Bringing Expert Computational Practices into High School Science Classrooms

Elham Beheshti, David Weintrop, Kai Orton, Michael Horn, Kemi Jona, Laura Trouille, Uri Wilensky
Northwestern University

Introduction

Science is increasingly becoming a computational endeavor, as computational tools are now pervasive across the scientific disciplines and transforming scientific practices. This shift creates a pressing need to train future scientists, engineers, and mathematicians who understand how to make use of computational tools and approaches to achieve scientific goals. These skills extend beyond programming to include a larger set of skills broadly captured by the concept of computational thinking (Wing, 2006). The importance of bringing computation to science education is particularly evident from the inclusion of "Computational Thinking Skills" in the Next Generation Science Standards (NGSS Lead States, 2013). To better understand how to address the challenge of preparing computationally literate scientific scholars, we conducted a study to identify how professional scientists use computation and computational thinking in their work. This work was done in support of our larger goal of defining what constitutes computational thinking within STEM disciplines in the form of a CT skills taxonomy.

In this paper, we present preliminary findings from interviews with 15 professional STEM practitioners. The two key research questions that this study seeks to answer are: (1) *what computational thinking skills are used by professional STEM practitioners?* and (2) *when and how do STEM professionals use these CT skills in their professional lives?* This approach takes the exercise of defining computational thinking in STEM out of the ivory tower and grounds our taxonomy in CT skills as they are used in the real world. In this way, we are drawing on a social-cultural research tradition that makes "a distinction between the laboratory, where cognition is studied in captivity, and into the everyday world, where human cognition adapts to its natural surroundings" (Hutchins, 1995, p. xiv). This approach has already been undertaken with respect to computational thinking among STEM professionals (Malyn-Smith & Lee, 2012). Our work builds on this project with an explicit emphasis on linking these practices to high school classrooms.

Theoretical Framework

In her 2006 article, Wing introduces the idea of Computational Thinking (CT) and argues that it is a fundamental skill that deserves a position alongside reading, writing and arithmetic as part of the core knowledge one needs to be successful in the 21st century (Wing, 2006). While this idea has been championed by various people under differing names for decades (diSessa, 2000; Guzdial & Soloway, 2003; Papert, 1980), Wing's most recent call for its inclusion in formal education has created a great deal of excitement and efforts towards the cause. The importance of CT skill and its role in education is becoming increasingly recognized (National Research Council, 2011), however, defining exactly what this skill is comprised of remains an open question (Grover & Pea, 2013). The goal of our larger study is to understand and define what CT skills are within the disciplines of Science, Technology, Engineering, and Math (STEM) in an effort to prepare future generations of scientists.

The Growing Role of Computation in Science

Recent advances in high-speed computation and analytical methods have created some of the most powerful tools in understanding phenomena across all spectra of human inquiry. In some STEM fields, such as molecular biology and chemistry, the advent has been recent but rapid. Though still a relatively young and emerging series of disciplines, computational sciences are now ubiquitous in all aspects of STEM professions (ACM/IEEE-CS Joint Task Force on Computing Curricula, 2013). Computational methods have expanded the range of phenomena that can be explored through the use of mathematical and simulation models. Wolfram (2002) went so far as to proclaim the emergence of a new kind of science based on his computational experiments into emergent patterns in nature, arguing such explorations are not possible without computation. STEM fields have seen a renaissance in experimental approaches primarily due to the availability of more powerful computers, accessibility of new analytical methods, and the development of highly detailed computational models in which a diverse array of components and mechanisms can be incorporated (Augustine, 2005). Moreover, effective computational approaches have become essential for dealing with the ever-expanding massive data sets characteristic of complex systems across disciplines.

Wing (2006) proposed that the advances in computing would allow researchers across all disciplines to envision new problem-solving strategies and to test new solutions in both the virtual and real world. Indeed, computational methods have aided in the interpretation of experimental data as well as informing more rational design of new experiments thereby facilitating future scientific, technological and engineering design breakthroughs. These advances have in turn created a growing need to educate students in computational methods and techniques to support the rapidly changing landscape of research across the STEM disciplines (Henderson, Cortina, & Wing, 2007, p. 195). Towards this end, a growing body research has argued for and documented success associated with bringing computational thinking into science classrooms (Hambrusch, et al., 2009; Jona et al., 2014; Sengupta et al., 2013; Wilensky, Brady, & Horn, 2014).

Methods and Data Sources

To better understand the nature of CT-STEM skills in practice, we conducted semi-structured clinical interviews with 15 STEM practitioners who used computation as a central component of their research. The interviewees included 9 doctoral students, one researcher, and 5 faculty members from various STEM disciplines (including physics and astronomy, biology, biochemistry, and earth and planetary sciences). Seven of the 9 doctoral students were fellows in an NSF GK-12 program that links STEM graduate students who use computation in their research with high school teachers to develop classroom-ready activities based on their research that incorporate computational thinking components. These participants were especially valuable as they have firsthand experience translating their computational thinking work practices into high school educational contexts.

The collected data was in the form of video recording of the interviews and also field notes taken by the interviewers. The interview protocol included questions about participants' background and research, computational tools, simulation packages, and programming languages that they use in their work (if any), the role of computers in their research, and different computational challenges they face when solving problems. An additional set of questions included how best to integrate these practices into high school classrooms and what concepts such curricula should include. Each interview session lasted approximately one hour. The interviews were transcribed and analyzed by two researchers. We used a Grounded Theory approach to identify CT skills that arose during the interviews.

Preliminary Results and Discussion

In this section, we present our preliminary findings of exploring interviews, focusing on two, complimentary participants, one theoretical researcher and one experimental researcher. Below we compare the similarities and differences between these two STEM practitioners and identify CT skills as they apply to the participants' professions as STEM practitioners. Don is a PhD student in physics and his research focuses on simulating and characterizing soft-matters such as polymers or colloids physics. He performs computer simulations to investigate the interactions of nano-sized particles that form materials of different structures and properties. In his research, he also develops new simulation algorithms and implements them in C++. Katie is a PhD student in chemistry and her research is on medicinal chemistry, which involves both organic chemistry and biochemistry. She studies enzymes involved in neurodegeneration, and her research involves synthesizing and lab testing inhibitors in a cell-based assay.

Similarities

As both Don and Katie use computation in their research, there are a number of CT practices that both identified as central to their work. These practices involve issues of data management and storage, using computational tools to assist with data analysis and communication of findings, notable for visualizing their data both for their own use and to help convey findings to others. In sum, computation augments domain knowledge for both researchers by facilitating these practices and also enabling explorations that are not possible without computation.

Uses and limitations of computational tools. The use of computational tools and strategies, that are more effective, are central to both researchers. For both Katie and Don, there are a set of standard tools that are common to their field that they rely heavily on. For example, Katie uses GraphPad to perform biostatistics and Don uses LAMMPS (a Molecular Dynamics Simulator) to model the molecules. Despite the centrality of these tools to their specific declines, they both know how the use of computational tools can restrict them in their research. This understanding is demonstrated as Katie talks about some "*outdated software that are a little frustrating"* but then she suggests "*by using more up to date software I would be able to get a whole slew of numbers and then I would be able to make my own graph from it"*. Similarly, Don conveyed the limitations of computational tools used in his work: "*I think it both directs and limits what we study. Because often we'll think of an idea and then we'll think can we do it? How can we do it? And knowing a software that has a feature and I know how to modify it means that we can go ahead. But if we have an idea and then we can't think of how we could actually do it, then it is not something we would do*". In both cases, the scholars identify how central and important computational tools are to their work, but also recognize that these tools are limiting to both what they do and how they can go about doing it.

Visualizing data. Visualizing data is a main part of both Don's and Katie's research and they both use some visualization software specialized for their field or some statistics software to create charts and infographics. The visualizations are mainly used as a way to share findings with peers and research community. For example, Don uses VMD (Visual Molecular Dynamics) software to visualize molecules. To share her findings with the research group, Katie also generates graphs and charts using excel. She also uses ChemDraw, which is a drawing tool for chemists to create scientifically intelligent drawings.

Differences

The central difference between the role of computation in Don and Katie's work stems from the nature of the STEM investigations being conducted. Computation is central to Don's work as many of his findings are purely computational in nature, being new and/or improved algorithms that describe or recreate observed phenomena. Alternatively, for Katie, computation is not the output of her research, but instead enables and facilitates her research. In other words, Don "creates" computational tools while Katie "uses/consumes" computational tools. In his research, Don examines an "unknown" problem, which involves "proving" that some solution works, whereas Katie's research problem is known and her research tries to "find" something that works. While this is the major difference between the two CT-STEM practitioners, we also observed the following differences in CT skills and computational tools used by Don and Katie in their research.

Scale of data. For Katie, "*it's not usually a massive set of data"* and data takes the form of a small set of numbers (of a spectrum from experiments). She mainly handles the data manually and less automation is used in storing and manipulating data. Alternatively, in Don's research, data is a large set of numbers stored in text files. To make meaning of those numbers, he has to do a lot of post-processing: "*So there is a lot of data manipulation there... You get the dump file, which is all the particle coordinates, and then you post process to find out what is interesting about it and what you want to know... then we would write our own scripts that convert raw data with all the particle positions to something more useful... for example it searches over those coordinates and identifies the structures we are interested in, and then adds them up and stores them and then reports like a histogram*". Unlike Katie who has a manageable set of data that is produced by computational tools, Don needs to use further computational tools to interpret the results of his computational experiments.

Testing and debugging. Testing and verifying the new algorithms is an important aspect of Don's work due to the fact that "*the problem is unknown*". For Don, testing takes two steps. First, he tests his algorithm by "*comparing it to some known case,"* and he does this by running the simulation and checking the output numbers for a long run time to make sure that the numbers match at a high degree of precision. Then he compares a new structure of interest on two different codes, his algorithm and a verified code: "*In both cases I try to compare to a known thing or something that I trust more".* Whereas, in Katie's research the problem is known, so there is less need to verify the approach. However, in some parts of her research, before running the experiments she might use computational molecular modeling to "*test to see how well it [the new molecule], in the big picture, could be the molecule we want to make."*

Programming knowledge. Due to the nature of Don's research, his work involves intensive computation and programming. As such, he has to be able to understand and modify programs written by others as well as write new programs or scripts from scratch. In his research, he uses these programming skills to perform a variety of practices such as data generation, data analysis, building new models, and visualizing information. Alternatively, Katie does not have any programming knowledge; she prepares problems for computational solutions and "uses" programs written by others.

Conclusion

In this study, our goal is to bring current STEM educational efforts in line with the increasingly computational nature of STEM research practices. We conducted interviews with STEM practitioners in various fields, which gives us greater insight into the nature of CT skills in practice. This data has been used to inform the development of a more complete and accurate set of CT skills in a form of a taxonomy. The taxonomy, in turn, can help us with our long-term goals which focus on the design of learning environments and classroom activities to teach these CT skills.

Above we began to outline the nature of computational thinking as it happens in authentic scientific research settings. In the preliminary analysis we present in this paper, we highlight similarities and differences in CT-STEM practices across two disparate forms of scientific research. Whereas Don works in a theoretical space and makes computation the output of his research, Katie, uses computation as a means to accomplish her research goals. Despite this fundamental difference, the two scholars share CT practices, which helps us identify what practices are important to make their way into high school CT-STEM learning contexts. It is important to note that these CT skills in STEM practices are part of a larger set of computational practices that can be observed in "data-driven" computational fields across disciplines. From research laboratories, engineering and medical practices and beyond, computational thinking practices and skills are core to what defines STEM in the 21st century.

References

Augustine, N. R. (chair) (2005). Rising above the gathering storm: Energizing and employing America for a brighter economic future. Washington D.C.: National Academies Press.

ACM/IEEE-CS Joint Task Force on Computing Curricula. (2013). Computer Science Curricula 2013. ACM Press and IEEE Computer Society Press. Retrieved from http://dx.doi.org/10.1145/2534860

diSessa, A. A. (2000). *Changing minds: Computers, learning, and literacy*. Cambridge, MA: MIT Press.

Guzdial, M., & Soloway, E. (2003). Computer science is more important than calculus: The challenge of living up to our potential. *SIGCSE BULLETIN*, *35*(2), 5–8.

Grover, S., & Pea, R. (2013). Computational Thinking in K-12: A Review of the State of the Field. *Educational Researcher*, *42*(1), 38–43.

Hambrusch, S., Hoffmann, C., Korb, J. T., Haugan, M., & Hosking, A. L. (2009). A multidisciplinary approach towards computational thinking for science majors. In *ACM SIGCSE Bulletin* (Vol. 41, pp. 183–187).

Henderson, P. B., Cortina, T. J., & Wing, J. M. (2007). Computational thinking. In *ACM SIGCSE Bulletin* (Vol. 39, pp. 195–196). ACM.

Hutchins, E. (1995). *Cognition in the wild*. Cambridge, MA: MIT Press.

Jona, K., Wilensky, U., Trouille, L., Horn, M. S., Orton, K., Weintrop, D., & Beheshti, E. (2014). Embedding Computational Thinking in Science, Technology, Engineering, and Math (CT-STEM). Presented at the Future Directions in Computer Science Education Summit Meeting, Orlando, FL.

Malyn-Smith, J., & Lee, I. (2012). Application of the Occupational Analysis of CT-Enabled STEM Professionals as a Program Assessment Tool. *The Journal of Computational Science Education*, *3*(1), 2–10.

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

National Research Council. (2011). *Report of a Workshop of Pedagogical Aspects of Computational Thinking*. Washington, D.C.: The National Academies Press.

Papert, S. (1980). Mindstorms: Children, computers, and powerful ideas. New York: Basic books.

Sengupta, P., Kinnebrew, J. S., Basu, S., Biswas, G., & Clark, D. (2013). Integrating CT with K-12 science education using agent-based computation: A theoretical framework. *Education and Information Technologies*, 1–30.

Wilensky, U., Brady, C. E., & Horn, M. S. (2014). Fostering Computational Literacy in Science Classrooms. *Commun. ACM*, *57*(8), 24–28. doi:10.1145/2633031

Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, *49*(3), 33–35.

Wolfram, S. (2002). *A New Kind of Science* (1st ed.). Wolfram Media.