Programming as an artefact: what do we learn about university students' activity?

Ghislaine Gueudet[1], Chantal Buteau[2], Eric Muller[2], Joyce Mgombelo[2] & Ana Isabel Sacristán[3]

[1]CREAD, ESPE de Bretagne, University of Brest, ghislaine.gueudet@univ-brest.fr; [2]Brock University, Canada; [3]Cinvestav, Mexico

In this paper we discuss how the instrumental approach can contribute to our understanding of the activity of university students using programming in the context of an authentic mathematical investigation. We claim that they develop an instrument from programming considered as an artefact, incorporating a complex structure of schemes. We distinguish between m-schemes, p-schemes and p+m-schemes, for a goal concerning respectively only mathematics, only programming, or both. We illustrate this theoretical construct by studying the case of a student enrolled in a course encompassing programming-based mathematics investigation projects.

Keywords: Teachers' and students' practices at university level, Digital and other resources in university mathematics education, Programming, Instrumental approach, Authentic Mathematical Investigations.

INTRODUCTION AND CONTEXT

In the field of mathematics education, the use of 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 been framed with different perspectives (e.g., see Hoyles & Noss 1992). We present here a study concerning the theoretical contribution of the instrumental approach (Guin, Ruthven & Trouche 2005) –that articulates the mutual shaping of learners and artefacts (e.g., programming) in the learning process–, to analyse the activity of university students using programming in the context of an authentic mathematical investigation.

The instrumental approach has already been used in previous research about university students' use of various technological tools: for example Sketchpad (Ndlovu, Wessels & De Villiers 2011) or CAS (Zeynivandnezhad & Bates 2018). This theoretical framework has also been used in a study about programming by Misfeldt and Ejsing-Duun (2015), but their work concerns the primary and lower secondary levels. As far as we know, the instrumental approach has never been used in a research about programming at university level; we hypothesize that it can enlighten interesting phenomena, specific from this level and from programming.

Our study is part of a five-year naturalistic (i.e., not design-based) 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, mathematics majors and future mathematics teachers learn to design, program, and use interactive

environments to investigate mathematics concepts, theorems, and applications (Buteau & Muller 2010). 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 research question that we will investigate here can be presented as follows: What do we learn about the activity of students using programming in an authentic mathematical investigation by using the theoretical frame of the instrumental approach, considering programming as an artefact?

In the next section we present the instrumental approach, and how we propose to use it when the artefact is a programming language. Referring to the Theory of Conceptual Fields (Vergnaud 1998), we introduce in particular three different kinds of schemes. Then we briefly present our methods, and illustrate the use of the instrumental approach by analysing the case of a student, Jim, and of his work in a project concerning number theory. Finally we discuss the insights gained from the use of this approach.

INSTRUMENTAL APPROACH, PROGRAMMING AND SCHEMES

The instrumental approach (Rabardel 1995) 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. The instrument is composed by a part of the artefact and a scheme of use of this artefact (Vergnaud 1998). In mathematics education, the instrumental approach has been used firstly to study learning processes of secondary school students using calculators (Guin et al. 2005). These studies used a detailed definition of schemes, following the work of Vergnaud. A scheme has four components:

- The goal of the activity, subgoals and expectations;
- Rules of action, generating the behaviour according to the features of the situation;
- Operational invariants: concepts-in-action, which are concepts considered as relevant, and theorems-in-action, which are propositions considered as true;
- Possibilities of inferences.

In a given situation, a subject mobilizes a scheme corresponding to the goal of his/her activity. The inferences permit the adaptation of the scheme to the precise features of the situation. Sometimes this adaptation can lead to the emergence of new operational invariants, new rules of action, of even to the emergence of a new scheme. The schemes of use as defined in the instrumental approach come in fact from a more general theory elaborated by Vergnaud in the context of mathematics education: the Theory of Conceptual Fields (TCF). The couple (scheme, situation) is central in this theory to analyse conceptualization processes. We refer also to this more general theory, considering not only schemes of use of “programming”, considered as an artefact, but also mathematical schemes.

In the context of our study, the general goal of the students' activity is to "investigate a complex situation (mathematical or not), combining mathematical knowledge and programming". We claim that, by using programming for this general goal, the students develop an instrument, associating some aspects of programming and schemes of use for specific subgoals (Buteau, Gueudet, Muller, Mgombelo & Sacristán 2019). We also claim that for this general goal, students mobilize different kinds of schemes. Mathematical schemes (noted m-schemes) intervene when the goal (or sub-goal) is the search for a mathematical formulation of the situation, and their interpretation of solutions. Programming schemes (noted p-schemes) intervene when the goal concerns only programming, and could also appear when programming outside of any mathematical context. Combined programming and mathematics schemes (noted p+m-schemes) intervene when the goal concerns both. Along their investigation activity, students develop a complex network of m-schemes, p-schemes, and p+m-schemes. We will illustrate this below by studying the case of Jim.

METHODS

Concerning Jim, we collected and analysed the following data:

Jim was one voluntary student participant (among 6) enrolled in MICA I course (46 students) in the first year of our research. The MICA I course consists of 4 programming-based mathematical investigation projects (which count for 71 % of students' final grades): 3 assigned individual ones, and a fourth one where students select the topic. The course format includes a two-hour weekly lab, where students progressively learn to program in Visual Basic.net (vb.net) in a mathematical context, and two-hour weekly lectures that introduces students to the mathematics needed for their investigation project assignments (Buteau, Muller, & Ralph 2015).

Jim's data included his 4 project assignments (that include each, a computer program and accompanying report) and individual semi-structured interviews after completion of each of his projects. The interview guiding questions were informed by a development process model (referred onwards as 'dp-model'), established in previous works (e.g., Buteau & Muller 2010; Buteau et al. 2019), which describes an individual student's activity in the context of an authentic programming-based mathematical investigation (Figure 1). Jim's data also included weekly post-laboratory session online reflections (with guiding questions) and an initial baseline online questionnaire, followed by an interview, before the beginning of MICA I. We also collected all course material, including lab session and assignment guidelines. For this study, we focused on Jim's baseline questionnaire interview, his first 4 lab reflections, and his first assignment project and interview.

We analysed Jim's interviews by trying to observe in his declarations elements of schemes: goal of the activity; description of how he acted in the situation; reasons for acting this way; and inferences. How he acted can be interpreted as rules of action, if it is described by Jim as a regular practice. If it is described as something new, an original attempt, it can be interpreted as the emergence of something that can later

become a rule of action. The reasons for acting regularly in a certain way are interpreted as operational invariants: theorems-in-action, and associated concepts-in-action. We present examples in the next section.

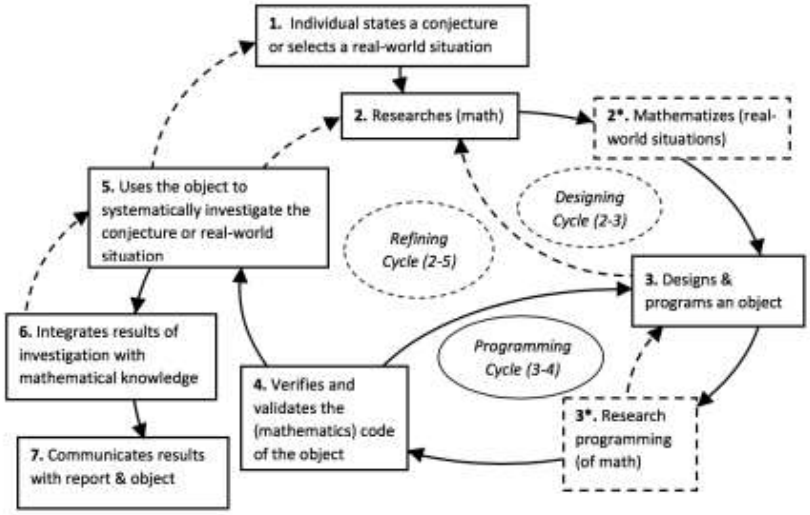

Figure 1: Development process model of a student engaging in programming for an authentic mathematical investigation or application (Buteau et al. 2019).

In this exploratory study, we did not have the possibility to directly observe Jim's work, in order to confront his declarations and his actual activity. Only a part of this activity was accessible through the assignment he produced. This is certainly a limitation of our ongoing naturalistic study, but on the other hand it incorporates all institutional constraints of Jim's activity. We mitigate this limitation by triangulating all the data available on Jim (listed above), and as such, we suggest that our analysis provides significant evidence of Jim's instrumental genesis.

THE CASE OF JIM

The first four weeks of the MICA I course prepare students for their first project assignment. In lectures, students are mainly exposed to prime numbers and hailstone sequences, and to conjecturing about those concepts. In lab sessions, students start learning about basics of programming in vb.net: variables, loops, conditional statements, and create, read from, and print in a graphical user interface (GUI). Starting in lab 3, students are progressively guided to code mathematics; e.g. in lab 3, the code for checking the primality of an integer is given to them for reproducing (and fixing a minor issue) whereas lab 5 guidelines gives a partial pseudo-code for powers in Z_n. The first project directly builds on lab 3 and asks students to state or to select a conjecture about primes, and create a program in vb.net to investigate it.

In this section we present examples of schemes identified in the case of Jim for his first project assignment, chosen to illustrate the three kinds (m-schemes, p-schemes and p+m-schemes). The schemes are presented with general aims: indeed they apply in the context of this assignment, but they are an invariant organization of the activity for all the situations corresponding to this aim. We attempt to give for each scheme

its general description, and elements about its application in context, involving precise mathematical and programming contents. The different elements of the schemes are inferred from the description Jim gave of his activity. For each scheme, we firstly present its main elements in a table, and then comment and discuss this table by drawing on excerpts of Jim's interview. There is no inference mentioned in the tables since none were identified for these examples of scheme.

Formulate a conjecture: example of a m-scheme

Rules of action	Investigate the math concept (search on the Internet, take notes); Search for a representation; Search for a pattern
Concepts-in-action	Representation; Pattern
Theorems-in-action	Understanding related concepts helps to formulate a conjecture; An appropriate representation is helpful to find a pattern; I can learn mathematics by exploration

Table 1: Jim's scheme of formulating a conjecture.

The scheme presented in table 1 is a mathematical scheme, since it corresponds to the goal "Formulate a conjecture". According to Jim, he started by trying to understand better the concept of primes.

> Jim: At first I was trying to kind of think of trying to understand more about the nature of primes before I would really do my conjecture (#2)

We interpret this as a rule of action, governed by a theorem-in-action: "to formulate a conjecture, a good understanding of the concepts involved is needed". It is possible to consider the goal: "investigate a mathematical concept" as a sub-goal, for Jim, of the goal "to formulate a conjecture".

After this first step, Jim tried to represent the prime numbers and to observe a pattern.

> Jim: me trying to figure out this conjecture basically where I would plot out the primes and look for any patterns of how they worked (#3)

We interpret this again as a rule of action, probably developed along many problems in mathematics. The concepts of "representation", "pattern" are relevant for Jim in this situation and guide his activity: they can be considered as concepts-in-action (which are explicit here). In the specific case of prime numbers, he started by representing them on a line (we interpret this as a rule-of-action for the sub-aim "formulate a conjecture about primes). He observed that it did not work, and that a two-dimensional representation was more relevant.

Articulate in a programming language a nested process: example of a p-scheme

Rule of action	Code nested loops articulating the nested process; Code them incrementally
Concepts-in-action	Nested system; Nested loops; Loop

Theorem-in-action	A nested system can be processed by programming technology as nested loops; Incremental coding helps to properly structure the nested loops

Table 2: Jim's scheme of articulating in vb.net a nested process.

The scheme presented in table 2 is a p-scheme, since it corresponds to the goal "Articulate in vb.net a nested process". Jim seems to suggest that the MICA course facilitates students to develop a scheme of "articulating, in vb.net, a process involving repetitions"—with a main rule-of-action: "to code loops"—and that through this assignment project, he (and his fellow students) had to then further elaborate it by developing a more general scheme of "articulating, in vb.net, a nested process".

> Jim: We actually went over how to build this kind of system [involving repetitions] in class. So the only thing new about the project was kind of learning how to nest them, properly structure them, to make this running program. (#18)

According to him, Jim codes nested loops to articulate the nested process. We interpret this as a rule of action, governed by: "a nested system can be processed by vb.net programming technology as nested loops". In addition, Jim seems to indicate coding such nested loops incrementally— a rule of action that we could associate to a theorem-in-action: "Incremental coding helps to properly structure the nested loops".

> Jim: to understand this idea of nested kind of system and how to build upon a single system into multiple ones …Like one system inside another and I think that's pretty key but you kind of just have to work with it and hope it works out... It [is] one of those inherent things. (#27)

In this situation, we identify "nested system", "nested loops", and "loops" as explicit concepts-in-action in Jim's activity. Furthermore, this scheme seems to be, for Jim, at the core of programming. As such, this suggests Jim's awareness of mobilizing or developing it further in his future programming-based mathematical investigations.

Articulating a mathematical process in programming: example of a p+m scheme

Rules of action	Organize the math process as a nested system; Decompose the nested system in individual processes before programming; Code individual processes; Start by 'translating' in vb.net what I would do by hand
Concepts-in-action	Mathematics & programming as a nested system; Solving-by-hand method; Decomposition of a system; Individual process
Theorems-in-action	A mathematical process can be seen as a nested system, i.e., made of many parts; To program a nested mathematics process, one can break it down and individually code the smaller parts; A programming language can work in a similar manner as one works by hand; Programming and mathematics as systems have embedded layers

Table 3: Scheme of articulating a mathematical process in the programming language.

Table 3 presents an example of a p+m-scheme: articulating a mathematical process in programming. It is a p+m-scheme because its goal involves both programming and mathematics. In describing how he approached the assignment Jim noted,

> Jim: I basically tried to organize and sort out what needed to be programmed but I kind of realized as I was going, I kind of knew everything that needed to be done. It just required a set of system nested within each other so once I know that I had to figure out how to program each individual system. This one check for prime. This one is a loop … that sort of thing. (#8)

We interpret this description by Jim as indicating many rules of action, such as "Organize the mathematics process as a nested system", which is supported by a concept-in-action, "mathematics and programming as a nested system" and a theorem-in-action, "a mathematical process can be seen as a nested system", i.e., made of many parts. Two other rules-of-action are: "Decompose the nested system in individual processes before programming" and "Code individual processes"; they are supported by two concepts-in-action, "decomposition of a system" and "individual process", and theorems-in-action, "to program a nested mathematics process, one can break it down and individually code the smaller parts" and "Programming and mathematics as systems have embedded layers". For example in this case he identified a part of the program checking primality.

Jim also described his coding by enumerating different processes that seem to align with a by-hand method. We interpreted it as indicating Jim's rule-of-action "Start by 'translating' in vb.net what I would do by hand", governed by a theorem-in-action "A programming language can work in a similar manner as one works by hand", identified in lab3 as potential components since they were not yet put into action.

> Jim: [I] do believe that… I would have…been able to make something resembling it … as the logic of how it searches for primes has already been ... in class. (Lab3)

DISCUSSION AND CONCLUDING REMARKS

The research question studied in this paper was: "What do we learn about the activity of students using programming in an authentic mathematical investigation by using the theoretical frame of the instrumental approach, considering programming as an artefact?". Drawing on the example studied above, we discuss here elements of answer to this question, and indicate directions for future research.

Firstly, we claim that this example of the activity of students using programming in an authentic mathematical investigation illustrated the relevance of the different kinds of schemes: m-schemes, p-schemes, p+m-schemes. Gerianou and Janqvist (2019) argue that the theory of instrumental genesis, and schemes in particular, allow to bridge mathematical competencies and digital competencies. Our study illustrates and confirms this, in the specific case of programming technology. The p+m-schemes can be considered as bridging mathematical and digital competencies; the identification

of p+m operational invariants in particular deepens our understanding of how mathematics and programming relate to each other.

This statement could apply to any context of learning programming in a mathematics course. Our second claim concerns specifically the university context, and the type of activity proposed in the MICA course: using programming for an authentic mathematical investigation. The different schemes developed by students along this course are inter-related; they constitute a complex structure. We mentioned above the dp-model describing students' activity in the context of an authentic programming-based mathematical investigation (Figure 1). We claim that the schemes developed by a student are related to the different steps of this model (Buteau et al. 2019). In the example presented above, the m-scheme: "to formulate a conjecture" corresponds to step 1; while the p-scheme: "articulate in a programming language a nested process" and the p+m-scheme: "articulate a mathematical process in the programming language" correspond to step 3 (see Figure 1). In our analysis of the case of Jim, we identified schemes corresponding to all the steps of the model (which cannot be presented here, for the sake of brevity). The steps of the dp-model can be considered as general goals of the students' activity; they correspond to what Rabardel (2005) calls "activity families", gathering situations with a similar aim of the activity.

Another direction for investigating how different schemes are linked concerns the level of generality of the goals. We could consider, for example, that "Designing and programming an object" is a goal (step 3 of the dp-model), and thus associated with a single scheme (a p+m-scheme, in this case). The schemes: "articulate in a programming language a nested process" and "articulate a mathematical process in the programming language" would then appear as sub-schemes (associated with sub-goals). In our study we have made a different choice, since we were interested in looking for precise operational invariants in particular. A study in terms of schemes always needs to choose a "favoured" level of generality; it is then possible to consider more precise goals, and obtain sub-schemes. For example in the case of Jim, we also identified a scheme labelled: "To formulate a conjecture about primes", which is a sub-scheme of the m-scheme presented above.

An important issue requiring further work concerns the complex dialectics between stability and evolution of the schemes developed by the students and of their components. The data that we analyzed for this paper did not include a long-term observation of Jim's activity. This had several consequences on our analyses in terms of schemes. Firstly in most cases (in particular in the examples above) we were not able to describe the "inference" part of the scheme. Indeed the inferences are linked with particular features of the situation, not always described in an interview. Second, some of the rules of action and operational invariants described above can be considered as stable while others are only "potential", since more evidence would be needed to acknowledge their stability; we provide below some examples.

We consider that the students in the MICA course (and Jim in particular) already had stable m-schemes: for example they might have met before situations in mathematics

classes where they would have needed to formulate a conjecture. These m-schemes will be adapted to the features of the new situation involving programming, but the organization of the activity will remain stable. In contrast, programming was for them a new activity; the p- and p+m-schemes we identified have most likely been developed during the MICA course.

Nevertheless we claim that some of the rules of action and operational invariants in p- and p+m-schemes are already stable at the end of the MICA course. Indeed the intervention of some of them has been observed on several occasions, and we consider that this acknowledges their stability. For example, we found evidence that, in his second and third MICA projects, Jim seems to utilize his rules of action of the p+m-scheme described above, while also developing additional rules-of-action (e.g. "I ignore coding special cases of the mathematics process that are not needed for the mathematical investigation") and theorems-in-action (e.g. "Special cases in the mathematics code potentially leading to bugs but that don't affect the mathematical investigation, can be ignored").

Some authors have researched the development and evolution of schemes (e.g. Coulet 2011), and consider that along his/her activity, the subject receives feedback which can lead to three kinds of evolutions. Productive loops lead to changes in the rules of action; constructive loops lead to changes in the operational invariants; changes scheme loops can even lead to a new scheme. Studying the stability of potential rules of action and operational invariant requires the study of these loops; this is a perspective for future research.

Finally, another issue requiring further work concerns the nature of the operational invariants. As mentioned earlier, the general goal of the students' activity in the MICA course is to "investigate a complex situation, combining mathematical knowledge and programming". Some of these students, Jim in particular, developed an instrument for this goal from the programming artefact. We hypothesize that these students have developed a theorem-in-action like: "Using programming, I can be creative in mathematics" (and some indices of such a theorem-in-action appear in Jim's interview). This kind of proposition is strongly linked with students' self-confidence and affects. We wonder if such components are involved in schemes.

The links between mathematical and programming competencies are complex and increasingly important at university in several strands: for mathematics majors, but also for future engineers etc. The approach we propose here with the instrumental approach can enlighten these links. Thus we consider important to investigate the future directions evoked above: in particular observe students' activity on a long term, in different contexts, to deepen our knowledge of the schemes they can develop, of their evolutions, and of the complex structure of scheme systems.

ACKNOWLEDGMENTS

This work is funded by SSHRC (#435-2017-0367) and has received ethics clearance (REB #17-088). We thank all of the research assistants for their valuable work

toward our research, in particular Kirstin Dreise who contributed to the analysis of Jim and Marisol Santacruz Rodríguez who contributed comments to the proposal.

REFERENCES

Buteau, C., Gueudet, G., Muller, E., Mgombelo, J., & Sacristán, A. (2019). University Students Turning Computer Programming into an Instrument for 'Authentic' Mathematical Work. *International Journal of Mathematical Education in Science and Technology*. DOI: 10.1080/0020739X.2019.1648892

Buteau, C. & Muller, E. (2010). *Student Development Process of Designing and Implementing Exploratory and Learning Objects.* In *Proceedings of the Sixth Conference of European Research in Mathematics Education (CERME),* (pp. 1111–20), Lyon, France.

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.

Coulet, J.-C. (2011). La notion de compétence : un modèle pour décrire, évaluer et développer les compétences. *Le travail humain, 74*, 1-30.

Geraniou, E., & Jankvist, U. T. (2019). Towards a definition of "mathematical digital competency". *Educational Studies in Mathematics*, *102*(1), 29-45.

Guin, D., Ruthven, K., & Trouche, L. (Eds.). (2005). *The didactical challenge of symbolic calculators: Turning a computational device into a mathematical instrument.* New York, NY: Springer.

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

Ndlovu, M., Wessels, D., & De Villiers, M. (2011). An instrumental approach to modelling the derivative in Sketchpad. *Pythagoras*, *32*(2). https://doi.org/10.4102/pythagoras.v32i2.52

Papert, S. (1972). Teaching Children to Be Mathematicians vs. Teaching About Mathematics. *International Journal of Mathematics Education and Science Technology*, 3, pp. 249–262

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

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 Academic.

Zeynivandnezhad, F., & Bates, R. (2018). Explicating mathematical thinking in differential equations using a computer algebra system. *International Journal of Mathematical Education in Science and Technology*, *49*(5), 680-704