Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 172 (2020) 655–659

Procedia Computer Science

www.elsevier.com/locate/procedia

9th World Engineering Education Forum, WEEF - 2019

Problem Solving Archetype - Computer Science

Douglas Lecorchick[1], Scott Nichols, Lauren Tabor,

Berea College, CPO 2188, Berea, KY 40404, USA
Maryland State Department of Education, 200 W. Baltimore Street, Baltimore, MD 21201, USA
Suzhou North America High School, Suzhou - China

Abstract

Problems that are carefully formulated lead students to develop more sufficient and realistic solutions. By front-loading the process of problem solving through problem formulation, students are able to reduce the amount of time spent on solution development, and thus increase their efficiency towards meeting their main objective. By teaching students problem formulation, especially in computer science related activities, foundational skills in computational thinking are introduced, used, and refined. Using a problem solving archetype as a means for this formulation is an effective tool for students to leverage. As computational thinking skills are honed, these concepts can translate across barriers into other content areas.

Keywords: problem formulation; computational thinking; problem solving archetype;

1. Background

It is commonplace for one to view education as a series of activities with the end goal being to learn and understand new information and to solve complex problems. However, problem formulation is not being explicitly taught in schools even though better formulated problems allow the solver to suggest more acceptable solution ideas. Better formulated problems also minimize time spent solving for the wrong problem and suggesting solution ideas that ultimately get rejected.

Better formulated problems allow solvers the opportunity to utilize computational thinking skills; decomposition, pattern recognition, abstraction/generalization, and algorithm construction. By dissecting a problem and identifying

1

* Douglas Lecorchick, Ed.D. Tel.:+1 724 466 2037 ;
E-mail address: lecorchickd@berea.edu

10.1016/j.procs.2020.05.085

common patterns, solvers can generalize methods to future problems with similar constraints and specifications. This lends itself tremendously to problems involving computer science/computing. Global economy and workforce needs indicate there is a deficit of workers that have the capability to dissect a large-scaled problem into smaller, more manageable sub-tasks, and develop an algorithm to address the large-scaled problem.

A study engaged by the National Science Foundation (NSF) suggested that computational thinking includes a series of dimensions including, "confidence in dealing with complexity, persistence in working with difficult problems, tolerance for ambiguity, the ability to deal with open-ended problems, and the ability to communicate and work with others to achieve a common goal or solution" (Barr, Harrison, & Connery, 2011, p. 21). As students circulate through the P-12 system, the development of these skills is necessary for success in industry or post-secondary studies. Additionally, the skills embedded in computational thinking (CT) theory is transferable to practically any discipline, thus preparing better students in all fields. The aforementioned NSF reinforces "developing examples of what CT skills look like in the classroom as well as assembling resources to support and guide the implementation of computational thinking concepts in PK–12 education" (Barr, Harrison, & Connery, 2011, p. 22). Essentially, enhanced problem formulation provides students with the opportunity to develop and implement a skillset that can be utilized on a broad spectrum of problems.

Furthermore, as content areas fight to maintain their stake in the PK-12 education arena, there has been a significant push for up-to-date, robust standards. In 2017, the Computer Science Teachers Association (CSTA) released their newly updated K-12 standard, with a main goal of expanding access to computer science, especially to traditionally underrepresented groups and female students. Aligning with the idea of problem formulation as a vehicle for computational thinking, CSTA introduced middle school standards that require students to "decompose problems and subproblems into parts to facilitate the design, implementation, and review or programs" (CTSA, 2017). Additionally, as found in the standards, it is being suggested that students at the primary level "decompose (break down) the steps needed to solve a problem into a precise sequence of instructions" (CSTA, 2013), which signals the fact that students should be developing algorithmic thinking skills in their early years. The addition of these computational thinking skills into the already full palette of teachers across various contents can be challenging. The NSF has convened "a diverse group of educators with an interest in CT from higher education, PK-12, and industry to help define a common language surrounding computational thinking, articulate the challenges and opportunities of integrating it throughout PK-12 education, and identifying the most promising practices and strategies for moving computational thinking from concept to deep integration" (Barr, Harrison, & Connery, 2011, p. 20). Thus developing a problem formulation model, that can be used across the contents, by all students and teachers, would lead to more sequential and algorithmic thinking styles, as well as, result in better formulated solutions.

Recognition of the need for both problem definition and computational thinking skills is spreading to content areas other than traditional engineering and technology fields. Many and varied organizations including the National Science Teachers Association, American Association for the Advancement of Science, and the National Research Council worked together in developing the Next Generation Science Standards (NGSS), a set of science standards which includes the Science and Engineering Practices. As part of the rationale for including these practices in the standards, the National Research Council's *Framework* states "Any education that focuses predominantly on the detailed products of scientific labor—the facts of science—without developing an understanding of how those facts were established or that ignores the many important applications of science in the world misrepresents science and marginalizes the importance of engineering." (NRC Framework, 2012p. 42-43) Asking Questions and Defining Problems is the first of the Science and Engineering Practices while Using Mathematics and Computational Thinking is the fifth practice. When published in April 2013, the NGSS standards recognized problem formulation and computational thinking as essential components of an education in science.

Finally, the result of better formulation problems, preparing students with the requisite skill set to be successful industry of post-secondary is paramount in today's global economy and workforces. Chae et al. (2005) posit, "As organizations become ever larger and increasingly complex, they become more reliant on information systems and decision support systems (DSS), and their decisions and operations affect a growing number of stakeholders" (p.

197). Inevitably, these referenced decision support systems and information systems must be managed, supported, and implemented by rising students, thus reinforcing the need for students with computational thinking mindsets. As stated, as industry and global economy expands, these decisions are offering a broader impact on the stakeholder groups, thus making it even more integral for students to possess these skills.

All fields which use and collect data as part of decision making, from traditional scientific research to marketing, are facing the relatively new problem of overabundance of data. The new field of "e-science" has been proposed as a field dedicated to addressing this issue within scientific research (National Research Council, Report of a Workshop on the Pedagogical Aspects of Computational Thinking).

2. Contribution

Using a problem formulation archetype, shown in Figure 1 as a way of teaching and guiding students through the problem solving process by specifically engaging in problem formulation by exploring the unmet need first and then presenting solution ideas allows one to become a more effective and efficient problem solver. A solver using an archetype can list the unmet need and progress through the document toward reaching an acceptable solution. This progression allows the solver to explore the unmet need and the problem setting before suggesting a solution. This exploration should provide better solution ideas and provide a more efficient navigation of problem formulation within the problem solving process.

An archetype of the Computer Science Design Process (CSDP) derived from the closed-loop structure of the Engineering Design Process and Scientific Method, utilizing feedback from prior attempts to inform next steps could provide solvers with direction regarding next steps toward solution identification and implementation within computer science activities. Similar to the idea of problem formulation in the beginning steps in terms of breaking down a problem into incremental steps. This allows for much easier redesign of solutions as solvers can identify the exact point/location of failure based on incremental sub-tasks.

By developing a problem formulation archetype, which can be utilized across the content, but is geared towards computer science, students are not locked down to a rigid problem-solving process. As seen in the diagram, the archetype developed provides a fluid model for students to identify all necessary constraints and needs prior to identifying a solution. Then students can explore the sequence of steps, or algorithm, they think will effectively solve the problem, including sub-problems that may arise. When utilising a traditional problem-solving model, students generally do not encounter any extraneous sub-problems until they have already built their solution, prototype, or program. By outlining the implementation procedure, which is merely another description of an algorithm, extraneous problems will expose themselves more readily before a significant portion of work is completed towards the solution. Solutions can be looked at from an abstract point-of-view and generalized to future problems and tasks.

The problem-solving archetype facilitates integration of both problem formulation and computation thinking into courses and content where such thinking has not been traditionally emphasized. For example, when following the NGSS standards k-12, by the end of high school students are expected to be able to "Define a design problem that involves the development of a process or system with interacting components and criteria and constraints that may include social, technical, and/or environmental considerations" in their science classes (Appendix F, NGSS). The problem-solving archetype easily guides students through finding a problem (the unmet need), identifying criteria (requirements) and constraints (both current and anticipated), and then using information about the social, technical, and environmental considerations (need setting and existing systems in use) to further inform their criteria and constraints. Modifying the original criteria and constraints based on new consideration of influencing variables and then identifying sub-problems with the solution (interacting components, in the standard) follows the model for computation thinking outlined above, without having to explicitly introduce computational ideas into the archetype. The implementation procedure element of the solution then addresses the need for the development of a process or

system. Thus, the problem-solving archetype can be used in classrooms where the teacher is not an expert in problem formulation and computational thinking, thus making these skills more accessible to students.

In a different approach, the problem-solving archetype can be used to enhance the integration of computational thinking into the traditional content areas. Computational thinking can be integrated into all content areas. Barr and Stephenson (2011) have developed multiple examples of this, including finding patterns when comparing sentence structures in language arts classes and using computer simulations of historical events in social studies courses. In a specific example in Biology, students were tasked with using code and rules to modify simulations of predator-prey interactions and firefly group interactions. Part of developing "rules" within these simulations was considering the actions of individual animals in response to the environment around them to make assumptions, which were written as "rules" into the simulation (Wilensky & Reisman, 2006). This would be an excellent opportunity for students to use the problem-solving archetype to think about the problems, constraints, and criteria for how species interact and then how to implement rules as their solution.

2.1 Problem Solving Archetype

Figure 1. Problem Solving Archetype.

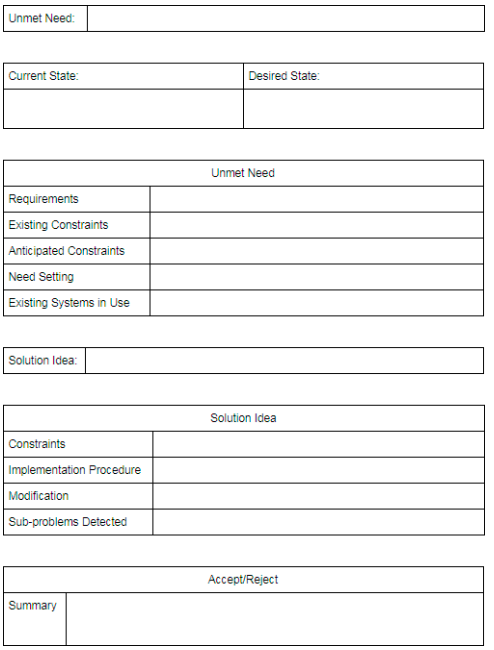

Figure 1. Problem Solving Archetype

3. Discussion

Further the ability for academic exercises to more closely represent problems one experiences in work and life and an enhanced ability to successfully identify and apply solutions to complex and ill-structured problems.

While many content areas have crossover to some extent, computer science/computing is the only content that touches all other contents and reaches all corners of the industry. In every post-secondary study, profession, or future endeavour, students are required to solve problems that are aligned to real-world experiences. While critical-thinking skills have been effective in the past, the 21st century is beginning to require problem-solvers to possess a

greater skill set that includes concepts embedded in computational thinking. Problem-solvers a required to solve multi-faceted, complex problems that contain levels of ambiguity and unknown. By developing a problem formulation archetype that can guide problem-solvers through the process in an algorithmic way that helps identify sub-problems and other issues that could impact the solution, more high-quality solutions can be created. Furthermore, based on the archetype and its integration of computational thinking-style skills, if used effectively, solutions can be generalized to future scenarios, as well.

The problem-solving archetype will be used at Suzhou North America High School to enhance student understanding of the engineering design process. In an introductory, interdisciplinary course designed to deepen student understanding and skills in STEM, students will work through a 4-week unit focused on engineering design. First, the students will use the problem-solving archetype to map the UN's Sustainable Development Goals as problems, sub-problems, interacting components, and the solutions developed by the committees on the project. The students will then use the problem-solving archetype when working on their own engineering design challenge, based on one of the Grand Engineering Challenges. Students will have to use computation thinking skills developed in earlier units, such as use of If, Then statements in Excel, to justify their solution process. We anticipate that as a result, students will have solutions that better fit the original problem and are more capable of independent revisions to their designs.

References

[1] Barr, D., Harrison, J., & Conery, L. (2011). Computational thinking: A digital age skill for everyone. *Learning & Leading with Technology*, *38*(6), 20-23.

[2] 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.

[3] Chae, B., Paradice, D., Courtney, J. F., & Cagle, C. J. (2005). Incorporating an ethical perspective into problem formulation: implications for decision support systems design. *Decision support systems*, *40*(2), 197-212.

[4] *CSTA K-12 Computer Science Standards.* (2017). (2-AP-13: 6-8, 1A-AP-11: K-2). Chicago, Il. Retrieved from https://www.csteachers.org/Page/standards.

[5] National Research Council (2012). A framework for K-12 science education: Practices, Crosscutting Concepts, and Core Ideas. Washington, DC: The National Academies Press. https://doi.org/10.17226/13165.

[6] Next Generation Science Standards (2013). Appendix F - Science and Engineering Practices in the NGSS. Retrieved from https://www.nextgenscience.org/resources/ngss-appendices.

[7] Wilensky, U., & Reisman, K. (2006). Thinking like a wolf, a sheep, or a firefly: Learning biology through constructing and testing computational theories - an embodied modeling approach. *Cognition and Instruction*, 24(2), 171-209.