The constructionist nature of an instructor's instrumental orchestration of programming for mathematics, at university level

Ana Isabel Sacristán, asacrist@cinvestav.mx
Department of Mathematics Education, Cinvestav, Mexico

Marisol Santacruz-Rodríguez, marisol.santacruz@correounivalle.edu.co
Institute of Education and Pedagogy, Universidad del Valle, Colombia

Chantal Buteau, cbuteau@brocku.ca
Department of Mathematics and Statistics, Brock University, Canada

Joyce Mgombelo jmgombelo@brocku.ca
Faculty of Education, Brock University, Canada

Eric Muller, emuller@brocku.ca
Department of Mathematics and Statistics, Brock University, Canada

Abstract

In this paper we describe and analyse the orchestration of an instructor in a Canadian university program, called *Mathematics Integrated with Computers and Applications* (MICA), where students program computer microworlds for mathematical investigations. More specifically, we analyse the particular constructionist elements of the instructor's instrumental orchestration that aim to support students use of computer programming as an instrument for thinking mathematically. We analyse how this instructor conducted his classes in terms of the three components of instrumental orchestration (see Figure 1) and illustrate, briefly, how it is reflected in the activity of one of his students.

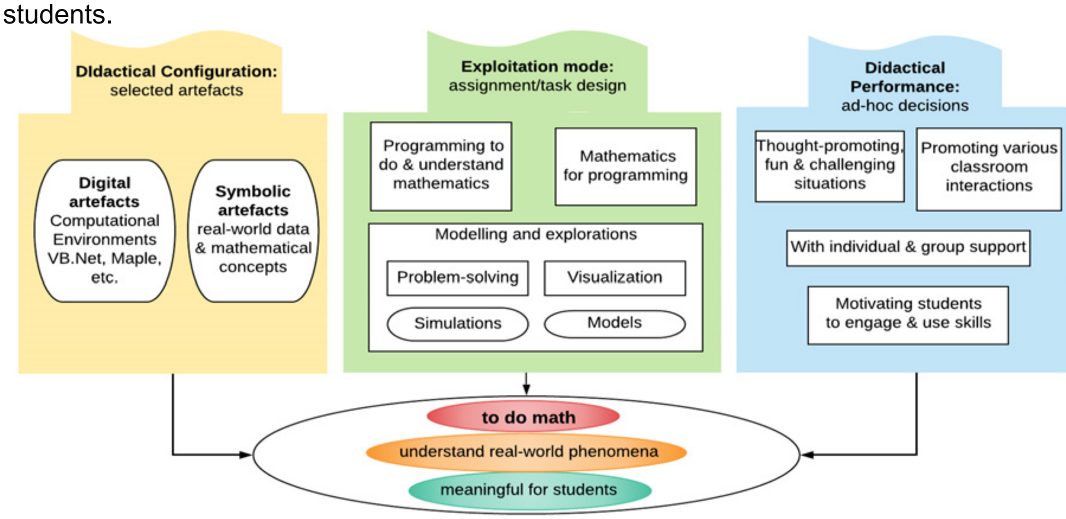

Figure 1. Selected components of a MICA instructor instrumental orchestration related to Constructionism

Keywords

Computer programming; mathematics; university; instrumental orchestration; constructionism.

Introduction

In the past 2018 constructionism conference (Buteau et al., 2018), we discussed the roles and demands on instructors to create constructionist environments in university computer programming for mathematics courses –specifically, the *Mathematics Integrated with Computers and Applications* (MICA) courses implemented, since 2001, at Brock University, Canada (Buteau et al., 2015b). In those courses, math majors and future mathematics teachers learn to design, program, and use interactive environments –called *Exploratory Objects* (EOs)– for the investigation of mathematical concepts, conjectures, and theorems or real-world situations (Buteau et al., 2015a). As explained in the latter paper, those MICA courses fit the constructionist paradigm (Papert & Harel, 1991), by requiring students to design and program the EO computational objects for mathematical investigation and learning.

Since our 2018 paper, we have been analysing the work of MICA instructors in their courses and how they steer their students' investigations and learning, as part of an on-going five-year naturalistic (i.e., not design-based) research project. In that project, we study students' activity and their instructors' work using as framework the instrumental approach (Rabardel, 1995; Guin et al., 2005) and the instrumental orchestration theory (Drijvers et al., 2010), with the following research questions:

- How do students appropriate programming as an instrument for mathematics investigations?
- How do instructors create and orchestrate a learning environment that supports students' instrumental geneses development?

In this paper we focus on the latter through a case study. We look specifically into a MICA instructor's practice in terms of: what artefacts he integrates into his courses, and why; what are the considerations and aims underlying the design of his assignments; what does he expect his students to learn, and in what ways does he support his students. We attempt, specifically, to identify particular constructionist elements in this instructor's instrumental orchestration.

Background: The constructionist design of MICA courses

MICA is a sequence of three semi-annual courses (MICA I-II-III). In Buteau et al. (2018) we described how MICA undergraduate students have to create (i.e., design and code) Exploratory Objects (EOs), including, as final course projects, original EOs for which they select the topic – created individually or in groups of 2-3 students. The EOs can be thought of as mathematical microworlds, which Mavrikis et al., 2008 (cited in Buteau et al., 2015a, p. 138) define as exploratory learning environments that "allow students to explore not only the structure of accessible objects in the environment, but also construct their own objects and explore the mathematical relationships between and within the objects, as well as the representations that make them accessible". That is, EOs are "digital environments to explore mathematics concepts and their relationships represented in diverse forms (in the code and in the interface)" (Buteau et al., 2015a, p. 144).

In MICA I, students learn to create and use mathematical EO microworlds; while in MICA II-III courses, EOs serve to investigate more sophisticated mathematics, with MICA II focusing on applications of mathematics using technology, including simulations and modelling. As can be seen in the MICA guidelines URL (n.d.), the central intent is for students to learn mathematics through creating EOs: Buteau et al. (2015a) connect to constructionism principles (see Appendix), the dominant elements of the MICA courses: 1. The EO mathematics projects require engagement with programming where students learn to do mathematics (learning by making) conducting their own mathematical explorations. 2. Students are empowered when they develop their own strategies and make their own choices for finding solutions to a diverse range of mathematical problems. 3. The course fosters and values students' creativity in mathematical work. And, as explained in Buteau et al. (2018), MICA instructors design and implement a constructionist environment in order to support the intended students' experiences.

Aim of the paper and theoretical elements

In this paper we explore, more in-depth, the constructionist nature of that design, through a case study in which we analyse how a MICA II instructor creates (orchestrates) a constructionist environment. For that, we carried out a literature review to identify some main constructionist principles and ideas from Papert's writings and from the broader constructionist literature.

Main constructionist principles

We organized the identified constructionist principles into four themes, and put them in a table (see Appendix):

- (1) Epistemology, and conceptions of mathematical knowledge and of mathematics;
- (2) Conception of learning and of the role of the student;
- (3) Pedagogy and design; and
- (4) Computer programming and microworlds.

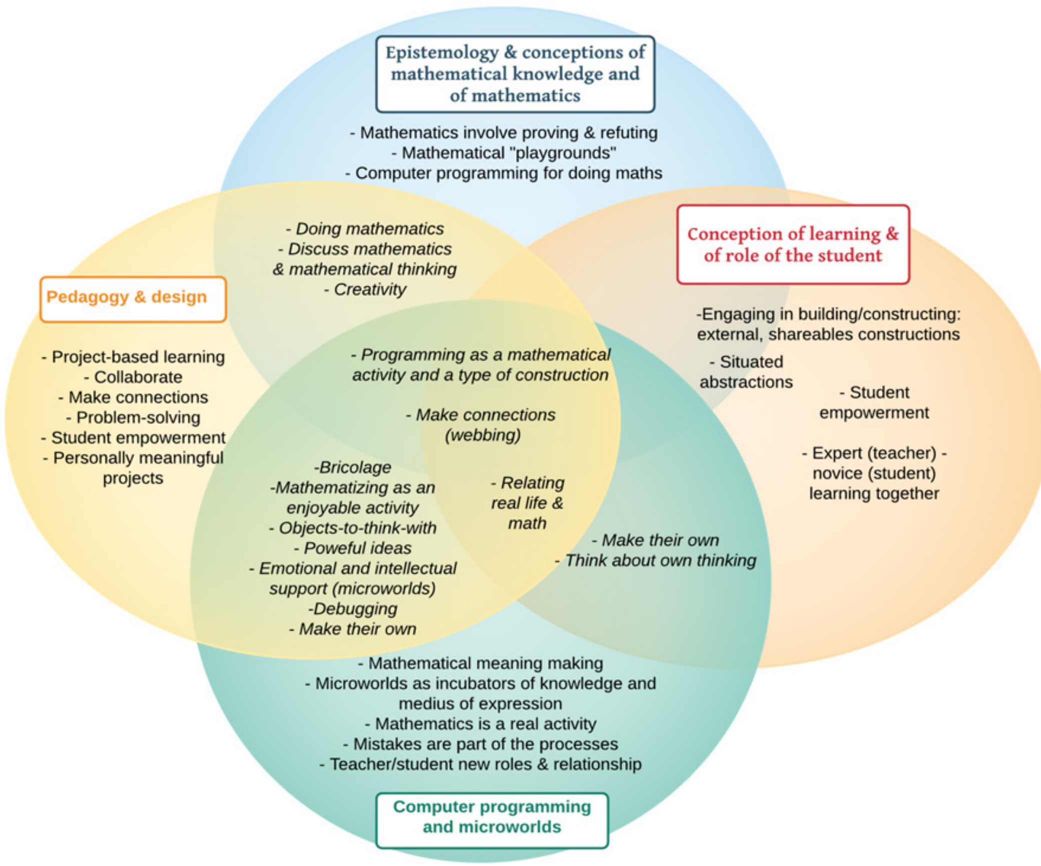

Figure 2. Some main constructionist principles by themes, as presented in the Appendix.

In Figure 2, we present a diagram that represents the main relationships between the constructionist principles that we identified. These principles served as guiding elements for our analysis, for which we used concepts from the instrumental approach (Rabardel, 1995), specifically those of instrumental orchestration (Drijvers et al., 2010).

The instrumental approach and the notion of instrumental orchestration

The instrumental approach, proposed by Rabardel (1995), focuses on how human activity with technology mediates the construction of knowledge: i.e., how people, through a process called instrumental genesis, appropriate an artefact (not necessarily physical), through its use, and turn it into an instrument –a psychological construct developed by a subject over time through the use of the artefact– (Drijvers et al, 2010).

Trouche (2004) considers that students' instrumental geneses may need to be steered by a teacher. He thus proposed the concept of instrumental orchestration to refer to the teacher's intentional and systematic organization and didactic use of various artefacts in the class (including digital ones), that steer students' instrumental geneses. He explains that, through the teacher's orchestration, not only individual aspects are mobilized in the class, but also aspects of a social and collective nature that influence student learning. As an extension to this theory of instrumental orchestration, Drijvers et al. (2010) introduce the idea of didactical performance to explain the different modifications, adjustments and changes to the didactic configurations, which are made in response to the events of the class. Drijvers et al. (2010; p. 215) consider three instrumental orchestration components:

- (i) *Didactical configuration* – "an arrangement of artefacts in the environment, or, in other words, a configuration of the teaching setting and the artefacts involved in it";
- (ii) *Exploitation mode* – "the way the teacher decides to exploit a didactical configuration for the benefit of his or her didactical intentions [it] includes decisions on the way a task is introduced and worked through, on the possible roles of the artefacts to be played";
- (iii) *Didactical performance* – which "involves the ad hoc decisions taken while teaching on how to actually perform in the chosen didactic configuration and exploitation mode."

Using the Instrumental Approach for analysing constructionist activities

Relating the instrumental approach (including the idea of instrumental orchestration) with constructionism is not new. Kynigos and Psycharis (2013) drew connections between constructionism and instrumental theory, highlighting the process of instrumentalisation for the generation of mathematical meanings. Previously, Hoyles and Noss (2004) noted that work with computational tools and the development of learning communities around their use, pointed the way "towards a consideration of the complex process of instrumental genesis, the role of the teacher, and the connection of tool use and traditional techniques" (p. 214). Also Hoyles, Noss & Kent (2004) drew links between the notion of situated abstraction, with those of instrumental genesis and orchestration, elaborating the ways in which technological artefacts can provide means of mathematical expression; they discussed how the process of orchestration at a first level could foster the growth of situated abstractions, by establishing a ''cognitive scaffolding'' for a second level of orchestration taking place over an extended period through a combination of collective activity in the classroom and individual work by students. Later, Trouche and Drijvers (2014) showed how the notion of "webbing" proposed by Noss and Hoyles (1996) –the process of constructing and using connecting structures, such as those provided by digital tools– and the notion of orchestration, are interrelated views on how learners develop a relationship with mathematics through technology, and the role of the teacher to support, through orchestration, such a development.

Analysis of the constructionist nature of an instructor's instrumental orchestration

Data and methodology

In the case study presented in this paper, we analyse the instrumental orchestration of Bill, a MICA II course instructor. As data, we had: (i) From the instructor: course design; EO assignment guidelines; EOs grading rubric; instructor interviews (6 in total) after each assignment (coded A1I to A4I), and after the final projects at the end of the course in two parts (FI1-F12). And (ii) from the students: lab session reflections (LR1-LR9), the completed EOs and accompanying reports (R1-

R5), and individual task-based interviews after each EO (A1I-A5I).

As said above, the analysis of the instructor's orchestration was guided by the main principles of constructionism, which were summarized as keywords (preset codes), as presented in Figure 2 and in the Appendix. Those helped us identify the constructionist elements (marked in bold in the analysis in the following section) in Bill's course orchestration, organised according to Drijvers et al.'s (2010) instrumental orchestration components: didactical configuration, exploitation mode and didactical performance. We also did a preliminary analysis of the effect of Bill's orchestration on one of his students (Kassie –pseudonym); her comments illustrate some ways in which students' responded to his orchestration.

In Figure 1, we present a schematic representation of selected aspects of Bill's instrumental orchestration components, with his main goals, showing some constructionist principles (see Appendix) involved, and the relationships between the elements.

Bill's MICA II course: structure, conception, orchestration and constructionist elements

MICA II's stated objective is for students to learn basic methods of **mathematical modelling** and of "experimental mathematics", with emphasis on computational and algorithmic methods, and computer simulations (Buteau et al., 2015a). We assume that Bill adhered to this course aim and that it guided his orchestration; this course aim can be re-phrased as: to learn **programming for modelling and investigating mathematical problems in real-world contexts**.

Bill's semester-long MICA II course consisted of:

- Four individual programming-based EO mathematical investigation assignments
- A final original project where students, working in pairs (see MICA guidelines URL, n.d.), were given three options to choose from, for the type of object they had to program:
  (i) "an **investigation** of a mathematical problem" (an EO);
  (ii) "a **learning object** (LO) designed to teach/test a mathematical concept";
  (iii) "a 'real world' **application** of mathematics".

Students receive guidelines for each assignment, including, in written form: guidelines with several steps or parts (see as example, the description of Assignment 4, below). Bill's assignments **focus on programming activities, where students investigate mathematical ideas, through simulations, graphics, modelling**. For the programming EOs to be produced, he asks that all the "code should be carefully structured and very easy to read with all variables, functions and subroutines labelled in a helpful way"; with a user-friendly and attractive interface. The proposed topics come from contexts he considers interesting and relevant for his students, but that require a computer to investigate properly: he wants his students to find the **projects meaningful** so that they become engaged in it. The assignments use content resources –"**objects-to-think-with**" (Papert, 1980a)– from daily life, which provide the mathematical programming activities and explorations with **realistic** significance, and may **promote engagement and excitement** in his students. How the task design and context were conceived and presented, is part of the *exploitation mode* of Bill's orchestration.

- Assignment 1 dealt with probability investigations: Students investigated, through programming EOs, four situations, including the Buffon Needle problem where they explored the probability that a needle dropped onto a plane of lines, touches one of the lines (for this problem, students were given a start-up piece of code in VB.net that they could modify and build upon).
- Assignment 2 presented three discrete deterministic dynamics situations to be modelled for investigating and predicting: two population-related phenomena (the spread of a disease; how parents socioeconomic status affects income-earning); and daily return percentages, using real stock exchange Excel data.
- Assignment 3 focused on producing an EO for exploring the chaotic system of the bifurcation diagram for the logistic map (a discrete dynamical system) –Figure 3a– through its bifurcation and fixed points (found through calculations in Maple).

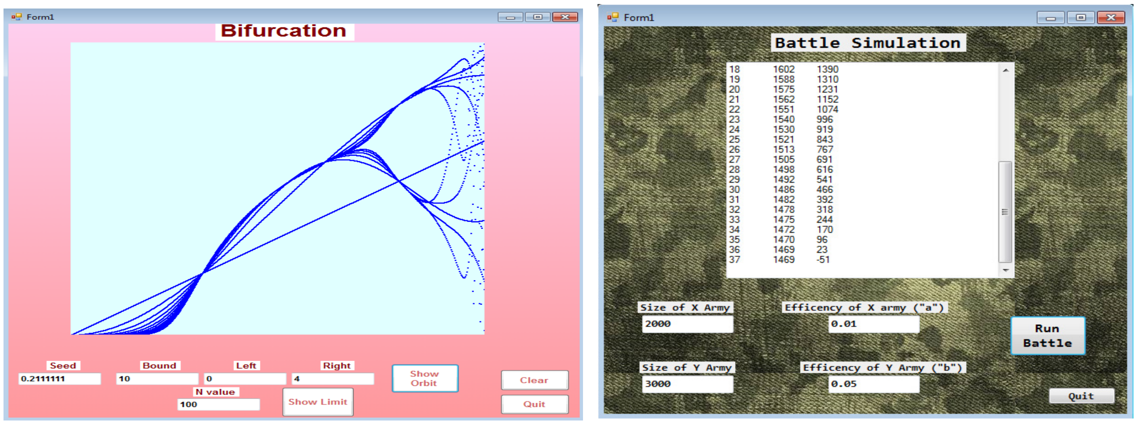

Figure 3. Screen captures of some of Kassie's programs (a) L: Logistics map; (b) R: Battle simulation.

- Assignment 4 focused on the discrete Lanchester equations that model battles (which deal with differential equations and with a modification involving stochastic processes).

Bill and several of his students said that this Assignment 4 was the one that they engaged in, and enjoyed the most. Bill explained some reasons of why he proposed it:

> *I went into the literature of warfare and discovered that at the heart of a lot of military planning are several equations called the Lanchester equations; and it turns out that the world is still using these equations […]; so, this is very current!* (Bill; A4I)

He also explained that there were many mathematical (and scientific) ideas and activities involved in this assignment, such as drawing trajectories, finding probabilities, integrals, differential equations, hyperbolas, and energy conservation, etc. Assignment 4's guidelines outlined the Lanchester equations, provided assumptions, and asked students to create an EO for a "Battle Simulation" (Figure 3b) in order to determine how long a battle would last, depending on the initial size of the participating armies. The program would also show the size of each army on each day of the battle. The guidelines take students through several steps of writing programs to investigate the effects of changing different inputs and aspects; they also ask students several questions, such as making estimations and conclusions on the basis of the explorations using the programs. Furthermore, the guidelines provide a piece of computer code to help students in writing one of the programs.

Bill's *didactical configuration* consists of the artefacts (material or symbolic) that he provides his students with, and their organisation. The "symbolic artefacts" are the topics, problems and mathematical contents that he chose for his assignments, including the data provided and other problem variables. He looks for symbolic artefacts that are engaging: **"**I'm thinking 'what's fun?', 'what can I do?', 'what can I play with?'". He provides **interesting** elements to work with (e.g., **real-world data**, such as from the stock; certain mathematical elements, such as the Lanchester equations, etc.) –i.e., **objects-to-think-with–**, that he himself enjoys (he says: "I teach a topic that I'm interested in, and doing something I care about") and that can lead to **engaging explorations**.

The material artefacts include the computational tools to use in the assignments (mainly VB.Net, but also Excel and Maple). From Bill's interviews, we inferred that he took into account two aspects, in order to decide the selection of the artefacts:

- He considers them effective tools (instruments) for ***doing* mathematics**.
- He considers it important that his students learn to use tools that are normally used in the workplace and in **"real-life".**

For example, with regards to Excel, he explains:

> *I think, how could a math major not be able to [download the past ten years of the stock, and chart its behaviour] in Excel. I also think […] that every math major has to be able to use*

Excel, because this is one of the standard tools in the outside world. It is everywhere. And Excel will do […] almost all the statistics that we teach an undergraduate to do. It has all the tasks, it has everything you could imagine. So, the point [is]: I want them to work with data that they believe is important, and I want them to believe it's important, and I want them to learn how to use Excel. […] We're using another technological tool, we're using the power of Excel. And, of course, the graphics in Excel are extremely powerful, so it's trivial that they learn all the graphics. They learn how to do statistics using Excel. (Bill; A2I)

Behind his selection of the artefacts for his course, Bill shows a conscious reflection for his decisions, that include several constructionist elements. For instance, he added that it was important for his students to learn about the characteristics (e.g., commands) of all the computational environments that were part of the course and that they learn to use them. We infer that he considers those **computational environments as mediums of expression** where students could **do mathematics** and **model**. These are decisions that involve both the *didactical configuration* but also its *exploitation mode*.

In his *didactical configuration,* the selected artefacts are meant to help students explore the complex mathematical ideas of his assignments; they are means to help connect and build relationships between those ideas and previous ones (**webbing**).

And the way in which Bill promotes that building of relationships (the making connections) is through the *exploitation mode:* an **exploration of mathematical ideas** through **modelling,** with students working both independently and **collaboratively.** In this *exploitation mode*, Bill designed programming tasks for **doing mathematics**, that would force students to go beyond what they would normally do in regular mathematics courses. He said: "my first thought in designing […] a lot of the MICA assignments, is to move them as far away as possible from standard mathematics teaching." Thus, Bill selects task contexts using challenging social and science topics related to interesting mathematical ideas, that he thinks students would **enjoy** (so that they become **engaged** in the projects, finding them **meaningful),** but also make them **reflect on their own thinking:**

*I want this to **be part of their life in a real** way […], something they say, "oh, wow! This is something good right now". […] One reason why I like the MICA courses, is because it can be a reflection of themselves. […] I think it's so important that students see a reflection of themselves in their work.* (Bill; FI-1)

His student Kassie's comment below reflects her own **thinking about her learning**:

"Doing mathematics" means to understand why, practice how, and then relate the answer back to the question. This helps to absorb learning something and remember why you did it. If you can understand why, then you will know how to do an advanced question in the future. (Kassie; LR1)

Bill's goal is for his students to use mathematics to **investigate problems, through programming models** (e.g., simulations and models of real-life phenomena). In learning to program, he expects his students to understand better the math, how to apply it and use it to explain the phenomena (**do mathematics**); it implies, **relating programming and mathematics,** as well as **relating various mathematical ideas:**

*[Mathematical concepts are] covered in other courses. My goal […] is to show people the power of introducing computing into **doing mathematics**, teaching mathematics, and so on. […] Part of this course is to try to understand the idea that we can take a real-world situation, and we can distil from that the mathematics, and then take that mathematics and write a simulation based on that mathematics. […] So it's a kind of a two-part process: the real world to the model, write the formal model that we do in the classroom, and then finally the computer simulation that we do in the lab. […] We are looking at the data first, we are building the programs […], observing what happens, and then we go and we try to prove and try to find mathematical evidence for what's going on. (Bill; A1I)*

Bill explains the **back-and-forth process** and relationships between the **real-world phenomena**, its **model** and **mathematics,** in the case of Assignment 1 (the Buffon needle problem),:

> *We can actually toss straws in the computer […], we can write a simulation. And part of this course is to try to understand the idea that we can take a real-world situation, and we can distil from that the mathematics, and then take that mathematics and write a simulation based on that mathematics, so it's a kind of a two-part process, the real world, to the model, right the formal model that we do in the classroom, and then finally the computer simulation that we do in the lab. (Bill; A1I)*

He emphasized this aspect of **making connections** in one of the last interviews:

> *Sometimes the connection is purely through programming skills. […] Sometimes I Bill say: "In this course, what are the kinds of things we've done, what types of techniques have we learned? You're going to show me an amazing way in which someone could learn mathematics using all the resources of a computer."* (Bill; FI-1)

His student Kassie explained the challenges she faced and how, in writing the computer program of the assignment, she had to understand the mathematics and programming concepts and relate them, and how that enhanced her learning:

> *[Some] parts of the assignment, I understood the mathematical parts […] and just needed to do the coding part of each. […] Certain topics and necessary components of the program that I had remembered and were reminded were crucial. This actually helped me to learn more […], which is pretty amazing. I completed the program. […]*
> *…it teaches you a different way of thinking and allows you to expand your horizons in a subject that you may tend to find repetitive. Learning programming is so important in this current society, as technology is a huge part of society and will become an even bigger on in the future.* (Kassie; LR1)

The explorations also take place in environments where students can **try** and **test** their programs**,** simulations and **models,** but where it is **safe to make mistakes, debug** and try again –as Papert (1980) described in his **microworlds'** idea. In this respect, Bill said:

> *How we get people to explore is that we create an environment where they are completely safe to do that and: "Oh, I can just generate anything, and look at anything. And I don't think I did that well enough here…."* (Bill; A3I)

In his *exploitation mode* and *didactical performance*, Bill also takes into account the **affective aspects.** For instance, encouraging students' curiosity and creativity, and being supportive**:**

> *The students […] were fascinated by the warfare stuff. […] I can get them to go a lot further in programming because they're curious, and they believe that this is important. […] Especially in MICA, […] they've written a creative project in mathematics. This is the first time. They need support and encouragement. They really do need that.* (Bill; A4I)

Bill's *didactical performance* involved classroom interactions where he helped students who were "stuck". His **approachable guiding** role **(**that support his students, gives them ideas or suggestions, discusses with them, and ask questions) helps promote students' connections of **programming** and **mathematical** knowledge and skills, without giving out all the information ahead of time, so students have to **play** and **discover**, in a **fun** and **engaging** environment:

> *I mean, it was really fun to do the lab, and their engagement is fun. So, it makes it fun […] because they're doing it, and they're working on it and asking me questions. And it just makes it fun. […] I get to have good conversations.* (Bill; A2I)
> *Sometimes I will say: "In this course, what are the kinds of things we've done?, what types of techniques have we learned?" […] They have to build a mathematical model that makes sense, and [if] that goes terribly wrong […] I have to pull them back from that and say "No, no. no, OK?, On paper, just you and I." And I sit with them with the paper to: "show me, draw me a picture, show me what happens" […]: the model first, the programming second. […] I*

had a lot of nice conversations about [what don't you know about this?" –the math problem]. (Bill; A3I)

Bill guides and supports his students, but lets them be responsible for their work (**student empowerment**), encouraging student exchanges and discussions. In that respect, Kassie said:

It helped definitely because […] everyone has different ideas […] if you look at like someone else's code and like maybe it's not working, then you […] kind of understand more why yours is. (Kassie; LR1)

The Final Projects, in particular, are a **collaborative** process with both **novices and expert working and learning together**:

Right at the beginning when we start the projects, I tell them this: "It's not something you're going to do by yourself. You're working in a team" […] But I tell them I'm going to be an integral part in this process, we're all going to work together. (Bill; A5I)

Concluding remarks

Our analysis focused on identifying the constructionist elements that Bill integrated into his orchestration; we did not attempt to critically appraise Bill's approach in any other respect. We observed that Bill's *didactical configuration* involved mathematical and social considerations, and a web of ideas and actions that provide a creative structure (a webbing) for drawing connections between programming and mathematics. Yet, it is in his *exploitation mode* and *didactical performance* that these provisions manifest in a constructionist way. In Figure 1 (see abstract), we illustrate selected aspects of Bill's orchestration components that aim to support students' programming for investigating math problems in real-world contexts.

The *didactical configuration* involved the computational artefacts that Bill selected for each assignment, the real-world data used in the tasks and the symbolic artefacts at play (e.g., codes, mathematical concepts, representations). The *exploitation mode* relates to his aims and the didactical design of the assignments, which involve modelling, problem-solving, simulations, and explorations; and where "mathematics is used in programming", and "programming is used to do and understand mathematics". And Bill's *didactical performance* aims at supporting and empowering his students, while also taking into account and promoting affective aspects (e.g. fun, motivation, creativity).

Our analysis shows provides insights into how the constructionist elements in Bill's orchestration helped achieve his, and the MICA course's, aim: to engage students in programming for mathematics investigations. This analysis informs our on-going work, where we study MICA students' instrumental geneses of programming, pointing to how the constructionist elements in the MICA approach may promote those geneses.

Acknowledgment

This research is funded by SSHRC grant #435-2017-0367, with ethics clearance REB #17-088.

References

Barabé, G. & Proulx, J. (2017). Révolutionner l'enseignement des mathématiques: le projet visionnaire de Seymour Papert. *For the Learning of Mathematics*, 37, 2.

Buteau C., Muller E. & Marshall N. (2015a). When a university mathematics department adopted core mathematics courses of an unintentionally constructionist nature: Really? *Digital Experiences in Mathematics Education*, 1(2–3): 133–155.

Buteau, C., Muller, E. & Ralph, B. (2015b). Integration of programming in the undergraduate mathematics program at Brock University. In *Online Proceedings of Math+Coding Symposium*, London, ON. http://researchideas.ca/coding/proceedings.html

Buteau, C., Sacristán, A.I & Muller, E. (2018). Teaching in a Sustained Post-Secondary Constructionist Implementation of Computational Thinking for Mathematics. In Dagienė, V. & Jasutė, E. (Ed.). *Constructionism 2018. Constructionism, Computational Thinking and Educational Innovation: conference proceedings*, 528-535.

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

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

Hoyles, C. & Noss, R. (2004). Situated abstraction: Mathematical understandings at the boundary. *Proceedings of Study Group 22 of ICME-10*, 7, 212-224.

Hoyles, C., Noss, R., & Kent, P. (2004). On the integration of digital technologies into mathematics classrooms. *International Journal of Computers for Mathematical Learning*, 9, 309-326.

Kynigos, C. (2015). Constructionism: Theory of Learning or Theory of Design? In S.J. Cho (ed.), *Selected Regular Lectures from ICME 12*. doi: 10.1007/978-3-319-17187-6_24

MICA guidelines URL (n.d.) http://ctuniversitymath.files.wordpress.com/2019/10/mica-ii-assignments.pdf

Noss R. & Clayson J. (2015) Reconstructing constructionism. *Constructivist Foundations*, 10(3): 285–288. http://constructivist.info/10/3/285

Noss, R. & Hoyles, C. (2017). Constructionism and Microworlds. In E. Duval, Sharples M., Sutherland R. et al. (eds.), *Technology Enhanced Learning*, (eds). Cham: Springer. DOI 10.1007/978-3-319-02600-8_3

Noss, R., & Hoyles, C. (1996). *Windows on mathematical meanings: Learning cultures and*

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

Papert, S. (1980b). Teaching children to be mathematicians vs. Teaching about mathematics. In R. P. Taylor (Ed.), *The Computer in the School: Tutor, Tool, Tutee* (pp. 177–196). New York, NY: Teachers College Press, Columbia University. (Original work published in 1972).

Papert, S. & Harel, I. (1991). Situating constructionism. In I. Harel and S. Papert (Eds), *Constructionism*. NY: Ablex Publishing Corporation, 1-11. http://www.papert.org/articles/SituatingConstructionism.html

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

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

Trouche, L. and P. Drijvers (2014). Webbing and orchestration. Two interrelated views on digital tools in mathematics education. *Teaching Mathematics and its Applications*, 33, 193–209. doi:10.1093/teamat/hru014.

Kynigos, C., & Psycharis, G. (2013). Designing for instrumentalisation: Constructionist perspectives on instrumental theory. *International Journal for Technology in Mathematics Education*, 20(1), 15–20.

Appendix: The Main Constructionist Principles and Keywords

Themes	Main constructionist principles	Keywords
Epistemology & conceptions of mathematical knowledge and of mathematics (How is mathematical knowledge constructed and what it means to do mathematics	"Constructionism (in mathematics education) is an epistemology of (i) mathematics as a discipline and (ii) of doing mathematics" (Kynigos, 2015).	Doing mathematics
	"Mathematics is conceived as something that needs to be proved or refuted" (Barabé & Proulx, 2017).	Mathematics involves proving & refuting
	"it is possible for [students] to do creative mathematics (that is to say: to do mathematics)" by creating new branches of mathematics where beginners can play" (Papert, 1972/1980b; p.178) and that "enable beginners to discuss his mathematical thinking in a clear articulate way" (ibid, p. 180)	Mathematical "playgrounds" Creativity Doing mathematics Discuss math & math thinking
	"mathematical thinking involves communication irrespective of the (virtual or real) presence of others. It is the mediating tools – and especially representations – that […] form a major focus" (Hoyles & Noss, 2004; pp. 212-213)	Communicate (discuss) math / math thinking Mediating tools / representations (computational infrastructures)
	"The everyday-life experience of procedures and programming becomes a resource for doing mathematics" (Papert, 1980a, p. 154).	Computer programming for doing math
Conception of learning & of role of the student	"Principles of mathetics – ideas that illuminate and facilitate the process of learning": 1. "relate what is new and to be learned to something you already know. 2. "take what is new and make it your own: Make something new with it, play with it, build with it." (Papert, 1980a, p. 120).	Make connections (webbing) Make their own Play Building/constructing
	"mathematics might be learned in [….] **settings that are real**, socially cohesive, and where experts and novices are all learning [….] **Learning is not separate from reality**" (Papert, 1980a; p. 179).	Real settings / mathematics is a real(-life) activity Expert (teacher) – novice (student) learning together
	Students learn better when they build personally meaningful objects	Meaningful objects/projects
	-For learning to take place, knowledge has to be **(re) constructed** by the subject -"Meanings are created by experiences" (Noss & Hoyles, 1996, p. 46) i.e., when students consciously engage in constructing (e.g., program) The student's learning is facilitated by building external, "tangible" (in the sense of perceivable by the senses), shareable, constructions (Papert & Harel, 1991)	(Re)-construction by the subject Engaging in building/constructing: external, shareable constructions
	Students should have the opportunity to **engage with their own learning** (be conscious of own learning process): i.e. think (e.g. reflect about the problems as well as **think about own thinking** - analyze) (Noss & Hoyles, 2017) **make connections (webbing)** (Noss & Hoyles, 1996). "Meanings are constructed by **action** on virtual objects and relationships. Within a computational environment, some at least of these objects and **relationships become real** for the learner […]: **learners web their own knowledge and understandings** by actions within the microworld, and simultaneously articulate fragments of that knowledge encapsulated in computational objects and relationships - abstracting *within,* not *away from,* the situation." (Noss & Hoyles, 1996; p. 125)	Think about own thinking Make connections/relationships (webbing) through actions (Mathematical) meaning-making Situated abstractions
	The action of the student is central in his/her learning: s/he should have a central and active role and be in charge of the activity.	Active role in learning process Student empowerment
Pedagogy & design (e.g. of the learning environment)	**A project-oriented approach:** gives "time to **talk about it**, to establish a common language with a **collaborator** or an instructor, to **relate it** to other interests and problems" (Papert 1972/1980b; p. 179) "A project is long enough to have recognizable phases—such as planning, choosing a **strategy** of a attempting a very simple case first, finding the simple solution, ***debugging** it*, and so on." (Papert, 1972/1980b; p. 180)	Project-based learning (long-term) Discuss (talk) Collaborate Make connections (webbing) Strategizing Problem solving Debugging
	"the learning processes of an individual must be considered in **a meaningful context** of goal-directed, socially-situated activity." (Hoyles & Noss, 2004; p..212). E.g.: Personally meaningful projects	Personally meaningful projects Student empowerment
	"In a learning environment with the proper **emotional and intellectual support**, [students] can learn […] not only that they can **do mathematics** but that they can **enjoy it as well**". (Papert, 1980): Lovable mathematics (Papert, 1980b)	Emotional and intellectual support Doing mathematics Enjoyable (lovable) math
	Provide "**objects-to-think-with**": objects that children **can make theirs** for "themselves and in their own ways" (Papert, 1980a)	Objects-to-think-with Make their own
	-The learning environment and activities should provide opportunities for **students' explorations** (**"bricolage**): **building, adapting, testing** and **rebuilding,** and allow them to **express themselves** (Papert & Harel, 1991). - Provide opportunities to **adapt** ,have **new ideas**, **test** and **share** them: **discuss** and **collaborate** with others (Kynigos, 2015; Noss & Hoyles, 2017).	Bricolage / exploring Building / constructing (new objects/ideas) Testing

	- Provide a **medium of expression** for students, e.g. the **construction of models**. The actions are expressed in a model (e.g. computer programs or codes, texts, diagrams, narratives, sketches, etc.). (Noss & Clayson, 2015). -**Modelling** promotes the learning of **powerful ideas** through use (Noss & Clayson, 2015); it emphasises the utility of a mathematical concept (Kynigos, 2015)	Adapting /re-mixing Have medium of expression Sharing / Discussing /Collaborating Construction of models / modelling Powerful ideas
	Testing the models Bill lead to a need of **fixing** (**debugging)** (Papert, 1980): that implies analysis and reflection.	Debugging / Analysing Adapting/ Re-mixing
	The teacher's role is to support students' "bricolage" and encourage their creativity to use, explore, build, adapt, test and rebuild their code.	Teacher/student new roles & relationship Creativity
	"the **relationship of the teacher to learner is very different**: the teacher introduces the learner to the microworld in which **discoveries** Bill be made, rather than to the discovery itself." (Papert, 1980b, p. 209). "The **flow of ideas and even of instructions is not a one-way street**," with **a culture that "enriches and facilitates the interaction between all participants** and offers opportunities for more articulate, effective, and honest teaching relationships" (p. 180), where the line between learners and teachers can fade. The […] teacher Bill answer questions, provide help if asked, and sometimes sit down next to a student and say: 'Let me show you something.' […]. Sometimes it is something the student can use for an immediate project. Sometimes it is something that the teacher has recently learned and thinks the student would **enjoy**. (p.179)	Teacher/student new roles & relationship Enjoyable
Computer programming and microworlds	The computer is "a mathematics- speaking being in the midst of the everyday life" that can provide **links between everyday life and fundamental and engaging mathematics**. Computer programming can bring us into **a new relationship to mathematics**": entering into a "**mathematical conversation**," showing possibilities that may have previously seemed "too hard." (Papert 1980a; p. 47) It provides a webbing structure that learners can draw upon and reconstruct for support for creating meanings (Noss & Hoyles, 1996)	Mathematical meaning-making Making connections with mathematics (webbing): "mathematical conversations" Relationship real-life & mathematics
	Programming is a type of construction (e.g. of a model) that allows students to **connect with insights into mathematics**, **science and other fields** (Papert, 1980).**Programming as a mathematical activity**: "**Programming** offers the student **an environment for mathematizing.** When mathematizing familiar processes is a fluent, natural and enjoyable activity, then it is about mathematizing mathematical structures, as in a good pure course on modern algebra." (Papert, 1972).	Programming as a mathematical activity and a type of construction Mathematical meaning-making Making connections (webbing) Mathematizing as an enjoyable activity
	Microworlds are **incubators of knowledge and powerful ideas** (Papert, 1980a): "… a successful microworld is both an epistemological and an **emotional universe**, a place where **powerful (mathematical, or scientific, or artistic) ideas** can be explored; but explored "in safety", acting as an **incubator** both in the sense of fostering conceptual growth, and a place where it is **safe to make mistakes** and show ignorance. And, centrally these days, it is a place where **ideas can be effortlessly shared, remixed and improved**" (Noss & Hoyles, 2017; p. 32). In "computational learning environments […] the process of generating and **expressing meanings** with the available representational infrastructure tends to produce individual and collective understandings and ways of working that are divergent from standard mathematics." […] "computational systems [are] a setting in which new kinds of representation for mathematical objects **generates new possibilities for mathematical expression**" (Hoyles & Noss, 2004; p. 213). In computer programming environments "mathematics is a **real activity** that can be **shared by novices and experts**" where "start interacting mathematically because the product of their mathematical work **belongs to them and belongs to real life**" (Papert 1980a, p. 179)	Microworlds: Contain objects-to-think-with /powerful ideas - incubators of knowledge and powerful ideas (epistemological universes) - emotional universes Experiment / investigate Mistakes are part of the process Testing/fixing/ debugging Adapting/ Re-mixing Mathematics is a real activity Mediums of expression Mathematical meaning-making Make their own Relationship real-life & mathematics Teacher/student new roles & relationship
	The (digital) tools play an important role in learning, since in themselves they are **expressions of mathematical meaning**. In addition, through the **feedback** that digital tools do to students' work, they allow them to **express their ideas**, individually or collaboratively. (Noss & Hoyles, 2017)	[Computational environment / digital tools are] mediums of expression Feedback [of computational environment]
	Computer feedback: impersonal, leads to **debugging** and reflection. (Papert, 1980): It allows students to **explore how** they **think (themselves)** and understand their mistakes (understand what went wrong, why and how to fix it). (Papert & Harel, 1991)	Feedback [of computational environment] Debugging / fixing / Analysing Think about thinking