Examining how problem design relates to computational thinking practices

Theodore E. Bott,[1] Tyler Stump,[1] Marcos D. Caballero,[1,2,3] Daryl R. McPadden,[1] and Paul W. Irving[1]

[1]*Department of Physics & Astronomy, Michigan State University, 567 Wilson Rd., East Lansing, MI, 48824*
[2]*CREATE for STEM Institute, Michigan State University, 567 Wilson Rd., East Lansing, MI, 48824*
[3]*Department of Physics & Center for Computing in Science Education, University of Oslo, N-0316 Oslo, Norway*

With the growing ubiquity of computation in STEM fields, understanding how to teach computational thinking (CT) practices has become an active research area in the last two decades, with particular emphasis on developing CT frameworks. In this paper, we apply one of these CT frameworks and compare the results with a task analysis to examine how CT practices relate to specific design features of an in-class problem. We have analyzed video data from two separate groups working on one computational class period, which utilizes a minimally working program to model magnetic field vectors. While still in the initial stages of the study, our preliminary results indicate that what is left out of the minimally working program will impact the CT practices students use, particularly around building computational models. Ultimately, we hope this work will help instructors to design activities that can target & build specific CT practices.

2022 PERC Proceedings edited by Frank, Jones, and Ryan; Peer-reviewed, doi.org/10.1119/perc.2022.pr.Bott

I. INTRODUCTION

Computation is widely recognized as a pivotal resource for academic and industrial purposes [1]. The Next Generation Science Standards (NGSS) emphasize using mathematics and computational thinking as one of the eight major scientific practices that all students should encounter in their K-12 education [2]. With the growing utility of computation across STEM and beyond, research in the area of computation education is also expanding [3–5], including research in computational thinking (CT) [6]. There are multiple definitions for CT, but it has recently been described as "the thought processes involved in formulating a problem and expressing its solution(s) in such a way that a computer – human or machine – can effectively carry out" [7].

The development of expertise with CT practices is often a learning goal for physics with integrated computation classrooms [8, 9]. However, assessing CT practices and their development has proven to be a challenging task for teachers. Part of the struggle to develop CT practice-based assessments is based in the question of what can and should be tested in particular learning environments. The development of a plethora of CT frameworks has highlighted that CT can manifest differently depending on contextual differences, such as discipline (math vs. biology vs. physics class), education level (K-12 vs. collegiate level), and audience (experienced programmers vs. inexperienced programmers) [10]. This complexity of context has resulted in a variety of descriptions of CT practices [11–15]. However, these descriptions often lack concrete instantiations or fail to discuss how their form of CT manifests in the classroom and how to assess it.

The paucity of research-based assessments in CT means computational activities in physics classrooms are designed without clear assessment goals as would be suggested by best practices [16, 17]. As a consequence, there is a lack of understanding about how activity design connects to student engagement with CT practices in the classroom. This is true for the computation-integrated context in which this study occurred. However, if we understand the connection between problem design and CT, then we can use that as a foundation for how to construct assessments around specific CT practices for a particular context. We expect transferable CT practices to be observable in the stages of students' solution processes when working on a computationally integrated assignment. Under this assumption, we ask the research question: How do the stages involved in completing a computational model relate to the CT practices that students engage in across the activity?

II. BACKGROUND & METHODOLOGY

We selected CT framework that best fit our context and used a task analysis to relate CT practices with specific parts of an activity. For the CT framework, we used the framework developed by Weller et al. for high school and introductory physics learning environments. The Weller et al. framework was grounded and developed in high school classrooms that used group-based learning combined with minimally working programs (MWPs), which provide a structured code that compiles without errors but lacks the correct model of the physical phenomena [18, 19]. The Weller et al. framework contains a total of fourteen CT practices, which are categorized according to the stages of a model development. To provide an example, one CT practice in the framework is Utilizing Generalization, which is defined as "Importing previous approaches, algorithms, or specific code into a model" and is contained within the "Building Computational Models" category [10]. This practice would be flagged when student discussion revolved around pulling code from a previous problem, copy/pasting existing code with small modifications, or referencing the same computational strategies utilized elsewhere for the current problem. The Weller et al. framework highlights key words, actions and phrases said by students that are used as "indicators" for an occurrence of a particular practice [10].

We selected the Weller et al. framework because it had these operationalized markers and because our context similarly used group-based learning and MWPs, and covered similar introductory physics content. Our study took place in a course called Electricity and Magnetism Projects and Practices in Physics (EMP-Cubed), which is a second semester, calculus-based, introductory physics classroom at a large R1 university that integrates computation throughout the curriculum, including the in-class activities, online notes, homework, and exams [20–25]. The EMP-Cubed course was fundamental as a basis setting for the research that lead to the design of the Weller et al. framework. As a result, many of the features of EMP-Cubed were used when teachers integrated computation into their classrooms especially the emphasis on group work and MWP's. We believe these similarities allow for the use of the Weller et al. framework in the EMP-Cubed context with the acknowledgement that some practices might exhibit in different ways in the university context. EMP-Cubed is designed as a flipped, project-based learning classroom [23, 25–27]. Students meet two times a week for two hours, with each meeting focused on completing a single, complex problem in small groups of 4-5 students [24]. About one third of the in class problems are MWP computation problems. During class periods, students work with their groups to add or edit the given code rather than writing a program from scratch [21, 22, 24, 28].

For this study, we have used in class video data of two student groups from one class period of EMP-Cubed, nine weeks into the semester. Groups are determined randomly at first, but after some time in the semester, students are rotated into new groups based on factors such as computational experience, talkativeness, facilitation skills, and more. This is the 6th computation problem in the semester, so students may have differing levels of computational experience, but have at least seen similar excerpts of code previously within the course. During this class period, students are asked to use

Expert Task Analysis	
Step number	**Description of step**
1. *Define charge of Hawkions*	Set as Q=-1.602E-19 under ##Parameters and Initial Conditions.
2. *Make an observation point*	Move p = sphere(pos=vector(-1,-1,0), radius-0.1, color=color.cyan) line from the comments to the ##Objects section so that the observation point p is defined with the other objects.
3. *Define/declare an arrow*	Either define one by yourself or copy over Barrow=arrow(color=color.red) line from the comments to before the while loop so that the arrow is defined outside of the calculation loop.
4. *Set the position of the arrow to be at the observation point*	Move Barrow.pos=p.pos line from the comments to before the while loop (under the line of code from Step 2) so the position of the Barrow is set to the position of the observation point.
5. *Calculate the separation vector*	Inside the while loop (before the charge.pos update line, after the rate(100) line), calculate the separation vector by typing r_sep=p.pos - charge.pos
6. *Calculate the magnetic field*	After defining r_sep (still before the charge.pos line), calculate the magnetic field using B=1E-7*Q*cross(velocity,r_sep)/mag(r_sep)**3
7. *Set the axis of the arrow to be equal to the B-field*	(Scale will be needed for visibility). Do this by moving the Barrow.axis=vector(0,0,0) line from the comments to below the B calculation line and change it to Barrow.axis = B*scale.
8. *Define scale so the arrow is visible*	Estimate the order of magnitude from the B-equation, use guess & check, or use a print statement to find the scale. Set the scale to be scale=1E26 since B is extremely small.
9. *Double check direction of arrow*	Utilize the "right hand rule."
10. *Change parameters*	Parameters include sign of charge, magnitude of charge, velocity direction. Make sure arrow responds accordingly.
11. *Add more observation points by copy/pasting lines of code*	Create new observation point and a new arrow (renamed accordingly). Set the new arrow location to the new observation point. Calculate the new separation vector and the new B-field. Make the new B-field the axis of the new arrow.
12. *Repeat step 11*	As needed, for the desired observation locations.

TABLE I. An expert's solution path through the design of computational model. The initial minimally-working program can be found in Ref [30] , and the expert's finished model can be found in Ref [31]:

GlowScript (VPython) to construct a model of the magnetic field created by a single moving point charge [29]. For this particular problem, the MWP shows the charge moving along the x-axis; however, there are no magnetic field arrows [30]. The given code provides the object definitions, the velocity of the charge, and a while loop that updates the position of the charge. Students must then add an arrow to represent the magnetic field at an observation point as the charge passes by. The final working code can be found at Ref. [31].

To relate the CT practices to the problem, we first outlined all of the essential tasks needed to complete the MWP, shown in Table I. This list of tasks was produced by one of the designers of the EMP-Cubed class, in which the specific steps and chronology of the task are outlined. It is important to note that this is the minimum number of steps, and that student groups may include extra steps or complete them out of order. When originally designed, this problem intended to have students extensively discuss both the physics of the problem (creating observation points, calculating separation vectors, and implementing physical equations), as well as the computational elements of the model, (structure/order of lines of code, cosmetic changes such as color and size, and syntax of mathematical operations and functions). However, it is important to note that while EMP-Cubed does have its own unique computational goals (such as familiarizing students with computation and helping students gain experience with constructing computational models), the magnetic field activity was designed prior to the Weller et al. framework and was not designed with any specific CT practice-based learning goals in mind.

Using the Weller et al. framework two researchers tagged the videos separately for CT practices and compared results to establish inter-rater reliability [32]. Both researchers documented which indicators were present for every particular coded segment, as well as the specific start times (referring to the moment(s) students began engaging with the practice) and stop times (the moment(s) students stopped engaging with the practice) for each instance of CT engagement. The minor differences between the start and stop times determined by the researchers were compared, but the differences in these times was never comparable to the length of the overall engagement, so they were deemed negligible and were thus ignored. Researchers disagreed on 20% of tagged segments initially. Through collaborative discussion, the majority of the disagreed-upon tags were simply due to overlooked audio cues, rather than disputes about the interpretation of markers from the framework. We reached a consensus for all CT engagement tags for both videos.

Alongside the CT analysis, the videos were analyzed for the tasks the groups engaged in while working on the magnetic field activity. Initially, the first 30 minutes of the two videos were analyzed by two researchers using the expert-

Computational Thinking Engagement													
	Extract Insight		Build Computational Models					Data Practices					**Task Count**
	Dec	H&F	TPIC	ACL	AB	UG	ACM	CDRF	IGD	MD	AD	Deb	
1. Define charge	0	0	1	0	2	0	0	0	0	0	0	0	**3**
2. Make obs point	0	2	1	1	1	1	0	0	0	0	0	0	**6**
4. Set arrow position	0	1	1	1	0	0	0	0	0	0	0	0	**3**
5. Calc separation vector	0	1	3	3	2	2	0	1	1	0	0	1	**14**
6. Calc magnetic field	1	1	3	0	0	2	0	0	1	0	0	2	**10**
7. Set the arrow axis	0	1	1	12	1	2	0	1	4	0	1	3	**26**
8. Define arrow scale	0	0	0	0	0	1	1	0	1	0	0	2	**5**
9. Check arrow direction	0	0	0	0	0	0	0	0	0	0	1	0	**1**
11. Add more obs points	0	1	0	0	0	2	0	0	3	0	1	0	**7**
A. Planning	0	3	1	1	0	0	0	0	0	0	0	0	**5**
B. Coding discussions	0	0	1	0	1	2	0	0	0	0	0	1	**5**
C. Off-topic discussions	0	0	1	0	0	0	0	0	2	0	1	1	**5**
Total CT Counts	**1**	**10**	**13**	**18**	**7**	**12**	**1**	**2**	**12**	**0**	**4**	**10**	**90**

TABLE II. Total counts of the computational thinking engagement for individual practices at different stages of the task. The CT practice labels are as follows: Dec=Decomposing, H&F=Highlighting & Foregrounding, TPIC=Translating Physics Into Code, ACL=Applying Conditional Logic, AB=Algorithm Building, UG=Utilizing Generalization, ACM=Adding Complexity to a Model, CDRF=Choosing Data Representation Form, IGD=Intentionally Generating Data, MD=Manipulating Data, AD=Analyzing Data, Deb=Debugging. Stages 3, 10, and 12 had no CT engagement.

level task analysis shown in Table I. For example, in the "Defining a charge" step, students must identify the charge of a hawkion (fictional particle) and implement it in the code. Key words and phrases were used to identify the focus of a group, and an inter-rater reliability check was performed on these tags [32]. Initially, we disagreed on less than 10% of tagged segments, which was resolved fully after discussion.

For clarity, instances that were identified to have similarities with any of the expert's steps were labeled as that step. For example, students discussing a plan to create an observation point or actively coding the observation point were both classified under the expert step "Make an observation point" within the task analysis. Student actions that did not align with any specific facet of the task analysis were also categorized. Examples of these categories include non-task-oriented discussions, coding discussions, and planning. After both the CT practice analysis and the task analysis, the instances of each were compared.

III. RESULTS & DISCUSSION

We summarize the results of our analysis in Table II. The CT practices in the Weller et al. framework appear as columns and provide the counts for observed coded segments in each CT practice. Notably, the highest total number of counts appears for the category Building Computational Models, arguably a central practice of this activity. The task analysis (Table I) provided the flags resulting in the counts along each row of Table II. Quite importantly, the highest counts across these tasks focus on the work needed to produce the magnetic field vector. Below, we unpack these results further.

We find that the Weller et al. framework sufficiently captures the macro scale alignment between our design intentions and the observed CT practices. The design of these activities sought to not only facilitate physics learning around magnetic fields, but also to help students develop experience with computational modeling of physical phenomena, including syntactical language, design features, and script structure. Reading along the bottom of Table II, the most prevalent practices evident in the CT engagement analysis were those contained within the "Building Computational Models" category. Most counts occurred when students were "Translating the Physics into Code", "Applying Conditional Logic", and Utilizing Generalization." Practices within both the "Extracting Computational Insight" and "Data Practices" categories were relatively uncommon by comparison. This particular magnetism problem occurred in the ninth week of class, when students may have progressed to a point where the "Extracting Computational Insight" steps were more implicitly carried out as an experienced group, and thus were not explicitly codeable in the analysis. Moreover, the activity was not originally designed for rampant engagement with data visualization or analysis, but was rather primarily concerned with getting students to apply fundamental magnetism laws and to produce an accurate visualization of the model. Thus, it is not unexpected to find a lack of CT engagement from these categories. The Weller et al. framework positions the CT practice of debugging as something that occurs at any point in working through a MWP activity and the analysis did find it ubiquitous throughout the entirety of the activity, rather than specific to a particular category. Finally, our analysis of student video using the task analysis in Table I demonstrates where students in these groups focused their efforts. Looking down the last column of Table II, many of the task analysis codes applied to the work students did focus on determining, calculating, and representing a single magnetic field vector as an arrow, which is consistent with our experiences teaching EMP-Cubed.

Moving to a narrower focus, we consider the patterns in Table II, which can show stronger connections between the tasks students were completing and the CT practices being used. We note a large spike (12 counts) of "Applying Conditional Logic" engagement when students were focused "Setting the axis of the arrow to the magnetic field." In computational problems prior to the magnetism activity, students primarily modeled electric field lines in static cases. Computationally, this typically does involve the creation of multiple arrows representing the electric field, but the arrows are never changing in size or direction, and are instead modeling some electric field at one instance in time. With the problem analyzed, the students must realize for the first time that they need to adjust the "axis" parameter (this parameter controls the size and direction of an arrow object) of the magnetic field arrow for each new location of the source, a moving point charge. Additionally, students cannot place the arrow function inside of the loop, because this will lead to the creation of a new arrow on every iteration, rather than the desired goal, which is an adjustment of the parameters of a single arrow. The alignment between the surplus of "Applying Conditional Logic" codes and the "Setting the axis of the arrow to the magnetic field" codes is indicative that this particular stage of the task excels at eliciting CT engagement in this area, a fact which can be utilized for assessment-design purposes.

By looking at a single CT practice across all tasks, we can determine how prevalent this CT practice is in a given activity and in which tasks it is most readily present. "Utilizing Generalization" is one CT practice that appears in across most tasks in Table II. "Utilization Generalization" is associated with importing a general approach from other examples of programs (either a computational strategy, or in some cases, specific code). Students have worked with models of vector fields (static electric fields, specifically) previously. It follows that students would seek to implement similar strategies to this problem, based on what they had seen previously in the course. From an activity design perspective, this gives us a marker that this magnetism MWP activity has enough code to scaffold the necessary model building. Similarly, the percentage of tasks that involve "Utilizing Generalization" could be an interesting design rubric for examining MWP's. From an assessment perspective, it tells us that students are able to recognize when it is appropriate to reuse previous code, and assessments focusing on generalization could be structured around the tasks that were navigated via this CT practice.

IV. LIMITATIONS & CONCLUSION

From this study, we were able to compare students' CT practices with the specific tasks needed to solve the problem, which provided some initial insight into problem design and suggestions for future assessment. One unavoidable limitation of this study is our inability to account for students' engagement with CT when not explicitly stated or demonstrated. If students worked alone, without any discussion amongst a group, there would still be CT engagement present, but it would be fruitless to try to classify this engagement without indicator phrases or vocal cues of any kind, rendering this type of analysis approach impossible. Although students are collaborating in this context, there is surely some level of CT engagement among the students – either as a whole or individually – that is unspoken. Because of this, it is possible (and albeit likely) that there is more CT engagement happening than what is indicated in the video analysis. Additionally, the focus of the analysis is on the quantity of unique instances of CT engagement, instead of on how long the group engages with a given practice. This is because we were more interested in the patterns of CT occurring in tasks; however, one could envision looking at the duration of practice engagement in the future. Theoretically, the amount of time a group spends on a given task could be indicative of difficulty or complexity of that particular task. On the other hand, spending a large amount of time on a individual instantiation of a CT practice could be indicative of an inappropriate use of a CT practice for a particular task by the group.

Future work should involve the addition of further groups to analyze for this particular problem. The dynamics and resources that each group can call on are going to be different; completing the analysis for subsequent groups could further validate the patterns identified in this initial analysis or highlight alternatives. Additionally, examining the background and dynamics of the groups in more detail and connecting it with the combined analysis of task and CT practice could tell us about the impact of group composition on engagement with activities and CT practices. Linking tasks with CT practices provides us with preliminary insight for assessment development by grounding CT practices in specific contextually appropriate examples of the types of tasks that activate particular CT practices. For example, if we wanted to design an assessment item that focuses specifically on applying conditional logic, then choosing the task of setting and manipulating an arrow axis as the context for that problem may be appropriate. In summary, applying the Weller et al. CT framework in conjunction with a task analysis of a particular activity has provided insight into the alignment of the activity with the CT practice framework and has highlighted connections between specific tasks & students' engagement in resolving those tasks via specific CT practices.

ACKNOWLEDGEMENTS

This work was supported by the National Science Foundation (DRL-1741575) and MSU's Lappan Phillips Foundation.

[1] D. Mohaghegh and M. McCauley, Computational thinking: the skill set of the 21st century, International Journal of Computer Science and Information Technologies 7, 1524 (2016).

[2] N. R. Council, A Framework for K-12 Science Education (2012).

[3] S. Bocconi, A. Chioccariello, G. Dettori, A. Ferrari, K. Engelhardt, P. Kampylis, and Y. Punie, Developing Computational Thinking in Compulsory Education - Implications for policy and practice, Joint Research Centre (JRC) , 1 (2016).

[4] C. Brackmann, D. Barone, A. Casali, R. Boucinha, and S. Munoz-Hernandez, Computational thinking: Panorama of the Americas, 2016 International Symposium on Computers in Education, SIIE 2016: Learning Analytics Technologies , 1 (2016).

[5] H. J. So, M. S. Y. Jong, and C. C. Liu, Computational Thinking Education in the Asian Pacific Region, AsiaPacific Education Researcher 29, 1 (2020).

[6] J. M. Wing 2008 Computational thinking and thinking about computing. Phil. Trans. R. Soc. A.3663717-3725

[7] J. M. Wing, Computational thinking's influence on research and education for all, Italian Journal of Educational Technology 25, 7 (2017).

[8] D. Barr, J. Harrison, and L. Conery, Computational Thinking: A Digital Age Skill for Everyone, Learning and Leading with Technology 38, 20 (2011).

[9] C. Sneider, C. Stephenson, B. Schafer, and L. Flick, Exploring the Science Framework and NGSS: Computational Thinking in the Science Classroom, Science Scope 038, 10 (2014).

[10] D. Weller et al. Developing a learning goal framework for computational thinking in computationally integrated physics classrooms. (2021).

[11] B. V. Barr and C. Stephenson, Bringing Computational Thinking to K-12: What is Involved and What is the Role of the Computer Science Community?, ACM Inroads 2, 48 (2011).

[12] M. Berland and V. R. Lee, Collaborative strategic board games as a site for distributed computational thinking, International Journal of Game-Based Learning 1, 65 (2011).

[13] K. Brennan and M. Resnick, New frameworks for studying and assessing the development of computational thinking, AERA , 135 (2012).

[14] D. Weintrop, E. Beheshti, M. Horn, K. Orton, K. Jona, L. Trouille, and U. Wilensky, Defining Computational Thinking for Mathematics and Science Classrooms, J. Sci. Educ. Tech. **25**(1), 127 (2016).

[15] AAPT Undergraduate Curriculum Task Force, AAPT Recommendations for Computational Physics in the Undergraduate Physics Curriculum, American Association of Physics Teachers (2016).

[16] J. Biggs, Enhancing teaching through constructive alignment. High Educ 32, 347-364 (1996).

[17] G. Wiggins, Grant, and J. McTighe, (1998). Backward Design. In Understanding by Design (pp. 13-34). ASCD.

[18] S. Weatherford, 2011, Student Use of Physics to Make Sense of Incomplete but Functional VPython Programs in a Lab Setting, North Carolina State University.

[19] P. Hamerski, Centering Students' Perspectives in Computation-Integrated Physics Curricula. PhD Thesis. (2021) pa.msu.edu/sites/_pa/assets/File/dissertation_hamerski.pdf

[20] J. Roberts, Limits to communities of practice, J. Manage. Stud. 43, 623 (2006).

[21] A. Pawlak, Collaborative Learning Environments in Introductory Physics, Michigan State University, 2018.

[22] A. Pawlak, P. W. Irving, & M. D. Caballero, Learning assistant approaches to teaching computational physics problems in a problem-based learning course. Physical Review Physics Education Research, 16(1), (2020)

[23] J. Bergmann and A. Sams, Flip Your Classroom: Reach Every Student in Every Class Every Day (2012).

[24] P. W. Irving, M. J. Obsniuk, and M. D. Caballero, P3: A practice focused learning environment, Eur. J. Phys. 38, 055701 (2017).

[25] P. W. Irving, D. McPadden, & M. D. Caballero, Communities of practice as a curriculum design theory in an introductory physics class for engineers. (2020)

[26] L. Torp and S. Sage, Problems as Possibilities: Problem Based Learning for K12 Education (Association for Supervision and Curriculum Development, Alexandria, Virginia, 1998).

[27] K.M. Edens, Preparing problem solvers for the 21st century through problem-based learning, Coll. Teach. 48, 55 (2000)

[28] K. Handley, A. Sturdy, R. Fincham, and T. Clark, Within and beyond communities of practice: Making sense of learning through participation, identity and practice, J. Manage. Studies 43, 641 (2006).

[29] R. Chabay, B. Sherwood, Restructuring the introductory electricity and magnetism course, American Journal of Physics 74, 329 (2006)

[30] https://www.glowscript.org/#/user/botttheo/folder/AAPTPERC2022/program/MWP-unchanged

[31] https://www.glowscript.org/#/user/botttheo/folder/AAPTPERC2022/program/MWP-finished

[32] W. P. Vogt, and B. Johnson, (2011). Dictionary of statistics & methodology: A nontechnical guide for the social sciences. Sage.