converge towards this exact probability and used this mathematical result to check their program (in particular, Mark described his actions in this way).

In the questionnaire, while Kassie said that she "Always" uses RoA 2, Mark answered "Rarely" (nevertheless we found evidence that he used this rule on several occasions when completing his MICA II assignments). One possible explanation for this difference is that Mark is aware that a hand calculation is not always possible, depending on the mathematics involved. For example, about assignment 2, where students had to use the daily return percentages of the Dow Jones from 1950 to 2020 to compute the probability of a 2% drop, Mark said:

Mark: If you're given, uh, a mean … or a standard deviation or anything like that, um, you can't just verify your answer because you'd have, what over 50 years of data there.

It seems Mark developed another TiA associated with RoA 2: “The validation through hand calculation is only possible in some particular cases”. Thus we observe in this case that in the “Validating” scheme, even if RoA 2 can be considered as social among the upper MICA community because it seems to be shared by a majority of students, the associated TiAs are not necessarily shared: Mark has developed a TiA about the possibility, or not, to apply this rule, but we did not observe the development of this TiA by Kassie.

6. Conclusion

In our study we attempted to identify patterns in the culture of a community of students learning programming for mathematics inquiry, and more precisely to identify common RoAs in the schemes students develop. We also have investigated the development over time of these shared RoAs, and the TiAs associated with them. In this paper we focused on the scheme developed for the goal of “Validating the programmed mathematics”, which associates mathematical and programming knowledge, and RoAs for this goal that were identified in an earlier study.

Students’ answers to a questionnaire confirm that some of these rules are shared by more than 70% of the students—we consider these rules as social aspects of the “Validating” scheme. A further study of schemes developed individually by two students evidences that the same rule can be associated to different operational invariants, perhaps depending on the profile and experience of the student. We also observed evolutions between MICA I and upper MICA students. We claim that the social aspect of the teaching (the orchestration by the teacher, collective students’ work, etc.) contributed to create a community, with its shared patterns. MICA I students progressively entered the upper MICA community and aligned with its practice (Lave & Wenger, 1991) in terms of using programming for mathematical inquiry. In this university context, the horizon is given by the practice of mathematicians using programming for their own research.

The patterns of this practice were not explicitly stated in the MICA curriculum; they can be considered as operational knowledge, which is often not explicit. The teachers might be aware of some of these patterns and try to support students in addressing a goal in “proper” ways—using certain RoAs, based on certain TiAs. Nevertheless, they might also ignore some of these patterns; the identification of patterns by a research study can be helpful for them.

Most research using the concept of schemes has focused on individual schemes. However, as emphasized by Vergnaud (2009), the scheme–situation pair is a powerful tool in mathematics education research. Studying the aspects of schemes shared or not by students in a situation can also help refine our understanding of the situation. Focusing on social aspects of schemes is important to better understand what the situation is, for a group of students. We claim that using schemes not only as a theoretical tool for research but also as a lever producing interesting results for teachers requires a consideration of the social aspects of schemes, and their social-individual dialectics. In our future research we will further investigate social schemes by networking the theoretical frames of the instrumental approach (Trouche, 2003) and communities of practice (Lave & Wenger, 1991).

Acknowledgment

This work is funded by SSHRC (#435-2017-0367) and received ethics clearance (REB #17-088). We thank K. Dreise and J. Sardella and all research assistants for their work on the analysis.

References

Buteau, C., Gueudet, G., Muller, E., Mgombelo, J., & Sacristán, A. (2020). University students turning computer programming into an instrument for "authentic" mathematical work. *International Journal of Mathematical Education in Science and Technology*, *51*(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). Institut national de recherche pédagogique. http://ife.ens-lyon.fr/publications/edition-electronique/cerme6/wg7-07-buteau-muller.pdf

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. https://tinyurl.com/ynb65tmu

Gueudet, G., Buteau, C., Muller, E., Mgombelo, J., & Sacristán, A. (2020). Programming as an artefact: What do we learn about university students' activity? In T. Hausberger, M. Bosch, & F. Chellougui (Eds.) *Proceedings of INDRUM 2020 Third Conference of the International Network for Didactic Research in University Mathematics*, (pp. 443-452). University of Carthage and INDRUM. https://indrum2020.sciencesconf.org/data/pages/INDRUM2020_Proceedings.pdf

Lave, J., & Wenger, E. (1991). *Situated learning: Legitimate peripheral participation*. Cambridge University Press. https://doi.org/10.1017/CBO9780511815355

Rabardel, P., & Béguin, P. (2005). Instrument mediated activity: From subject development to anthropocentric design. *Theoretical Issues in Ergonomics Science*, *6*(5), 429–461. https://doi.org/10.1080/14639220500078179

Trouche, L. (2003). From artifact to instrument: Mathematics teaching mediated by symbolic calculators. *Interacting with Computers*, *15*(6), 783–800. https://doi.org/10.1016/j.intcom.2003.09.004

Vergnaud, G. (2009). The theory of conceptual fields. *Human Development*, *52*(2), 83–94. https://doi.org/10.1159/000202727

Vermersch, P. (2006). Les fonctions des questions [The functions of questions]. *Expliciter*, *65*, 1–6. https://expliciter.fr/IMG/pdf/Les_fonctions_des_questions.pdf