An instructional framework, model lessons, and professional learning program for science standards-aligned computing in 4th-12th grade classrooms

K. Virginia Lehmkuhl-Dakhwe
College of Science
San José State University
San José, USA
virginia.lehmkuhldakhwe@sjsu.edu

***Abstract*— This *Innovative Practice, Work in Progress* paper outlines an Instructional Framework for integrating computing into science instruction in 4th-12th grade classrooms. It presents a model lesson example and results from two years of implementing a *Professional Learning (PL)* program for teachers developed and offered by the STEM Education Office of the College of Science at San José State University and teachers from eight high need school districts. The program model focuses on increasing teachers' skills and improving practices related to Scientific Computational Modeling as the Next Generation Science Standards and Science and Engineering Practices 2 (develop and use models), 4 (analyze and interpret data), and 5 (use math and computational thinking) are implemented.**

Keywords—science, computer programming, modeling, computational thinking, Next Generation Science Standards

I. INTRODUCTION

This Innovative Practice, Work in Progress paper outlines an Instructional Framework, model lessons, and a professional learning program for integrating computing into science instruction in 4th-12th grade classrooms.

The Next Generation Science Standards (NGSS) offer a potentially transformative vision for science education. They provide new opportunities for K-12 teachers to engage students in computing and technology-rich classroom instruction. These instructional approaches can be routinely applied to help students develop the disciplinary knowledge and skills, career awareness, critical thinking, reasoning, and communication skills needed for persistence in Science, Technology, Engineering and Math (STEM) and Computer Science (CS) education pathways and workforce sectors. They also offer students new opportunities to develop Computational Thinking (CT) skills. CT involves solving problems, designing systems and understanding human behavior by drawing on the concepts fundamental to computer science [1]. CT has been argued by some to be at the core of all modern STEM disciplines [2].

The NGSS articulates K-12 science standards that will require all students to engage in scientific inquiry that integrates the NGSS Cross Cutting Concepts (CCC), Science and Engineering Practices (SEPs), and Disciplinary Core Ideas (DCIs) [3]. For example, the integrated teaching of SEPs 2 (develop and use models), 4 (analyze and interpret data), and 5 (use math and computational thinking) can prepare students to engage in *Scientific Computational Modeling (SCM)*. We define SCM as the modeling of scientific phenomena using computational tools and skills.

Translating the NGSS and SCM to classroom practice will require teachers with the requisite skills, support, infrastructure, positive self-efficacy, and, in some cases, reform-minded identities [4], to teach in a standards-aligned fashion. To help prepare teachers for these endeavors, we've developed an SCM Instructional Framework, model lessons and a Professional Learning (PL) program that focuses on SCM. These efforts involve staff and affiliated faculty of the STEM Education Office of the College of Science at San José State University (SJSU) and teachers from eight high need school districts. Our longer-term vision is broad adoption of the model in which all K-12 students in target high need areas will have equitable access to developmentally appropriate instruction in NGSS-aligned science, CS, SCM, and career exploration, as part of their classroom work, and where all educators will possess the skills necessary to provide this instruction. We expect the intermediate outcome of teachers embracing and implementing in their classrooms this SCM approach will result in the intended long-term outcomes of teaching science in a manner that is consistent with the NGSS, improving students' awareness and motivation to pursue STEM- and CS-focused work, and develop conceptual and epistemic learning outcomes relevant to SCM and CT.

Goals for this paper are to describe the SCM Instructional Framework and PL opportunity and report some preliminary data from investigations aimed at determining the effectiveness of the intervention in improving teachers' confidence in teaching CS and understanding of the connection between computer programming and modeling and CT, essential components of the NGSS. These are data from pre- and post-surveys that reflect teachers' changing perspectives on the importance of CS as it relates to student learning and to the NGSS and their confidence in and willingness to teach CS.

II. SCIENTIFIC COMPUTATIONAL MODELING INSTRUCTIONAL FRAMEWORK

A. Scientific Computational Modeling Instructional Framework and Model lesson

Our SCM Instructional Framework is founded in Christina Schwarz's ([5]) *Engage-Investigate-Model-Apply* (EIMA) framework that "incorporates inquiry and modeling components to further emphasize, clarify, and incorporate the

978-1-5386-1174-6/18/$31.00 ©2018 IEEE

scientific practice of modeling-centered scientific inquiry" (p. 731). We've adapted this framework to accommodate the study of a relevant computer model during the course of lesson instruction (Figure 1). Many of our computational models are constructed using *Scratch* (https://scratch.mit.edu/) that offers a visual programming environment with reasonable supports that are appropriate for students at the middle school level. Scratch is familiar to many students and teachers in our local area, has robust online support, is aesthetically pleasing and relatively easy to use, and offers both an offline editor and an online environment that allow teachers to create Chromebook-compatible accounts for their students. We note that while much of our work has been conducted using Scratch for reasons noted above, other languages and programming environments could be used.

In our adaptation of the EIMA framework [5], during the Engage-Investigate-Model phases, students engage in and elicit their prior knowledge about a topic, career, and/or NGSS Phenomenon [5, 6] and are also introduced to an aesthetically pleasing and relatively simply programmed computer model (Figure 1, Visuals 1 & 2).

formulate their own ideas about alternative or new iterative processes that would improve and/or adapt the model. Students then refine a model according to their changing understanding of the science concept.

During the *Apply* phase of our SCM Instructional Framework, students apply and adapt the computer model to communicate ideas that are of interest to them (Figure 1, Visual 3). As students refine and adapt the model based on their clearer understanding of the scientific content, they answer new questions and can represent other related phenomena. For instance, applying their programming skills and knowledge of waves, students might communicate how in modern sonography, analog signals are converted into digital signals and that a sampling process is involved. Students could alternatively generate a model to investigate *sonar* and a related career, such as a *Geoscientist*.

B. SCM-Associated Enhancements over current classroom practices

Our SCM Instructional Framework involves iterative processes that offer specific enhancements over current typical classroom practices related to modeling. Chief among them are

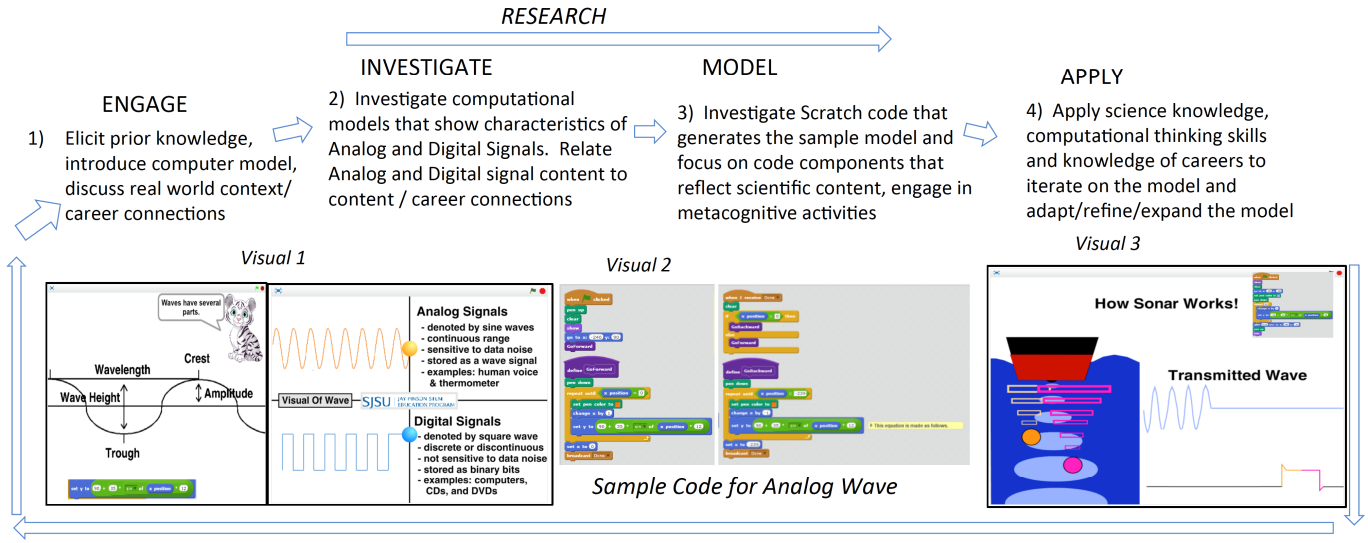

Figure 1: Summary of SCM Instructional Framework and model lesson focused on waves. The Engage, Investigate, Model, Apply phases are based on work by Christina Schwarz, see reference [5].

For example, a model could be presented that enables students to focus on capturing essential similarities and differences between *digital* and *analog* waves. After a brief orientation to how the Scratch program mechanics work, students interact with the model by changing the code to vary wave *amplitude* and *frequency*. Students review how and where the science content is reflected in the computer program and output. As students' understanding of a science concept deepens, they can pose questions about the visual design, application, content focus/emphasis, and function of the model. Students then *reflect scientifically and engage in reading and research activities* to further investigate applications of the model. They then engage in metacognitive strategies, for example, self- and group-explanations, reflective prompts, etc., to further check for understanding [6]. They can then start to

that, rather than using low-tech models in classrooms, digital tools, computer programming, and algorithms are routinely integrated into science instruction. Students focus on exploring, evaluating, and revising a computational model where the computer program used to generate the model can be easily accessed and changed. Students can both use the models to understand, communicate and predict scientific phenomena and information and they can adapt the model to communicate deeper conceptual understanding and ideas that are of interest and relevance to them. In so doing, they enhance their understanding of the scientific phenomena, while deepening their understanding of the computer programs, algorithms, and related output and object behavior. They can then apply these skills to model other phenomena. The Framework is also sufficiently flexible that teachers may implement this approach

in both NGSS-focused classes and in classes where other or new science standards are used.

III. Scientific Computational Modeling Professional Learnng Program

In early investigations of program feasibility, a key area of need for broad SCM implementation included providing a PL program for teachers that situates computing and modeling within interesting, relevant, and practical contexts and that makes explicit the connections between computer programming and the instruction of SEPs 2, 4, & 5. We therefore developed an SCM-focused PL program for teachers. It typically includes a 40-60 hour Summer Bootcamp, observed classroom implementation of SCM lessons, and academic year, Community of Practice (CoP) sessions (15 hours).

During the Summer Bootcamp, teachers are first immersed in foundational and contextualized training in computing and computer programming. Program instructors, including a lecturer from our CS department, and collaborators from science disciplines, lead a series of relatively structured activities during which teachers study and practice CT- and SCM-related content, resources, research and skills. They study the SCM Instructional Framework (Figure 1), work through existing model lessons and study associated computer models, and other resources. Teachers are introduced to fundamental programming constructs such as data structures and algorithms, iterations, functions, and other concepts. They engage in investigations that demonstrate the broad applications of computing and computer programming in various common endeavors and that reinforce NGSS skills and knowledge. They study CT skills, SEP 2, 4 & 5 and related modeling skills. They also investigate resources relevant to STEM workforce development that will guide them in integrating this instruction into their lessons. They also study NGSS assessments and become familiar with instruments to assess CT skills.

Once teachers have developed foundational CT and SCM skills and competence, they reflect on their own instructional practices and begin to consider short and longer-term goals for student learning and development, particularly with reference to SCM, CT, and SEPs 2, 4, & 5 and related NGSS. They reflect on their students' characteristics and preparation such as their prior exposure to computer programming, tendency to persevere through challenging tasks, their current interests, and academic preparation. Teachers are also encouraged to consider this technology-focused intervention as it relates to available computing infrastructure at their schools and explore strategies for leveraging existing infrastructure. They work through existing lessons and reflect on their usability, engage in processes of deconstructing SCM content and skills to identify and leverage connections to other content areas and skills. They also consider student thinking at different stages of the SCM process and begin to develop and refine learning supports intended to construct greater, deeper, and more sophisticated understanding of SCM skills. They also identify specific career connections relevant to the SCM lesson focus.

Once teachers self report sufficient comfort with foundational CT and SCM skills, they individually or in groups write detailed instructional plans for lesson(s) that they will implement in their own classrooms during the academic year. Lessons developed can range from application of very basic computer programming and modeling concepts in a simple classroom context to considerably more sophisticated programming techniques, developed by those teachers with a background in computer programming.

Written lessons include an instructional plan, materials, assessments, extensions, career connections, specific CT and SCM skill foci, questions and reflections, and related resources. Lessons also include an appendix that provides detailed instructions on the technical and computer programming aspects of the computational model/visual. Lessons are also tailored to the computing infrastructure available at schools and provide recommendations for leveraging existing resources. As part of the lesson development process, teachers also create observation / data collection protocols that will be used during classroom observation sessions and identify the student work to be collected, graded, and analyzed.

Throughout the lesson development process, groups of teachers consult with science and computer programming content experts so that the rigor of lessons is ensured. Lesson drafts submitted by teachers are reviewed and edited carefully by a team of content experts. This review is conducted in consultation with the teachers.

At the conclusion of the PL Summer Bootcamp, teachers present their lessons a "showcase of SCM lessons" attended by colleagues and district leadership and engage in planning activities for broader dissemination of the lessons and work throughout districts and schools.

During the academic year, teachers implement at least 2 hours total of observed SCM-focused lessons in their classrooms. They test and refine lessons, assess their students' learning, and report on their findings/experiences at PL CoP sessions. All teachers are also expected to present a model lesson at one 3-hour Leadership CoP meeting each year for educators and administrators. These Leadership CoP sessions focus on building district-wide PL systems focused on SCM- and NGSS-focused/aligned instruction and developing a robust SCM PL Network.

IV. Considerations Related to Equitable Instruction

Central to the SCM PL are opportunities for teachers to reflect on and implement practices to best serve our local high-need community of students. Some key considerations/ practices include creating a community of teacher who prioritize rigorous science instruction, starting in early grades; studying and implementing research-based strategies for creating equitable learning environments; using "balanced" pedagogical approaches in classroom instruction; providing support for academic language development in science; and contextualizing classroom work within our local "STEM education ecosystem" for ongoing student engagement.

V. SCM Professional Learning Program Implementation and Preliminary Findings

To date, 40 4th-12th grade teachers have participated in over 80 hours of intensive PL and over 700 4th-12th grade students have participated in over 2700 hours of SCM work in their classes. Over 150 teachers have participated in CoP and other extension activities related to this project.

The preparation and background of participating teachers has been very diverse with 50% reporting no prior training in computer science prior to enrolling in the PL program. Twelve percent reported having taken some high school- or college-level computer programming courses. Eighteen percent of respondents reported having a Bachelor's degree in science, math, or engineering, and 12% reported "other" preparation.

An ongoing goal of our research and data collection plan has been to determine the extent to which the intervention changes teachers' perspectives on the importance of CS as it relates to student learning and teachers' understanding of the connection between computer programming, scientific modeling and CT, essential components of the NGSS and SCM. Here we report preliminary findings from surveys administered to program participants prior to and immediately following the Summer Bootcamp. Figure 2 shows a summary of a subset of survey question responses from pre- and post-surveys.

Figure 2, A) - D) report survey data results that show teachers changing confidence in teaching CS and their perspectives on the importance of CS in education. Notably, on the post-survey, 87% of respondents strongly disagreed with the following statement: *I don't think computer science/programming is important for students to learn,* compared with 63% on the pre-survey. 71% of respondents agreed or strongly agreed with the following statement: *I am confident that I can teach an integrated science/ computer science/ programming course,* compared with 55% in the pre-survey. 64% of respondents agreed or strongly agreed with the following statement: *I am confident that I can teach computer science/programming to other educators,* compared with 45% in the pre-survey. Finally, 50% of respondents strongly agreed with the following statement: *I want to teach computer science/programming in future years,* compared with 27% in the pre-survey.

Figure 2, E) - G) report survey data results that reflect teachers' self-reported familiarity with CT and modeling, essential components of SCM. 79% of respondents agreed or strongly agreed with the following statement: *I am confident that I can explain the definition of a model,* compared with 54% in the pre-survey. 57% of respondents strongly agreed with the following statement: *I agree that computer programming can be an effective tool for developing scientific models,* compared with 45% in the pre-survey. Finally, 71% of respondents strongly agreed or agreed with the following statement: *I am confident that I can explain components of computational thinking*, compared with 27% on the pre-survey.

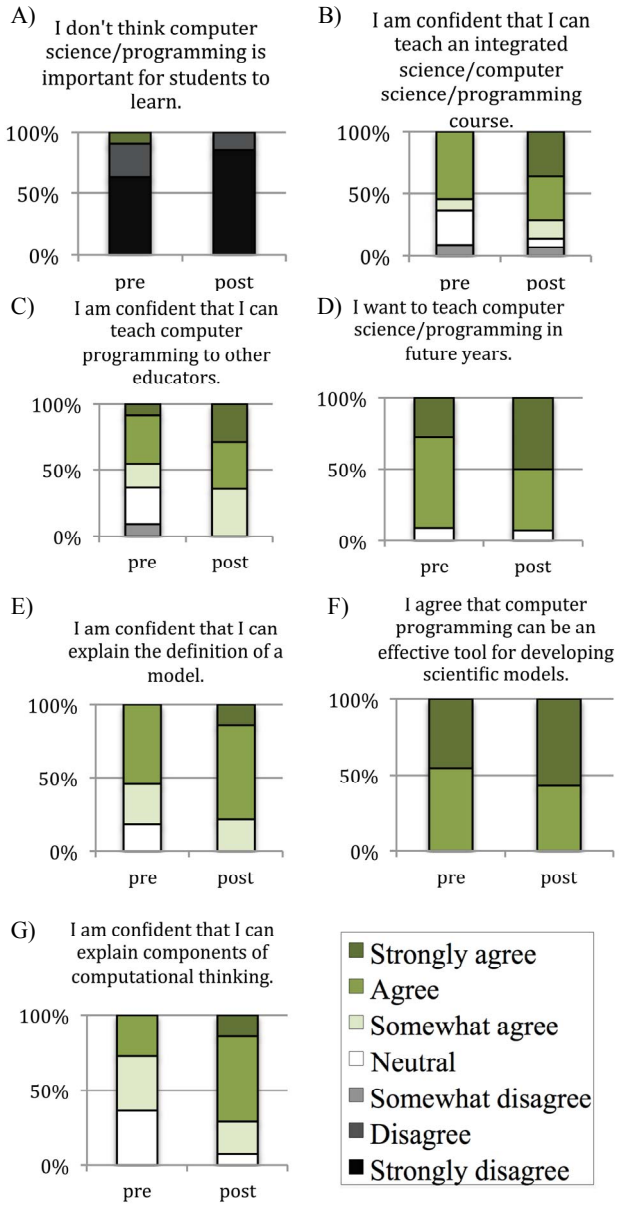

Figure 2: Summary of question responses from pre- and post-surveys.

These data show encouraging outcomes where teachers are reporting increased confidence in and willingness to teach computer programming, agree that computer programming is an effective tool to develop scientific models and report increased confidence in explaining components of CT and the definition of a model, essential aspects of SCM and NGSS SEP 2, 4, & 5. Subsequent iterations of data collection in our program will include more detailed assessment of teachers' understanding and mastery of SCM skills and relevant student learning outcomes.

Acknowledgment

Thank you to J. Sirma, D. Macias, C. Cheung, & Z. (J.) Huang for their contributions to the intervention.

REFERENCES

[1] Wing, J. (2006). Computational thinking. *Communications of the ACM* 49: 33–36.

[2] Henderson PB, Cortina TJ, Hazzan O, and Wing JM. (2007). *Computational thinking*. In Proceedings of the 38th ACM SIGCSE Technical Symposium on Computer Science Education (SIGCSE '07), 195–196. New York, NY: ACM Press.

[3] NGSS Lead States. (2013). *Next Generation Science Standards: For States, By States*. Washington, DC: The National Academy Press.

[4] Luehmann AL. (2007). Identity development as a lens to science teacher preparation. *Sci. Ed*. 91: 822–839.

[5] Schwarz, C. (2009). Developing preservice elementary teachers' knowledge and practices through modeling-centered scientific inquiry. Science Education, 93, 720-744.

[6] Kober, N. (2015). Reaching students: What research says about effective instruction in undergraduate science and engineering. Washington, DC: National Academies Press.