Kong, S.C., Andone, D., Biswas, G., Crick, T., Hoppe, H.U., Hsu, T.C., Huang, R.H., Li, K.Y., Looi, C.K., Milrad, M., Sheldon, J., Shih, J.L., Sin, K.F., Tissenbaum, M., & Vahrenhold, J. (Eds.). (2018). *Proceedings of the International Conference on Computational Thinking Education 2018*. Hong Kong: The Education University of Hong Kong.

Developing a Framework for Computational Thinking from a Disciplinary Perspective

Joyce MALYN-SMITH[1], Irene A. LEE[2], Fred MARTIN[3], Shuchi GROVER, Michael A. EVANS[4], Sarita PILLAI[1]
[1]Education Development Center
[2] Massachusetts Institute of Technology
[3] University of Massachusetts Lowell
[4] North Carolina State University
jmsmith@edc.org, ialee@mit.edu, fred_martin@uml.edu,
shuchig@cs.stanford.edu, michael.a.evans@ncsu.edu, spillai@edc.org

ABSTRACT

This paper describes progress towards the development of a Framework for Computational Thinking (CT) from a Disciplinary Perspective. The work aimed at discovering how CT can be encouraged, taught and practiced within disciplines throughout primary and secondary education. It identifies an initial set of "elements" describing CT practices that bridge learning and working in highly sophisticated STEM environments and shares examples of these practices used by STEM professionals at work and developed by students in schools. It is hoped that this paper will provoke dialogue among educators advocating for CT as a core skill for all and will contribute to breakthroughs in thinking about how CT should be learned and assessed in and out of school.

KEYWORDS

Computational thinking, K-12 education, workforce development, human-technology frontier.

1. INTRODUCTION

The proliferation of new technologies has changed the way we live, learn, and work. Although the future of work is unclear, experts envision a new machine age, where technologies (sensors, communication, computation, and intelligence) are embedded around, on, and in us; where humans will shape technology and technology will shape human interaction; and where technologies and humans will collaborate to discover and innovate. In short—the Human-Technology Frontier.

Without question, the global workforce will need a new set of skills and competencies to succeed in the future work environments on this frontier—that feels closer with each new technological advance. A recent report by EDC's STELAR Center (Malyn-Smith et al., 2017) identified computational thinking as one of the essential skills needed by future workers for success in work at the Human-Technology Frontier. As our society works to understand and identify strategies to overcome these complex and interrelated challenges, important questions include: What can we do to prepare today's students to succeed in work at the Human-Technology Frontier? and What steps can we take to make this happen? If we are to believe that the Human-Technology Frontier is upon us, we need to reconsider how computational thinking is taught in order to advantage our students, not only in developing CT skills, but also in developing the CT practices used in STEM workplaces (EDC, 2011).

2. BACKGROUND

Since noted computer scientist Jeannette Wing (2006) proposed CT as a new "core skill" various groups have tried to define CT for education and training purposes (e.g. Grover & Pea, 2013, 2018). CT (focusing on problem-solving, algorithms, data representation, modeling and simulation and connections to other fields) is a prominent strand of the K-12 Standards for Computer Science developed by the Computer Science Teachers Association (CSTA, 2011). Individual states (including Massachusetts and New Jersey, USA) have instituted computer science (CS) and digital literacy standards that use the term CT. Next Generation Science Standards (NGSS Lead States, 2013) include computational thinking in one of their eight scientific practice standards. National Science Foundation (NSF) funded projects are conducting research on several different approaches to CT. Data practices, modeling and simulation practices, computational problem solving practices and systems thinking practices are proposed by Weintrop et al. (2016). Lee et al. (2011) propose that youth develop CT skills as they use, modify and create with digital tools and technologies. While these initiatives signal a broad based, grassroots interest in computational thinking, their simultaneous development and independent implementation leaves us without consensus on a precise definition of CT. (Barr & Stephenson, 2011; Voogt, Fisser, Good, Mishra, & Yadav, 2015; Weintrop et al., 2016). Most agree, however, that Computational Thinking is formulating problems and their solutions in a way that a machine (computer) can be used to represent the problem and carry out its solution.

What has emerged from these varied research and practice efforts aimed at CT is a debate over how CT is best taught and learned. Many computer science educators believe that CT is best taught through programming where students' development of CT can be ensured and uniquely observed. Others believe that to best prepare today's youth for tomorrow's world, CT should be taught/learned in the service of disciplines. While many of the efforts described above define CT by dissecting it into its component parts, little has focused on what results from integrating CT and disciplinary learning. To guide teaching and learning of CT within the disciplines, a new kind of computational thinking framework was needed – one which captured and clarified what students were able to do using CT – and unable to do without CT.

3. DEVELOPING A FRAMEWORK

A group consisting of principal investigators, researchers, and educators from National Science Foundation funded ITEST (Innovative Technology Experiences for Students and Teachers) and STEM+C (STEM+Computing) projects convened in August and November 2017 to explore the development of an Interdisciplinary Framework for Integrating CT in K-12 Education. Their goal was to draft a framework defining computational thinking from a disciplinary perspective. The 54 workshop participants provided a good balance of researchers and practitioners, who represented grade spans Kindergarten-2nd grade, 3rd-5th grade, 6th-8th grade, and 9th-12th grade, as well as disciplines including science, mathematics, engineering, social science, computer science and the humanities. In total there were 31 researchers, 18 teachers / practitioners, 3 participant observers, and 2 staff members. (13 of the participants were from colleges/universities, 15 from schools, 15 from non-profits, 1 from business, 3 from foundations including the NSF). The primary goals were to develop a framework for computational thinking from a disciplinary perspective that built on the work of the foremost researchers and practitioners focused on helping youth develop CT skills. Progress towards the goals was guided by some of the foremost CT thought leaders in the U.S. including Irene Lee of Massachusetts Institute of Technology, Shuchi Grover, Fred Martin of University of Massachusetts Lowell and CSTA, and Michael Evans of North Carolina State University.

As a first step, participants were asked to submit examples of their work to share with other participants prior to the workshops. Educators/practitioners shared curriculum and activities that illustrated CT in action in their classrooms. Researchers shared their lessons learned through research on various aspects of CT skill development and integration. Together the group explored these examples and found that a number of common "elements" emerged. During the workshops, participants were asked to provide additional examples of CT integration by grade level and discipline. These examples were subsequently reviewed and discussed within the emerging framework of common elements.

Thought about the goal of developing a framework for CT in the service of disciplines crystallized around the larger goal of education – that of preparing youth for success for living, learning and working *after* compulsory education. Thus, focusing on building a bridge between the CT skills developed in school and the professional practices involving CT, particularly those in scientific workplaces became paramount.

A traditional way CT is integrated is shown at the bottom of Figure 1 illustrated with the Massachusetts digital learning and computer science (DLCS) standards component areas of abstraction, algorithms, programming and software development, data collection and analysis, and modeling and simulation. Typically, individual CT components are taught then linked in pairs and clusters leading up to potentially more powerful CT activities at with older age groups.

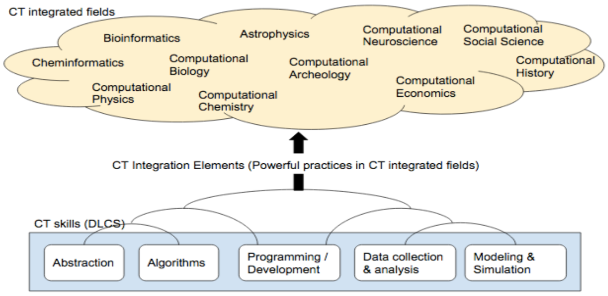

Figure 1. Bridging between traditional teaching of CT and CT as used in CT integrated fields.

Stronger connections between these CT components and the powerful practices used by professionals in CT-integrated scientific fields (e.g. computational biology, bioinformatics, cheminformatics, computational economics and others) were sought. The aim in making these connections was to ensure that the CT integrated in K-12 concept areas provided a strong foundation for the computational thinking used by practicing scientists and would bridge the skills transition from school to work.

4. CT from a Disciplinary Perspective – examples from STEM workplaces

To further explore the elements that might form a framework for CT from a disciplinary perspective, examples of CT commonly used by practicing scientists specifically, examples of what can be accomplished using CT that would be difficult, if not impossible, without CT were gathered. From these examples of CT used by practicing scientists in CT integrated fields, the elements emerged and were tested as organizers for other examples of CT. The initial examples considered follow.

4.1. Ensemble modeling

Scientist use multiple models are used to predict the behavior of complex systems. For example, weather forecasting now uses ensembles of models to understand weather patterns (Gneiting & Raftery, 2005; Krishnamurthy et al., 2000). Each model in an ensemble simulates the global weather system taking different sets of parameters or initial conditions into account. Instead of making a single forecast of the most likely weather, a set (or ensemble) of forecasts is produced. This set of forecasts aims to give an indication of the range of possible future states of the atmosphere.

4.2. Computational chemistry

Scientists innovate with computational representations - For example, the SMILES (simplified molecular-input line-entry system) notation is a representation for describing the structure of chemical compounds using short ASCII strings (O'Boyle, 2012). This revolutionized computational chemistry and drug design by enabling computers to read and operate on chemical sequences (including searching and database indexing).

4.3. Bioinformatics

CT is used in bio-informatics workplaces. In Next Generation Sequencing Data Analysis, dozens of whole genomes can be sequenced in rather short time, producing huge amounts of data (McKenna et al., 2010; DePristo, et al., 2011). Complex bioinformatics analyses are required to turn these data into scientific findings. To run these analyses quickly, automated workflows on high performance computers are state of the art. Scientists design processes to achieve high throughput processing of genomic data.

4.4. Environmental science

Environmental scientists use crowd-sourced data in water management (Fienen & Lowry, 2012; Stepenuck & Green, 2015; McKinley et al., 2015). When considering water management strategies for a region, data for various communities with different water usage and needs (for example, for growing different crops or industrial uses) is necessary to understand the larger picture of water usage and needs, as well as the local variations.

4.5. Machine learning

To a larger and larger extent, scientists are using machine learning to make predictions. In supervised machine learning, scientists build models by running algorithms on "training sets" of inputs matched with correct responses (Srivastava et al., 2014; Lecun, Bengio, & Hinton, 2015). These models can then be used to offer predictions (or responses) when given new inputs. Changes in the training set data can have implications on the machine learning model built and can introduce biases if the training data is not representative of the target.

5. The Elements of CT integration from a Disciplinary Perspective

The examples from advisors and researchers along with lessons and activities provided by educators were examined. Evidence was found that K-12 subject area teachers were integrating CT in ways that were consistent with its use in CT-integrated fields. The following five Elements of CT Integration from a Disciplinary Perspective that emerged from the reviews and discussions were:

1. Understand (complex) systems.
2. Innovate with computational representations.
3. Design solutions that leverage computational power/resources.
4. Engage in collective sense making around data.
5. Understand potential consequences of actions.

5.1. Understand complex systems

Modeling how interactions of many individuals or components in a system lead to aggregate level emergent patterns is difficult to do without CT. Complex systems in particular are not amenable to traditional mathematical analysis. Simulating a system's change over time and real-time feedback in the form of simulations help scientists visualize complex systems dynamics. These systems are often hard to predict due to having a multitude of interrelated factors and levels. In K-12 education, computer modeling and simulation of these systems offers a way to see how the systems behave under different circumstances, with different inputs.

5.2. Innovating with computational representations

The design and development of innovations is made possible through CT. New ideas, conceptualizations, representations, and processes can be thought of and developed as computations. For example, thinking of the brain as a network and creating neural networks as artificial brains has led to advances in artificial intelligence and cognitive science. In K-12, students can be introduced to computational representations by learning about how colors are represented on computers as RGB values.

5.3. Design solutions that leverage computational power and resources

Scientists working with large data sets or on computationally intensive calculations design solutions that leverage the efficient use of resources and computational power to optimize their time. In some cases, distal collaborators can pool and share computational resources and in other cases co-located collaborators can access distributed resources to achieve their goal. Some speedups are achieved by decomposing datasets and/or processes to run in parallel. In K-12 settings, educators can challenge students to think about how they would solve a problem differently if the input set was of large scale. For example, rather than developing processes to assemble 10 finished copies of an item, how would students go about assembling 10,000 copies?

5.4. Engage in collective sense making around data

Data sets can be amassed through crowd-sourcing or collection by multiple individuals or sensors. These data can be analyzed to uncover patterns. Visualization of multidimensional data enables students to see patterns that might not otherwise be apparent. When possible in the K-12 education setting, teachers can ask small groups of students to run simulations on a subset of the inputs, then share their output data and analyses. Gathering and analyzing the combined data illustrates how each part of the data contributes to the understanding of the whole.

5.5. Understand potential consequences of actions

Scientists envision the future through simulation and use machine learning to make predictions. Using parameter sweeping, the space of all possible combinations of inputs can be tested to see the variety and probability of outcomes. In K-12, students can learn how cause and effect relationships can be used to predict outcome. Students can also begin to understand the space of inputs created by parameterizing models.

Notably, these elements of CT integration go *beyond* the mechanics of learning to program a computer. They form a bridge between CT as it has traditionally integrated in K-12 classrooms (through the introduction of computer programming activities) and professional practices.

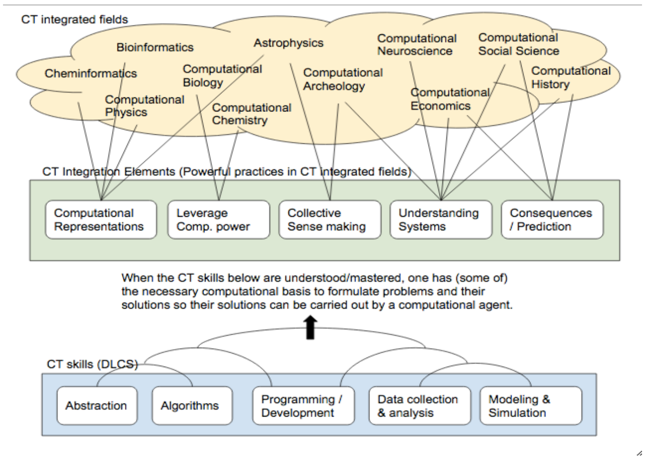

Figure 2. CT integration elements as a bridge between traditional CT integration in K-12 education and CT as powerful practices used in CT integrated fields.

Figure 2 illustrates how the thinking progressed from the idea of direct teaching of CT skills through programming - to a realization that to help students develop CT skills through STEM disciplinary learning, their education needs to include a stronger focus on computational tools, techniques, and processes used in the CT integrated fields.

6. CT from a Disciplinary Perspective – examples from K-12 classroom teachers

Through the examination of lessons provided by K-12 educators, it was determined that a subset of the disciplinary teachers were already integrating CT within K-12 that aligned with the elements presented above. Several lessons and activities teachers provided from their curricula illustrate how these elements can be introduced in K-12 to help students develop CT skills aligned with professional practices.

6.1. Middle school science

In middle school ecosystems lessons (Lee, 2011; Project GUTS, 2014) using the StarLogo Nova modeling and simulation environment, middle school students in science classrooms used, modified and created computer models and ran simulation to understand complex systems; multiple models were produced and compared; students engaged in collective sense making around data (by crowdsourcing data generated from multiple runs of each of the models); and students learned about potential consequences of actions (such as the impact of removing a top predator).

6.2. Elementary school mathematics

In a 5th grade mathematics classroom, students were asked to generate a language to describe a minimal set of actions to be performed by robots tasked to build a tower. Within this activity students were innovating with computational representations, and designing solutions that leverage how computers process data (in this case, instructions).

6.3. High school engineering

In a high school engineering classroom, a teacher used a multi-step physical construction task to illustrate domain vs. task decomposition as method of parallel processing in high performance computing. Students designed processes to make many copies of a Lego figure that leveraged "processing" resources (other students) then optimized the design based on collective sense making from data on time to complete the task.

6.4. Middle school mathematics

In a middle school mathematics classroom, students using the iSENSE data-sharing platform were able to collect and add locally generated data to a large student-generated data set. They could then analyze their data and compare it to data provided from other classrooms (Willis et al., 2015).

6.5. Across subject areas

There is a large window of opportunity for K-12 students to learn about consequences of actions, in areas ranging from cause and effect in programming to decision-making and prediction in machine learning.

7. CHALLENGES

While the path towards CT integration from a disciplinary perspective is growing clearer, many challenges remain. First, we acknowledge that the majority of K-12 teachers are still struggling with the integration of CT in terms of teaching the basics of computer programming. Introducing the elements of CT integration can be viewed as a conflicting definition instead of a further elaboration on a trajectory of CT from K-12 to professional practice.

Another challenge is the rate at which fields are innovating with CT. The examples of CT integrated fields presented in this paper are only a few of the many fields that have been greatly impacted by CT. Many additional fields are incorporating computational tools, techniques, and practices. Across fields, innovations and discoveries made possible by the integration of computational tools, techniques, and practices are increasing.

The rapid rise of machine learning raises yet another challenge. Across disciplines, the need for analysis of computational systems, especially those used to make predictions that greatly impact human life, is paramount. The inclusion of the CT integration element "*Understanding potential consequences of actions"* addresses this important need.

8. CONCLUSION

The authors believe that learning CT needs to extend beyond learning to program. It must include engagement in computational practices used in the sciences that harness the power of computers to enhance scientific discovery. The CT Integration Elements presented here provide a framework for foundational learning of CT within disciplines beginning in elementary school and extending through high school and beyond. Examples provided by K-12 teachers shed light on ways K-12 educators have integrated powerful practices from professional CT integrated fields. It is hoped that the framework can aid teachers in the development of CT lessons, and ensure that the CT that teachers promote has links to the CT used in scientific workplaces. Still, this Framework is a work-in-progress. It is hoped that it will evolve as researchers continue to examine—and K-12 educators increasingly engage in—CT integration in the classroom.

9. ACKNOWLEDGMENTS

We gratefully acknowledge the support of the "Workshop to Develop an Interdisciplinary Framework for Integrating Computational Thinking in K-12 Science, Mathematics, Technology, and Engineering Education" project from the National Science Foundation (DRL award# 1647018). We'd also like to thank the many participants in the workshop series who generously shared their curricula, research, and insights.

10. REFERENCES

Barr, V., & Stephenson, C. (2011). Bringing computational thinking to K–12: What is involved and what is the role of the computer science education community? *ACM Inroads*, 2(1), 48–54.

Computer Science Teachers Association (2011). K-12 computer science standards. http://csta.acm.org/Curriculum/sub/k12standards.html

DePristo, M. A., Banks, E., Poplin, R. E., Garimella, K. V., Maguire, J. R., Hartl, C., Philippakis, A. A., del Angel, G., Rivas, M. A., Hanna, M., McKenna, A., Fennell, T.J., Kernytsky, A.M., Sivachenko, A.Y., Cibulskis, K., Gabriel, S.B., Altshuler, D., & Daly, M. J. (2011). A framework for variation discovery and genotyping using next-generation DNA sequencing data. *Nature Genetics*, *43*(5), 491–498.

EDC (2011). *A Profile of a Computational Thinking Enabled STEM Professional in America's Workplaces – Research Scientists / Engineers*. (revised 2013). Waltham, MA: EDC.

Fienen, M.N., & Lowry, C.S. (2012) Social Water—A crowdsourcing tool for environmental data acquisition, *Computers & Geosciences*, 49(1), 164-169.

Gneiting, T., & Raftery, A. E. (2005, October). Weather Forecasting with Ensemble Methods. *Science*, 310(5746), 248-249.

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

Grover, S. & Pea, R. (2018). Computational Thinking: A competency whose time has come. In *Computer Science Education: Perspectives on teaching and learning*, Sentance, S., Carsten, S., & Barendsen, E. (Eds). Bloomsbury.

Krishnamurti, T. N., Kishtawal, C. M., Zhang, Z., Larow, T., Bachiochi, D., Williford, E., Gadgil, S., & Surendran, S. (2000). Multimodel Ensemble Forecasts for Weather and Seasonal Climate. *Journal Of Climate*, (13). 2000 American Meteorological Society.

Lecun, Y., Bengio, Y., & Hinton, G. (2015, May). Deep learning. *Nature*, 521, 436–444.

Lee, I., Martin, F. Apone, K. (2014). Integrating Computational Thinking Across the K-8 Curriculum. *ACM Inroads*, 5(4): 64-71.

Lee, I., Martin, F., Denner, J., Coulter, B., Allan, W., Erickson, J., Mayln-Smith, J., and Werner, L. (2011). Computational thinking for youth in practice, *ACM Inroads*, Vol. 2 No.1.

Malyn-Smith, J., Blustein, D., Pillai, S., Parker, C. E., Gutowski, E., & Diamonti, A. J. (2017). *Building the foundational skills needed for success in work at the human-technology frontier.* Waltham, MA: EDC.

McKenna, A., Hanna, M., Banks, E., Sivachenko, A., Cibulskis, K., Kernytsky, A., Garimella, K., Altshuler, D., Gabriel, S., Daly, M., & DePristo, M. A. (2010). The Genome Analysis Toolkit: A MapReduce framework for analyzing next-generation DNA sequencing data. *Genome Research*, *20*(9), 1297–1303.

McKinley, D. C., A. J. Miller-Rushing, H. L. Ballard, R. Bonney, H. Brown, D. M. Evans, R. A. French, J. K. Parrish, T. B. Phillips, S. F. Ryan, L. A. Shanley, J. L. Shirk, K. F. Stepenuck, J. F. Weltzin, A. Wiggins, O. D. Boyle, R. D. Briggs, S. F. Chapin III, D. A. Hewitt, P. W. Preuss, and M. A. Soukup. (2015). *Investing in citizen science can improve natural resource management and environmental protection.* USGS Publications Warehouse. http://pubs.er.usgs.gov/publication/70159470

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

O'Boyle, N. M. (2012). Towards a Universal SMILES representation - A standard method to generate canonical SMILES based on the InChI. *Journal of Cheminformatics*, 4(22).

Project GUTS CS in Science curriculum (2014). Ecosystems as Complex Systems. Downloaded at http://www.teacherswithguts.org .

Stepenuck, K.F., and Green. L. (2015). Individual and community level impacts of volunteer environmental monitoring: a synthesis of peer-reviewed literature. *Ecology and Society*, 20(3):19.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014, June). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929−1958.

Voogt, J., Fisser, P., Good, J., Mishra, P., & Yadav, A. (2015). Computational thinking in compulsory education: Towards an agenda for research and practice. *Education and Information Technologies*, 20(4), 715–728. Retrieved from http://link.springer.com/article/10.1007/s10639-015-9412-6

Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2016). Defining Computational Thinking for Mathematics and Science Classrooms. *Journal of Science Education and Technology*, 25(1), 127–147.

Willis, M. B., Hay, S., Martin, F. G., Scribner-MacLean, M., & Rudnicki, I. (2015). Probability with Collaborative Data Visualization Software. *Mathematics Teacher*, 109(3), 194–199.

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