Future teachers' appropriation of computer programming as a mathematical instrument and a resource for teaching

Ana Isabel Sacristán[1], Marisol Santacruz-R.[2], Chantal Buteau[3], Joyce Mgombelo[3] and Eric Muller[3]

[1]Cinvestav, Mexico; asacrist@cinvestav.mx

[2]Universidad del Valle, Cali, Colombia; marisol.santacruz@correounivalle.edu.co

[3]Brock University, Canada; cbuteau@brocku.ca, jmgombelo@brocku.ca, emuller@brocku.ca

The focus of this paper is on how preservice mathematics teachers appropriate computer programming as an instrument for their own mathematical learning and thinking, and for educational resources they create for others. This is part of a 5-year naturalistic on-going research, where we examine how university students use computer programming as a computational thinking instrument for mathematics inquiry, using a mixed methodology and an iterative design approach. We present the phases of instrumental, personal and professional, and documentational geneses that such future teachers go through, and exemplify parts of these through data from the case studies of two students. This work has implications for the design of future professional development programmes for the integration of computer programming in mathematics education.

Keywords: Programming, mathematics, preservice teacher education, instrumental and documentational approaches, naturalistic observation.

Introduction

Computer programming and computational thinking have taken renewed importance in the last decade in education. Many regions, such as some in Canada, now require the teaching of coding or computational thinking in schools, including in mathematics curricula (e.g., Ontario Ministry of Education, 2020). Thus, the importance that future mathematics teachers appropriate programming as an instrument, both for mathematical learning as well as for their teaching practice. However, little research has been done in terms of how to promote such appropriation, particularly in the case of teachers. In this paper we analyse elements of an undergraduate programme where mathematics students, including future mathematics teachers, learn to use computer programming for mathematics inquiry. Our analysis seeks to illustrate the various geneses that future teachers go through in order to appropriate programming and how the design of the university programme promotes those geneses. Such analysis has implications for the design of other professional development programmes.

This analysis is part of a larger five-year, naturalistic (i.e., not design-based), on-going research (see Buteau, Gueudet et al., 2019, Gueudet et al., 2022), which takes place at Brock University (Canada) where students have the option to take a sequence of three one-semester courses called *Mathematics Integrated with Computers and Applications* (MICA) courses. During those courses, mathematics students and future teachers engage in programming and developing interactive microworlds-type objects or environments for investigations of pure and applied mathematical ideas (see Buteau et al., 2016). Our research questions have focused on how students come to appropriate programming as an instrument for mathematical inquiry and on how the MICA learning environment supports the development of students' instrumental geneses.

In our previous research, we have been using the instrumental approach (Rabardel, 2002) to analyse how MICA participants, mainly mathematics majors, appropriate programming as an instrument that they can use for mathematical investigations (Buteau, Gueudet et al., 2019; Gueudet et al., 2022). And in Mgombelo et al. (in press), we presented the case study of a MICA preservice teacher using programming in the design of a learning object to teach a mathematics concept. Here, we build on that work, with the aim of analysing how the MICA program presents opportunities for future teachers to appropriate computer programming both as a personal instrument for their own learning of mathematics as well as a professional resource for mathematics teaching.

Conceptual and theoretical framework

As mentioned, our research uses as framework, the instrumental approach, which has at its core the concept of instrumental genesis: how an artefact becomes an instrument. In instrumental genesis, an instrument is psychologically constructed by attaching to the artefact (mobilized to realize a type of task), schemes that organise the activity of the subject (Trouche, 2004), through the dual processes of instrumentation (how the artefact affects the user) and instrumentalisation (how the user affects the artefact). Students' instrumental geneses can be steered by how a course is orchestrated.

In this paper, the task of analysing the future teachers' knowledge development and work, is more complex, since the instrumental genesis is not just for turning programming into a personal instrument for oneself, but also as an instrument for professional work. Thus, we rely also on two further frameworks that have emerged from the instrumental approach: the double instrumental genesis (Haspekian, 2014); and the documentational approach to didactics (Trouche et al., 2018).

Haspekian (2014) explains that a same artefact –in our case, computer programming– becomes two different instruments for a teacher: a personal instrument for mathematical activity and professional instrument for the teacher's didactical activity. She calls this *double instrumental genesis*, and says:

> The *personal instrumental genesis* leads (as for pupils) to the construction and appropriation of a tool into an instrument for mathematical work, and differs from the *professional instrumental genesis*, which leads to the construction and the appropriation of the previous instrument into a didactical instrument for mathematics teaching activity [...] these two geneses are not independent (in some cases [...] this double instrumental genesis may happen simultaneously), neither are they independent of pupils' instrumental geneses. Applying the instrumental approach to the [artefact] seen as a teaching instrument built by the teacher, let's precise the two processes of this professional genesis:
> - An instrumentalization process: the tool is instrumentalized by [the] teacher in order to serve her didactic objectives. It is distorted from its initial functions and its didactical potentialities are progressively created (or "discovered" and appropriated in the case of an educational tool).
> - An instrumentation process: [the] teacher, as a subject, will have to incorporate in her teaching schemes that were relatively stable some new ones integrating the tool use.
>
> (Haspekian, 2014, p. 98).

Another extension of the instrumental approach, also related to teachers' professional development, is the Documentational Approach to Didactics (Trouche et al., 2018). This approach focuses on how mathematics teachers interact and use resources (including the digital ones), through the design, re-design or 'design-in-use' of resources for their own work and/or the collective work with other teachers. Resources –which we can think of as "didactic artefacts"–, can be material (e.g., textbooks, digital resources, manipulatives, tasks), social (e.g., forum conversations) and cognitive (e.g., frameworks/theoretical tools used in work with teachers) (Trouche et al., 2018). When teachers interact with resources, they change and develop their professional knowledge; this is called the teacher's *documentation work,* generated through the dialectic *documentational genesis* process*,* involving *instrumentalisation* –how resource affordances influence teachers' practice– and *instrumentation* –the development of schemes of usage of these resources according to the teacher's instructional needs. Through documentational genesis, a system of resources together with their utilization schemes, becomes a *document.*

Context and methodological aspects

As mentioned above, our research focuses on the teaching and learning that takes place within the MICA programme at Brock University. This programme, launched in 2001, currently consists of three one-semester courses, taken over three years: MICA I, II and III/III* –where III is for mathematics and science majors, and III* is for preservice teachers. During these three courses, students design, program (in VB.Net, Python or, in MICA III*, one in Scratch), implement and test a total of 14 programming-based mathematics investigation projects (4 or 5 in each course) in various topics (e.g., conjectures about primes; stock market analysis; dynamical systems; prey-predator models). Most of these are Exploratory Objects (EOs) –microworld-type interactive and dynamic computer-based models " developed to explore a mathematical concept or conjecture, or a real-world situation" (Buteau & Muller, 2009, p. 1112). At the end of each term, students, individually or in groups of two or three, develop a final project, for which they select the topic.

In MICA I and II, for their final projects, most students generally have to create an original EO, but, in some cases, they may choose –preservice teachers, in particular– to create, instead of an EO, a Learning Object (LO) to teach a mathematics concept, which may be relevant to their future profession; an LO is defined as "an interactive and dynamic computer-based environment that engages a learner through a game or activity and that guides him/her in a stepwise development towards an understanding of a [school] mathematical concept" (Buteau & Muller, 2009, p. 1112). In MICA III*, the final project consists of developing and, when possible, implementing a teaching resource of programming-based mathematics activities, in accordance with a regional curriculum, for mathematics classrooms, that could be done in collaboration with a teacher and shared in collaborative resource networks (e.g., in http://mkn-rcm.ca). The 2020 MICA III* final project consisted of designing a teaching resource for grade 9 (with each team designing for a different mathematical topic). The design of the teaching resource used as a model the UK's ScratchMaths (UCL, 2018) curriculum and pedagogy. The aim was for MICA future teachers to develop fluency in Python programming and to put into practice their understandings of learning math through programming. That teaching resource had to include: tasks using Python programming, worksheets in Jupyter Notebook, a short video and follow-up resources for teachers (and optional additional resources, which could be in Scratch), investigate curricular and didactical strategies for their teaching and use their own knowledge about mathematics, programming and teaching to design it.

Although the three-course sequence was initially developed (and implemented) independently from a particular established theoretical framework, as argued in Buteau et al. (2016), it is considered to be framed by a constructionist (Papert, 1991) approach: We consider the EOs created by students to be microworlds and objects-to-think-with (Papert, 1980); furthermore, analyses of the learning environment of the course sequence show an orchestration that promotes students' engagement in constructionist experiences for learning mathematics (Buteau et al., 2016, Sacristán et al., 2020) that may facilitate the appropriation of programming as an instrument for mathematical inquiry.

Our research uses a mixed methodology and an iterative design approach. Over the years, we have collected data from all MICA courses, including course materials (syllabus, assignment guidelines, etc.); students' weekly lab reflections; pre/post-questionnaires; EOs with associated reports; final projects (original EOs or LOs; final MICA III* teaching resource project); and student/instructor semi structured interviews. In addition to that from several MICA I and II courses, we have data from two MICA III* courses (in 2020 and 2022), with detailed data of 7 participants from the first and of 4 from the second (as well as anonymous responses to a questionnaire from larger cohorts). For this paper, we use data from the 2020 MICA III* course; for one MICA III* participant, Kassie (pseudonym), we have analysed also her MICA II data.

Part of our research work has consisted in analysing in detail some students' instrumental geneses, by codifying their responses to the questionnaires and interviews, together with other source material, to analyse in-depth their schemes. We have also analysed the orchestration of the MICA courses (as noted above; e.g., see Sacristán et al., 2020). It is also worth noting that Buteau and Muller (2009) present several development process models (dp-models) of EOs and LOs, which point to some aspects of MICA students' instrumental geneses, as has been discussed in Buteau, Gueudet et al. (2019) and Mgombelo et al. (in press).

Instrumental and documentational geneses in MICA

Whereas some of our previous papers focus on more detailed analyses, in this paper, we attempt to present a broader perspective of the complex geneses (personal, professional and documentational) involved in developing programming –the initial artefact (or resource)– as both a personal instrument for mathematical work, as well as a didactic instrument in the professional teaching activity. The geneses begin with students creating EOs and developing schemes that allow them to appropriate programming as an instrument for mathematical inquiry. Creation of EOs begins in MICA I but continues up to MICA III/III*. Future teachers then develop further their personal and professional geneses by creating LOs. Later, in their design work for the final teaching resources of the MICA III* final project, future teachers learn how to integrate programming with didactic intentions in the design of tasks. In that work, the resource of programming needs to interact with other elements (other resources, that together with programming constitute a system of resources) – a process of documentational genesis. (These processes are schematised in Figure 1).

This resource system is enriched by the constructionist orchestration (Sacristán et al., 2020; Buteau et al., in press; also Buteau, Sacristán & Muller, 2019) offered by the MICA instructor that promotes future geneses: (i) a didactical configuration (based on a previous collective analysis by MICA's instructors –Buteau et al., in press) that centres on selected programming artefacts (e.g.

Python, Scratch, etc.); (ii) an extended didactical configuration that includes assignments and projects to work on collaboratively (with guidelines and lectures on how to design the programming-based teaching resource); and offering the possibility to share MICA III* teaching resources to a professional online network; and (iii) a didactical performance that offers guidance, expert support, discussion sessions, and encouragement to learn as a teacher.

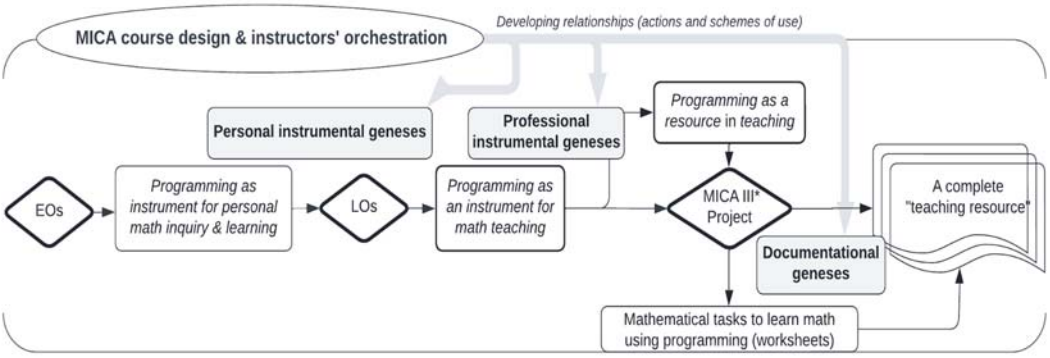

Figure 1. Future teachers' instrumental and documentational geneses in the MICA programme

Detailed examples of the various geneses of the future teachers, are beyond the scope of this paper, but we exemplify parts of these through data from two students: Kassie and Barbara (pseudonyms).

Personal and professional instrumental geneses

Before taking the first MICA course, Kassie had no previous programming knowledge. Through the MICA courses, she not only developed that knowledge, but she developed instrumental schemes relating math and programming (some data illustrating this from Kassie's EOs work in MICA II, is given in Sacristán et al., 2020, and Buteau et al., in press), and later schemes for using programming for teaching math concepts (e.g., a scheme for articulating a learning trajectory in programming language –see Mgombelo et al., in press).

In her personal instrumental genesis (from MICA I to MICA III*), there are two aspects: (i) appropriation of programming as a tool, and (ii) of using programming for mathematics.

In terms of the first, it is interesting how when, in MICA III*, she had to program in Scratch and Python, and she commented that her previous MICA VB.Net programming experience allowed her to "produce a program that works", although she acknowledged that she also needed to change her perspective (MICA III*, AR1). So, part of a scheme may be: "If I have experience in programming, then it is easier to program (math tasks) in other languages (Python/Scratch)." (This was also found in other students: for instance, Barbara, after recalling her MICA II LO creation, said "*So, we had a lot of VB.net [coding] experience at that point [...] so we didn't struggle that much*" –FI).

In terms of the second, Kassie made reference to the challenges she faced (she wrote: "*connecting what I am programming with the math that is involved in the actual program has been something that is rather difficult*" –MICA III*, AR3). But she explained how, in writing the computer program of an assignment, she had to understand the mathematics and programming concepts and relate them, and how that changed her thinking ("…*the program [...], it teaches you a different way of thinking and allows you to expand your horizons in a subject*" –MICA II; LR1); she later added:

Kassie: …when working through the program I was able to understand this problem better. This resulted in an increase of my understanding through the program aspect but as well as my math understanding. (MICA III*, AR1).

Also, the reflections promoted by the course orchestration (Trouche, 2004), prompted her to think about how to improve her abilities of programming for mathematics:

Kassie: In regard to the math aspect of programming I want to learn how to question "what if I changed this, what would the result be?" I believe this would further develop my understanding of the problems I am given. (MICA III*, AR3)

At the end of MICA II, Kassie created an LO with her partner for other users to learn about derivative –an interactive tool that would generate random equations and graphs and lead the user to find the derivatives. In Mgombelo et al. (in press), we showed some of Kassie's instrumented schemes that corresponded to some steps of the development process of an LO and how Kassie developed some schemes, including the above-mentioned one of articulating the learning trajectory in programming language. This illustrated part of her professional geneses. Later, Kassie and her partner designed a teaching resource (for the final MICA III* project) for teaching exponents, that not only included Python tasks, but also Scratch, which shows how she instrumentalised the different programming languages for designing tasks to learn about a mathematics concept.

Documentational genesis in a MICA III* final project

We consider that an extension of the professional genesis, are documentational geneses. Creating LOs already involves a certain degree of documentational genesis. But the final MICA III* project requires deeper interactions of diverse resources and knowledge (including that of programming for math). The orchestration of the project, which has a set of requirements, as outlined above, promotes this. And students respond to those requirements in the "teacher resources" –which are systems of resources with usage schemes, i.e., documents– that they develop for teachers.

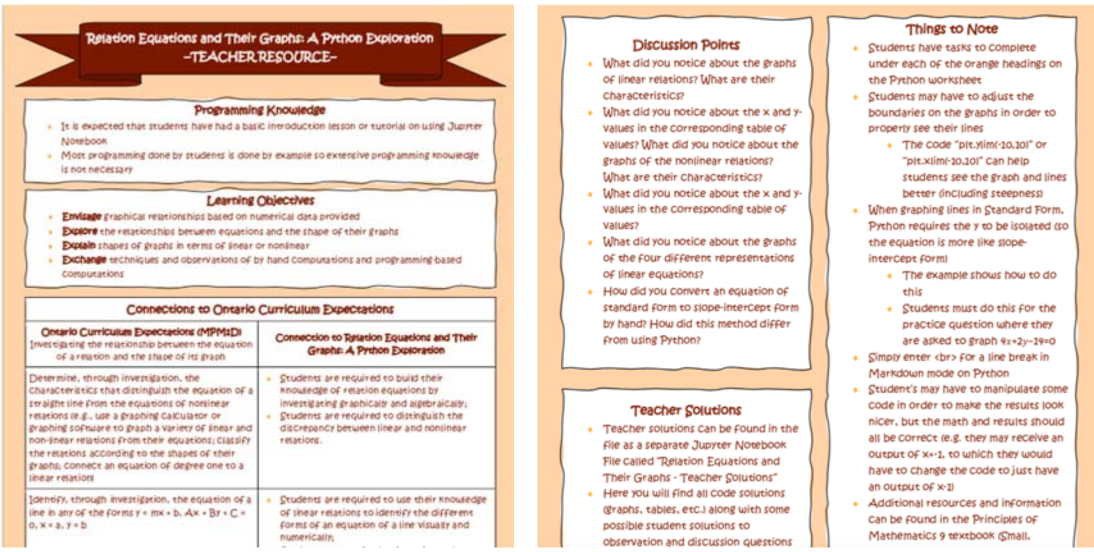

Figure 2. Fragments of Barbara and her partner's Teacher Resource (MICA III* final project)

For example, Barbara and her partner developed a teacher resource for exploring the relation of equations with their graphs with Python (Figure 2). That resource included the activity summary of the “Teacher Resource", together with accompanying self-contained Python worksheets (for students) in Jupyter Notebook (as well as a solution file for the teacher) and additional resources that include the required video, as well as posters, and cards for additional activities. The activity summary begins with the programming knowledge required for students (in this case, the future teachers chose for the programming done by students to be carried out “by example” so that extensive programming knowledge would not be needed, with a code that is re-used and modified); contains clear learning objectives in terms of the math content and alignment to the regional curriculum, as well as “the five Es" of the ScratchMaths pedagogy, activity instructions, discussion points, things to note, etc. Thus, in Barbara’s case, in her documentational genesis, one of her schemes, with the goal (for her teaching resource) of designing math tasks that integrate coding, was based on a previous programming and math (p+m) scheme (Gueudet et al., 2022) developed through her personal instrumental genesis, that now extended to the new situation of designing teaching tasks, which require professional knowledge (e.g., curricular considerations) in interaction with her programming (for math) knowledge.

Concluding remarks

In this paper we have attempted to illustrate how during the MICA activities, future teachers, through a constructionism-based orchestration, have the opportunity to develop both their double instrumental geneses of programming for mathematical investigations, as well as the dual processes of documentational genesis: Their *instrumentalisation* in terms of how programming affordances influence their task design for the programming-based teaching resources that they design, can scaffold other students’ activity. And their *instrumentation* involving usage schemes within the design of the teaching resource; for instance, a usage scheme that guides their task design provides future teachers with the opportunity to interact with a wide and diverse system of resources for their teaching: computer programming, curricular resources, worksheets, specialised software, etc. In this way, we have extended the research presented in our previous papers by analysing the complex geneses involved in teachers’ knowledge development, and illustrate ways for the design of other professional development programmes in the field. Nevertheless, our work continues with more detailed analyses of the data that we have collected.

Acknowledgment

Our research is funded by SSHRC (#435-2017-0367) grant with ethics clearance (REB #17-088).

References

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 CERME 6*, Jan. 28th-Feb. 1st, 2009 (pp. 1111–1120). Lyon, France: INRP. http://www.inrp.fr/editions/cerme6

Buteau, C., Muller, E., Marshall, N., Sacristán, AI., Mgombelo, J. (2016). Undergraduate mathematics students appropriating programming as a tool for modelling, simulation, and visualization: A case study. *Digital Experiences in Mathematics Education*, 2(2), 142–166. https://doi.org/10.1007/s40751-016-0017-5

Buteau, C., Sacristán, A. I. & Muller, E. (2019). Roles and demands in constructionist teaching of computational thinking in university mathematics. *Constructivist Foundations* 14(3): 294–309. https://constructivist.info/14/3/294

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*, 1–22. https://doi.org/10.1080/0020739X.2019.1648892

Buteau, C., Muller, E., Santacruz-Rodríguez, M., Mgombelo, J., Sacristán, A., & Gueudet, G. (In Press). Instrumental orchestration of using programming for authentic mathematics investigation projects. In A. Clark-Wilson, O. Robutti, & N. Sinclair (Eds.), *The Mathematics Teacher in the Digital Era* (2nd Edition). Springer Netherlands.

Gueudet, G., Buteau, C., Muller, E., Mgombelo, J., Sacristán, A. I. & Santacruz-Rodriguez, M. (2022). Development and evolution of instrumented schemes: A case study of learning programming for mathematical investigations. *Educational Studies in Mathematics*, 110(2), 353–377. https://doi.org/10.1007/s10649-021-10133-1

Haspekian, M. (2014). Teachers practices and professional geneses with ICT. *Research Journal of Mathematics and Technology*, *3*(1), 96–105.

Mgombelo, J. Pinar, A., Buteau, C., Muller, E., Sacristán, A. & Santacruz, M. (In press). Pre-service teachers using programming to design learning objects: A case study. *Proceedings. CERME 12, 2 – 6 Feb. 2022,* Bozen-Bolzano, Italy.

Ontario Ministry of Education (2020). *The Ontario Curriculum Grades 1-8: Mathematic*s. Queen's Printer for Ontario. https://www.dcp.edu.gov.on.ca/en/curriculum/elementary-mathematics/downloads

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

Papert, S. (1991). Situating Constructionism. In I. Harel & S. Papert (Eds.), *Constructionism* (pp. 1–12). Norwood, NJ: Ablex Publishing Corporation.

Rabardel, P. (2002). *Les hommes et les technologies: Approche cognitive des instruments contemporains* [People and technology: A cognitive approach to contemporary instruments]. Paris: Armand Colin. https://hal.archives-ouvertes.fr/hal-01017462/document

Sacristán, A.I., Santacruz-Rodríguez, M., Buteau, C., Mgombelo, J., & Muller, E. (2020). The constructionist nature of an instructor's instrumental orchestration of programming for mathematics, at university level. In T., Brendan, J.R. Byrne & C. Girvan (Eds.), *Constructionism 2020* (pp. 525–536). TARA. http://www.tara.tcd.ie/handle/2262/92768

Trouche, L., Gueudet, G., & Pepin, B. (2018). Documentational Approach to Didactics. In S. Lerman (Ed.), *Encyclopedia of Mathematics Education* (pp. 1–11). Springer International Publishing. https://doi.org/10.1007/978-3-319-77487-9_100011-1

Trouche, L. (2004). Managing the complexity of human/machine interactions in computerized learning environments: Guiding students' command process through instrumental orchestrations. *International Journal of Computers for Mathematical Learning* *9*(3), 281–307. https://doi.org/10.1007/s10758-004-3468-5

UCL (2018). *UCL ScratchMaths Curriculum*. IOE, University College London. https://www.ucl.ac.uk/ioe/research/projects/ucl-scratchmaths/ucl-scratchmaths-curriculum