Professional Development in Computational Thinking for teachers in Colombia

Alejandro Espinal
Department of Education.
Universidad del Norte
Barranquilla, Colombia
espinal@uninorte.edu.co

Camilo Vieira
Department of Education
Universidad del Norte
Barranquilla, Colombia
cvieira@uninorte.edu.co

Alejandra J. Magana
Computer and Information Technology
Purdue University
West Lafayette, United States
admagana@purdue.edu

***Abstract*—This Research to practice work in progress paper explores teachers' experiences about a professional development program focused on discipline-based computational thinking (CT). The goal of the program is to integrate computational practices into disciplinary learning environments at the K-12 level in Colombia. To promote this integration and to explore teachers' conceptions and possible scenarios to integrate CT in this context, the research team designed, implemented, and assessed a 20-hour professional development program with 21 teachers from public middle and high schools in Antioquia, Colombia. The program introduced the concepts and practices of CT and discussed why these are relevant for students to learn. A use-modify-create instructional sequence was used as the pedagogical approach to scaffold participants' learning process. At the end of the workshop, the teachers completed a survey regarding their experience in the program. Preliminary results indicate that the participating teachers increased their knowledge and interest to integrate computational thinking practices into their disciplinary learning environments.**

Keywords—computational thinking, k-12, Professional Development, teachers

I. INTRODUCTION

Computational thinking (CT) is a set of practices, concepts, and methods from computer science that support problem-solving and representation of complex phenomena across disciplinary areas [1]. These practices include abstraction, decomposition, algorithm design, automation, and debugging among others. Professionals from any discipline can use these practices for modeling complex phenomena, representing knowledge, and processing large datasets. These practices can support student learning within disciplinary learning environments, while the disciplines provide a meaningful context to develop computational thinking skills [2].

Although there is some progress at the undergraduate level to integrate computational thinking across disciplinary curricula [3], [4], there is no clear approach to integrate computational thinking across primary and secondary school curricula [5]. In Colombia since 2019 started a program called Coding for Kids (CFK). The goal of CFK was to develop student computational thinking skills in public schools. The program included professional development workshops and mentoring for teachers to integrate lesson plans with unplugged and plugged activities using the micro:bit device and the MakeCode programming language. But at the moment, are not clear way to integrate CT into Colombian Curricula.

Lee and colleagues [6] suggest that one of the existing challenges in the integration of computational thinking into the K-12 curriculum is the lack of teacher professional development programs focused on computational thinking. For instance, the International Federation of Information Processing (IFIP) discussed the role of computational science in the curricula. According to Webb [7], one of the emergent themes was professional development as a major challenge in different countries. Participants from these countries mentioned three points about this challenge: (a) Existing teachers who have taught a different curriculum may not have sufficient technical knowledge and skills, (b) The teachers' pedagogical content knowledge (PCK) [8] has not been developed about the new curriculum content, and (c) few new Computer Science graduates are coming into teaching.

In 2020, the research team designed and implemented a professional development program in Computational Thinking with K-12 disciplinary teachers in Antioquia, Colombia. This work in progress paper explores disciplinary teachers' reactions and experiences about the program, and the opportunities they identify to integrate this set of practices into different areas. The guiding research question is: *RQ1 - What are the opportunities, conceptions, and challenges that the teachers face to integrate CT into their learning environments?*

II. PEDAGOGICAL FRAMEWORK

The "Use-Modify-Create" (UMC) supports student learning by first "Using" a given artifact, "Modifying" an existing one, and then eventually "Creating" new ones [9].

The UMC progression has been widely used as a means to scaffold student engagement in CT [9]. The UMC progression started to be used within three contexts: (a) Computer modeling and simulation, (b) robotics, and (c) data science. The progression has now been used in new contexts including computer programming, machine learning, and computational science. We argue that this progression can also be used to scaffold disciplinary teachers in professional development programs involving computational thinking. Since most disciplinary teachers in Antioquia are novices to computational thinking, exploring, using, and modifying existing lesson plans may help them to develop the required schemata to design new activities.

III. CONCEPTUAL FRAMEWORK

To assess the professional development program, we used an adapted version of the Kirckpatrick's Training Evaluation Model [10]. This model is an approach to measure the impact of a training program.

This framework includes 5 levels to be considered for assessing the program: (1) Reactions: This level is assessed immediately after the workshop, and looks into participants' satisfaction, as well as their reactions or attitudes toward the learning experience, (2) Learning: This level of assessment aims to answer the question -what did participants of the professional development program learned? (3) Support: This level identifies participants' perceptions about the support they find in the training program, the resources accessible to them, and the transferability of those to their own context/institution, (4) Use of Knowledge and Skills: The framework proposes that this level can be assessed annually starting a year after the workshop, and determines the impact of the learning in their practices within their context/institution, and (5) Research Outcomes: The framework proposes that this level assessed annually starting a year after the workshop as the level four. This level explores whether the training leads to final results including increased collaborations, increased student performance, etc. [10]. In the context of this work in progress paper, we will explore participants' Reactions, to look at opportunities, conceptions, and challenges towards the learning experience.

IV. THE LEARNING EXPERIENCE

The online learning experience was designed as a 20-hour workshop for in-service disciplinary teachers. The learning goals for this program are: (1) Recognize the importance of integrating computation in different areas, (2) Explain how the CT can be integrated into different areas, and (3) Design a learning environment where CT is integrated within a disciplinary context.

The first part of the program focused on discussing existing initiatives to establish a CT curriculum in different countries, including the United Kingdom, Australia, and the United States. The goal of this section was to highlight the importance of integrating CT into K-12 curriculum in the global context.

After this, we introduced a set of lesson plans of Natural Science, Social Sciences, and Physics for the participating teachers to explore. The lesson plans contained a set of activities using the progression "Use-Modify-Create" to support student learning. For example, the Natural Science lesson plan had the structure presented in Table I. We adapted this progression to also use it with teachers. The teachers start by exploring sample lesson plans, and explain them to each other (i.e., use). They should then introduce a change into the lesson plan to extend it or adapt it to a specific context (i.e., modify). Then, as the final project of the program, the participating teachers submitted a new lesson plan to integrate computational thinking skills into their disciplinary courses.

TABLE I. LESSON PLAN – NATURAL SCIENCE

Activity	**Using NetTango to Model Natural Selection (https://tidal.northwestern.edu/nettango/)**
Learning goals	1. Describe the effects that can have the variation, the inheritance, and the selection in the Natural Selection process.
	2. Model behaviors of species to understand the implications of
	different variations in the survival of different species.
UMC progression	Use: Explore NetTango simulation to start understanding the Natural Selection phenomena.
	Modify: Vary initial parameters in NetTango simulation and add new variables.
	Create: Build a simulation to benefit a species with specific characteristics.

V. METHODS

The program was supported by the Secretary Education of the governors' office in Antioquia, which invited teachers from public schools to participate. We first delivered a Webinar to introduce the concept of Computational Thinking, and invited teachers to enroll in the program.

In total, we had a group of 21 participants from different cities and towns of Antioquia. These participants were divided into two groups: one with teachers of secondary school level, and another one with the teachers of primary school level. *Table III* displays the distribution of teaching levels and disciplines for the participants along with other demographic information.

TABLE II. PARTICIPANTS TEACHING LEVELS AND DISCIPLINE

Discipline	**Gender**		**Level**		**Area**	
	Male	**Female**	**E**	**S**	**Rural**	**Urban**
Informatics & Technology	0	2	1	1	0	2
Natural Science	2	0	0	2	0	2
Social Science	1	2	1	2	0	2
Maths	1	2	0	3	0	3
All the areas	4	7	7	4	9	2

**E: Elementary - S: Secondary*

VI. Results and Discussion

The results describe the participants' reactions and experiences with the program. The assessment at this level was intended to provide useful information for workshop designers and administrators in order to further improve the experience.

At the end of the workshop, 21 teachers completed a survey regarding their experience in the program. We qualitatively analyzed the teacher's answers related to their reactions or attitudes toward the learning experience. The evaluation questions and the respective answers are presented with the qualitative categories that summarize these experiences.

1. Please describe, which were the most relevant aspects in this workshop and how will you use that in the future?

The participants' responses to these questions mostly focused on three themes. The most common ones included the identification of technological tools to develop student computational thinking, and the pedagogical strategies to support this process. The program also helped them to better understand what computational thinking is.

TABLE III. Answers question 1

Category	#	Quotes
Technological tools	*15*	*"The most relevant for me, was use computational simulations to understand different phenomena"*
Pedagogical practices	*13*	*"The most important was the implementation of new pedagogical strategies in the classroom"*
Content Knowledge	*2*	*"The most relevant aspect was understand what Computational Thinking is and how use it in the classroom"*

2. Please describe, which topics do you want to deepen and why?

In this second question, the responses are evenly distributed among more tools for supporting student learning, and more practice on designing learning environments.

TABLE IV. Answers question 2

Category	#	Quotes
Programming tools for novice learners	*8*	*"Learn more about coding using block-based and code programming language"*
Unplugged activities	*7*	*"The unplugged activities are fundamental to the schools in rural area"*
Computational tools for education	*4*	*"Simulations are useful to teach in different areas and thematics"*
Learning environments	*3*	*"The design of learning environments are important to teachers"*

3. Please tell us, what opportunities do you identify to improve the workshop.

The main recommendations to improve the workshops include: additional time, more examples for different disciplines, and having the program in-person instead of online.

TABLE V. Answers question 3

Category	#	Quotes
None	*8*	*8 teachers affirm that they do not have suggestions to get better at the workshop.*
Workshop duration	*4*	*"More time to work in theoretical and practical components"*
Discipline-based computational thinking.	*4*	*"Use computational science to teach different areas"*
In-person program	*2*	*"Presentiality in the program to use computational tools to experiment during the workshops"*

4. Do you consider that your experience in this workshop has changed your mind about the importance of integrating Computational Thinking in your courses?

All participants suggested that they changed their vision about the importance of integrating computational thinking in their courses, but these changes included different technological tools, pedagogical practices, and a better understanding about computational thinking.

TABLE VI. Answers question 4

Category	#	Quotes
Technological tools	3	*"Yes, this workshop change my mind showing that it can be developed and strengthened from activities in computer simulators"*
Pedagogical practices	7	*"Yes, because it gave me new ideas and practices to play a better role as a math teacher"*
Content Knowledge	11	*"Before, my perspective associated these types of areas only with technology and information technology. After this workshop, I learned that it can be taken to any area of knowledge."*

With the goal of exploring teachers' conceptions and possible scenarios to integrate CT, we present the following question:

RQ1 - ¿What are the opportunities, conceptions, and challenges that the teachers face to integrate CT into their lectures?

Our results, and the different categories that emerged, can be interpreted under the lens of Technological, Pedagogical, and Content Knowledge (TPACK). TPACK is a theoretical framework for understanding teacher knowledge required for effective technology integration for a specific content. TPACK introduces the relationships and the complexities between all three basic components of knowledge (technology, pedagogy, and content) [11]. This framework can be useful to analyze the next level of the Kirckpatrick's framework: learning.

According to these results, it can be inferred that teachers identified opportunities to understand and use pedagogical strategies to integrate CT, and the use of computational tools for education. Other opportunities they identified were the use of Unplugged Activities related to Computer Science. Almost 80% of the schools are located in rural areas and do not have an Internet connection, so coding and unplugged activities may be appropriate for this rural context.

In this case, the use of the UMC progression to support teacher learning demonstrated a positive experience with the workshop, which suggests a theoretical framework for assessing the next levels. Nonetheless, the participants would have wished to learn more about programming tools for novice learners in order to create their own resources and activities for specific school contexts.

VII. Conclusion and Next Steps

This work in progress paper demonstrated teachers' experiences with a professional development program in CT. An identified limitation in this step, it that only reactions were evaluated, but as part of future work, the following levels of Kirckpatrick's framework will be analyzed: (2) Learning, (3) Organizational Support, (4) Use of knowledge and skills, and (5) Research Outcomes. According to the results of this study, TPACK will be used to analyze the teachers' learning.

Also, the research team will use the UMC progression with other professional development programs in computational thinking to better understand how this progression supports teacher learning.

VIII. Bibliography

[1] J. Wing, Computational thinking—What and why?, The Link: News from the School of Computer Science. vol. 6, pp. 20 - 23, 2011.

[2] D. Weintrop, E. Beheshti, M. Horn, K. Orton, K. Jona, L. Trouille, Defining computational thinking for mathematics and science classrooms, Journal of Science Education and Technology, vol. 25, no. 1, pp. 127-147, 2016.

[3] P. Sengupta, K. Krinks, D.B. Clark, Learning to deflect: Conceptual change in physics during digital game play., Journal of the Learning Sciences, vol. 24(4), p. 638–674, 2015.

[4] C. Vieira, A. J. Magana, A. Roy, & M.L. Falk, Student Explanations in the Context of Computational Science and Engineering Education., Cognition and Instruction, pp. 1-31, 2019.

[5] F. Heintz, L. Mannila, & T. Färnqvist, A review of models for introducing computational thinking, computer science and computing in K-12 education., Frontiers in education conference (FIE), pp. 1- 9, 2016.

[6] I. Lee, F. Martin, J. Denner, & B. Culter, Computational thinking for youth in practice, ACM Inroads, 2011.

[7] M. Webb, N. Davis, T. Bell,Y.J. Katz, N. Reynolds, D.P. Chambers, & M.M. Sysło, Computer science in K-12 school curricula of the 2lst century: Why, what and when?, Education and Information Technologies, 22(2), 445-468., 2017.

[8] L. S. Shulman, Those Who Understand: Knowledge Growth in Teaching, Educational Researcher, Vol. 15, No. 2, pp. 4-14, 1986.

[9] F. Martin, I. Lee, N. Lytle, S. Sentance, Extending and Evaluating the Use-Modify-Create Progression for Engaging Youth in Computational Thinking, SIGCSE '20: Proceedings of the 51st ACM Technical Symposium on Computer Science Education Pages 807–808, 2020.

[10] D. Kirkpatrick, The Four Levels of Evaluation., Brown S., Seidner C, editors. Evaluating Corporate Training: Models and Issues. Norwell, MA: Kluwer Academic; pp. 95–112., 1998.

[11] P. Mishra, & M.J. Koehler, Introducing technological pedagogical content knowledge., Annual meeting of the American Educational Research Association (pp. 1-16)., 2008.

[12] N. Lytle, V. Cateté, D. Boulden, Y. Dong, J. Houchins, A. Milliken, & T. Barnes, Use, modify, create: Comparing computational thinking lesson progressions for stem classes., In Proceedings of the 2019 ACM Conference on Innovation , 2019.