# A STUDENT'S COMPLEX WEB OF SCHEMES DEVELOPMENT FOR 'AUTHENTIC' PROGRAMMING-BASED MATHEMATICAL INVESTIGATION PROJECTS

Chantal Buteau[1], Ghislaine Gueudet[2], Kirstin Dreise[1], Eric Muller[1], Joyce Mgombelo[1] & Ana Isabel Sacristán[3]
[1]Brock University, [2]Université de Bretagne Occidentale, [3]Cinvestav-IPN.

## Introduction

• 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 (mostly addressing school level) have been framed with different perspectives (e.g., see Hoyles & Noss 1992).

• We present **a study concerning the theoretical contribution of the instrumental approach** (Rabardel, 1995) **to analyse the activity of university students using programming in the context of an 'authentic' mathematical investigation**, i.e., complete programming-based mathematical investigations "as mathematicians would do" (cf. Weintrop et al., 2016).

• The instrumental approach has already been used in previous research about university students' use of various technological tools.

• 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.

• The study presented in this poster **extends our work presented in Gueudet et al. (2020).**

## Research Question

**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?**

## A 5-year Naturalistic (not design-based) Research

• 2017-2022 research study, funded by the Canadian Social Sciences and Humanities Research Council (SSHRC)

• takes place in a sequence of three programming-based mathematics courses, called Mathematics Integrated with Computers and Applications I-II-III ("MICA"), implemented in the mathematics department at Brock University (Canada) since 2001:
  – Math majors and future math teachers learn to design, program, and use interactive environments to investigate math concepts, conjectures, or real-world applications
  – 4-5 math investigation projects in each MICA (counts towards ~75% of students' final grades in each MICA)

• 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.

CTmathU ctuniversitymath.ca
Computational Thinking in University Mathematics

## Instrumental Approach & Schemes

• The instrumental 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**.

•A **scheme** is a stable organization of the subject's activity for a given goal (Vergnaud, 1998). It 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
  iv. possibilities of inferences.

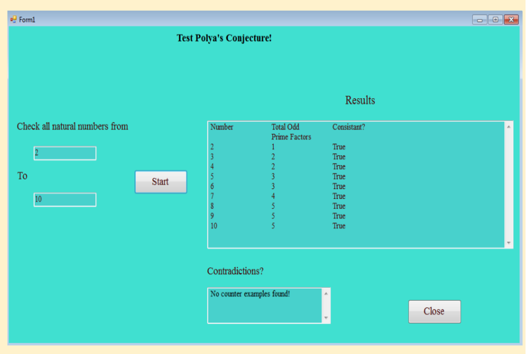
• In our context, the artefact is a syntax-based programming language, such as Python, C++, or vb.net (Buteau et al., 2019)

• Gueudet et al. (2020) distinguish between **m-schemes**, **p-schemes** and **p+m-schemes**, for a goal concerning respectively only mathematics, only programming, or both.

## The Activity

E.g. Jim selected Polya's conjecture on prime numbers (**step 1**) for his first math investigation using vb.net programming.

Jim created by drag&drop a GUI and wrote his program by building on his code about checking the primality of an integer to add a loop counting the number of prime factors for a given n, within a loop for all integers n up to an input given by the user, i.e. Jim himself (**step 3**).

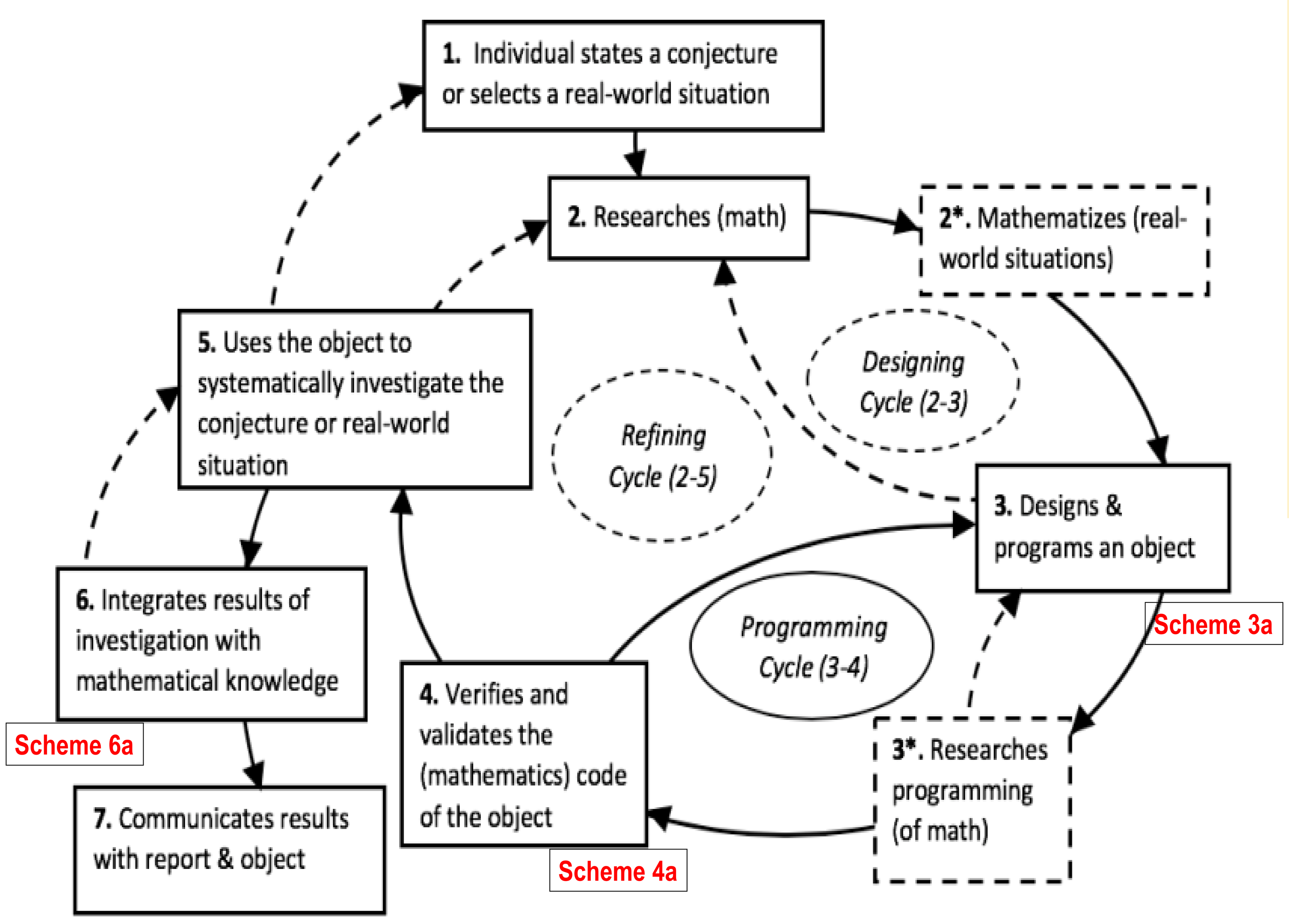


Jim programmed by steps and regularly checked his program as he progressed (**step 4**) while unavoidably needing to debug his math at times (**programming cycle**).

Jim entered large integer input into his program to test the conjecture (**step 5**), and summarized his findings (**step 6**), integrating what he also had learned from his research (**step 2**), in a written report (**step 7**).

**MICA I Project 1**: Conjecture about primes
**MICA I Project 2:** RSA encryption
**MICA I Project 3**: Discrete dynamical system based on a 2-parameter cubic
**MICA I Project 4**: *Students freely choose their topic*

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




## Methodology

• '**Jim**', one voluntary math major student participant (among 6) enrolled in the MICA I course (46 students) in the Year I of our 5-year research study. Before MICA: He had very limited knowledge of computer programming, except for some exposure with LOGO programming.

• **Data collection** for this poster study: Jim's 4 MICA I projects and 4 semi-structured individual task-based interviews following each project submission; a baseline questionnaire and interview; and 10 online weekly lab reflections.

• **Data analysis**: Jim's early data was first analysed for an initial identification and description of schemes, then reorganized in m-, p-, and p+m-scheme types, and ordered according to the development process model.

The whole data was thereafter coded and regrouped in 5 themes.

Using codes pertaining to *perceptions* (TiA) and *strategies (RoA)* themes, the initial table of schemes was then refined and chronologically extended to the whole data.

## Jim's (ongoing) Web of Schemes from his Engagement in 10 labs & 4 Projects

1a To formulate a conjecture (m-type)
  1aa To understand a concept in order to formulate a conjecture (Investigate the math concept)
  1.ab To formulate a conjecture about primes
1b. Assess feasibility of testing the math investigation through programming
2a To look for existing information about a math concept / conjecture online
3a. articulating a math process in the programming language
  3.aa. To ensure that my coded process works for any appropriate input
  3.ab. To Remix from a similar math process I coded previously
3b. To articulate a nested process as a nested loop
3*a. To look for existing information about the coding of a specific task
4a. debugging the program
4b. Validating the programmed math
PC-3c. Write a first draft program
PC-3d. Make an efficient program
  PC-3dd. making the program more efficient to test the mathematical statement with larger numbers
5a. Use programming to test a math conjecture
5b. Use programming to have a math insight of the result
6a. Make sense of program mathematical output in relation to the aimed mathematical investigation
6b. Consider the role that programming technology limitations may play in my interpretation of the math output
7a. Communicating math results in a report (m-type)
  7.aa Communicate in a report a surprising computational mathematical result

## Jim's Schemes: 3 examples

**Scheme 3a** Articulating a mathematical process in the programming language *(p+m type)*

RoA: Start by organizing, on paper, what needs to be programmed until I got the whole picture; Start by 'translating' in vb.net what I would do by hand; Organize the math process as a nested system; Decompose the nested system in individual processes before programming; Code individual processes; Code the nested system incrementally;

TiA: A math process can be seen as a nested system, i.e., made of many parts; To program a nested math process, one can break it down and individually code the smaller parts; Programming and math as systems have embedded layers.

**e.g. JIM (1.8):** «I had to figure out how to program each individual system. This one check for prime. This one is a loop that goes from number input down to two… Once I realized how to do that, I just kind of start programming and basically would add in more levels of complexity as they became necessary. »

**Scheme 4a** Validating the programmed math *(p+m type)*

RoA: Test each part of the "math system" in isolation; Combine parts and test the combination; test the final mathematics result.

TiA: If each part works and the combination process is correct, then the combination should work; if running the program with specific input values of a 'complex' computational mathematical model returns a correct math output, then it is assumed that the program works.

**e.g. JIM (3.24):** «You know, that all, all the individual parts are working, the final thing is working, the final thing looks like what it should be, so therefore it, I can assume it's working.. »

**Scheme 6a** Make sense of program output in relation to the aimed math investigation *(p+m type)*

RoA: Carefully look at collected data and compare to known math results; Accept known math statements based on my observations of the data; Look for pattern in the data to explain the math statements.

TiA: I verify for myself a known math statement by testing with a large number of cases; Patterns in the data may give insights into a proof of the math statement

e.g. **JIM** writes (report 1.5): *While this may not be useful, it does highlight the seemingly random nature of primes, and how strange they can be to follow a pattern for the first 900 000 000 examples, only to abandon it thereafter.*

## Concluding Notes

•Using the instrumental approach led to a first elaboration of the development process model as a composition of goals (and sub-goals), highlighting the **complex structure as a web of schemes** which ramifications are both vertical (zooming in the goals) and horizontal (sequencing during the activity).

• It also led to expose **how the activity** of programming-based pure or applied mathematical investigations **is organized** (through the rules-of-actions, 'RoA') **and why** (through the 'theorems-in-actions, 'TiA')

Social Sciences and Humanities Research Council of Canada
Conseil de recherches en sciences humaines du Canada
Canada

## References

Buteau, C., Gueudet, G., Muller, E., Mgombelo, J., & Sacristán, A. (2019). University Students Turning Computer Programming into an Instrument for 'Authentic' Mathematical Work. IJMEST. DOI: 10.1080/0020739X.2019.1648892

Gueudet, G., Buteau, C., Muller, E., Mgombelo, J., & Sacristán, A. (2020). Programming as an artefact: what do we learn about university students' activity? Submitted to INDRUM 2020, March 2020, Bizerte, Tunisia.

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

Misfeldt, M., & Ejsing-Duun, S. (2015). Learning mathematics through programming: An instrumental approach to potentials and pitfalls. In K. Krainer & N. Vondrova (Eds.), Proceedings of the ninth CERME (pp. 2524–2530). Prague: Charles University in Prague, Faculty of Education and ERME.

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.

Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2016). Defining computational thinking for mathematics and science classrooms. Journal for Science Education and Technology, 25, 127–147.