Teaching Computational Thinking Skills in C3STEM with Traffic Simulation

Anton Dukeman, Faruk Caglar, Shashank Shekhar, John Kinnebrew, Gautam Biswas, Doug Fisher, and Aniruddha Gokhale

Department of Electrical Engineering and Computer Science
Vanderbilt University, Nashville, TN, USA
{anton.dukeman, faruk.caglar, shashank.shekhar, john.s.kinnebrew, gautam.biswas, douglas.h.fisher, a.gokhale}@vanderbilt.edu

Abstract. Computational thinking (CT) skills applied to Science, Technology, Engineering, and Mathematics (STEM) are critical assets for success in the 21st century workplace. Unfortunately, many K-12 students lack advanced training in these areas. C3STEM seeks to provide a framework for teaching these skills using the traffic domain as a familiar example to develop analysis and problem solving skills. C3STEM is a smart learning environment that helps students learn STEM topics in the context of analyzing traffic flow, starting with vehicle kinematics and basic driver behavior. Students then collaborate to produce a large city-wide traffic simulation with an expert tool. They are able to test specific hypotheses about improving traffic in local areas and produce results to defend their suggestions for the wider community.

Keywords: Computational Thinking, Smart Learning Environments, Simulation, Visual Programming

1 Introduction

There is an increasing awareness that the United States is not doing well in K-12 STEM education. In a science proficiency assessment that included 400,000 students from 57 countries, the U.S. ranked 25th of 30 developed nations, with 25% of its students at or below proficiency (the largest percentage among the 30 developed nations) [1]. Improved STEM education in high schools will better prepare students for college education in the STEM disciplines, an important requirement for the 21st century workforce [2]. Developing paradigms that combine STEM learning and problem solving along with preparation for future learning is critical for our nation's future [3].

One way to revitalize STEM education is to make learning engaging and ubiquitous through real-world problem solving that extends beyond the classroom and into the community. In addition, cyber-enabled educational infrastructure that seamlessly integrates personalized and collaborative learning will further advance engagement and participation in STEM education. In this paper, we lay the initial groundwork for an innovative community-situated, challenge-based, collaborative learning environment (C3STEM) that harnesses computational thinking, modeling, simulation, and problem-solving to support ubiquitous STEM learning.

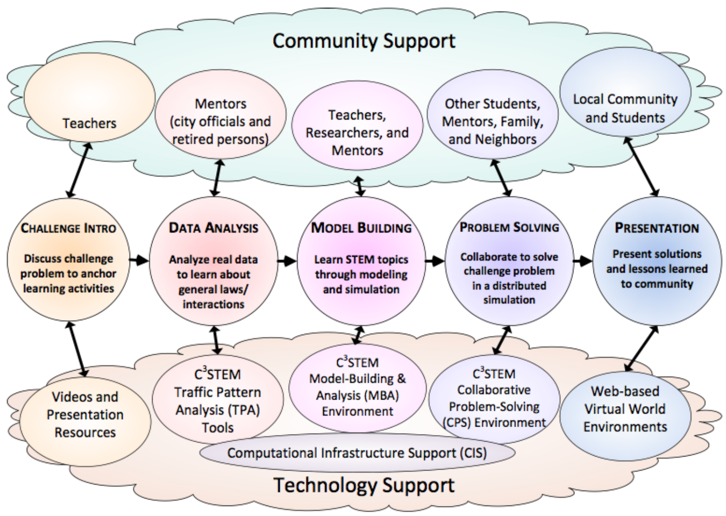

Fig. 1. C3STEM Workflow

Through these combined learning activities and larger-scale collaborations across schools, there will be concrete curriculum-related science and mathematics lessons that the activities will exercise and remediate. Figure 1 illustrates the workflow for participants in C3STEM. The proposed system will support a challenge-based curriculum that integrates STEM concepts, model building, problem solving, and collaboration. Through C3STEM projects high school students will collaboratively address problems of traffic congestion and safety in urban and suburban environments. The traffic domain is attractive because it is a rich source of STEM-related problems, data and simulations are readily available for building our pedagogical tools, and we have local expertise in modeling transportation systems. In particular, small classroom-based student groups will

1. identify traffic patterns by analyzing actual traffic data collected in the form of streaming video by city and state traffic departments;
2. interact with traffic engineers, city planners, and the C3STEM research team to understand how to analyze the traffic data, model traffic patterns, and develop and analyze their solutions;
3. develop agent-based models that align with the observed patterns (e.g., traffic congestion along selected thoroughfares at different times of day and the effects of stoplights and interstate on/off-ramps);

4. design and analyze interventions using their models (e.g., revising traffic light cycles, suggesting carpooling, or extending a public transportation system with additional buses or new hubs, stops, and routes), and observe the effects of these interventions in their region by simulation;
5. use a virtual environment (such as Second Life) and shared simulations to collaborate with student groups in other local schools, adapting and coordinating their region-specific solutions to arrive at globally-consistent solutions for an overall challenge problem; and
6. much like the STAR.Legacy cycle [4], the students will present their solutions to the community at large to receive additional suggestions and further community-wide discussion.

This paper discusses our initial work in designing computer-based tools to support STEM learning by building simulation models of traffic flow, and using the fundamentals learned in the modeling activities to solve problems in a collaborative, high-fidelity simulation of local traffic. Section 2 discusses the CT framework that we employ to help students learn STEM concepts through modeling and simulation. Section 3 discusses our visual programming environment that emphasizes learning of the physics and mathematics concepts. Section 4 then discusses our initial design efforts to construct the collaborative traffic simulation system for problem solving. Finally, section 5 presents conclusions and future work.

2 Background: CT and the CTSiM Environment

C3STEM employs a CT framework to promote effective STEM learning and preparation for future learning. Many of the epistemic and representational practices central to the development of expertise in STEM disciplines are also primary components of CT. Wing has described computational thinking as a general analytic approach to problem solving, designing systems, and understanding human behaviors [5, 6]. CT draws upon concepts that are fundamental to computer science, including practices such as problem representation, abstraction, decomposition, simulation, verification, and prediction. These practices, in turn, are also central to modeling, reasoning and problem solving in a large number of scientific and mathematical disciplines [7].

Although the phrase "Computational Thinking" was introduced by Wing in 2006, earlier research in the domain of educational technology also focused on similar themes, e.g., identifying and leveraging the synergies between computational modeling and programming on one hand, and developing scientific expertise in K-12 students on the other. For example, Perkins and Simmons showed that novice misconceptions in math, science and programming exhibit similar patterns in that conceptual difficulties in each of these domains have both domain-specific roots (e.g., challenging concepts) and domain general roots (e.g., difficulties pertaining to conducting inquiry, problem solving, and epistemological knowledge) [8]. Complementary work by Harel and Papert argued that programming is reflexive with other domains, i.e., learning programming in concert with concepts from another domain can be easier than learning each separately [9]. Along similar lines, several other researchers have shown that programming and computational modeling can serve as effective vehicles for learning challenging science and math concepts [10–12]. Games and simulation can also be effective at encouraging learning. Work at MIT has led to several programs designed to engage students while teaching subjects such as electromagnetism and the American Revolution [13]. Some groups have even taken commercial games not originally meant for education and developed modules conforming to core curriculum requirements. An example is SimCityEDU, for which lesson plans ranging from math to social interaction have been developed [14].

To engage students in computational thinking and STEM learning, we employ the Computational Thinking in Simulation and Modeling (CTSiM) learning environment [15, 16]. CTSiM provides an agent-based, visual programming interface for constructing computational models and allows students to execute their models as simulations and compare their models' behaviors with that of an expert model. In CTSiM, students design, build, simulate, and verify computational models through four interrelated sets of activities:

1. *Conceptualization*: Initially, students conceptualize the science phenomena and mathematical relationships by structuring their model in terms of the types of agents involved, their properties, behaviors, and interactions.
2. *Construction*: With the visual programming interface, students build a computational model by composing and parameterizing available actions/commands with values and properties in the form of visual primitives. Students select primitives from a library and arrange them spatially, using a drag-and-drop interface to generate their computational model, which defines the behavior of individual agents in the simulation.
3. *Enactment*: A microworld, in the form of a multi-agent simulation using NetLogo [17], allows students to simulate and visualize the agent-level behaviors defined by the student in the computational model [18].
4. *Envisionment*: Students go beyond simply simulating their own models by designing simulation experiments to analyze, refine, and validate the behavior of their model with comparison to a simulation of an expert model.

3 Building Traffic Models in CTSiM

Traffic modeling and simulation provides an excellent domain for learning a variety of STEM concepts in a computational thinking framework. There are many different levels of abstraction available from the lowest level involving basic kinematics equations to the interactions between adjacent neighborhoods with several levels in between. In this section, we present the design of initial CTSiM units for the micro-level traffic modeling activities in C3STEM.

3.1 Position, Velocity, and Acceleration

To understand traffic at a low level requires knowing the relationship between position, velocity, and acceleration, so this is where students start. Students are asked to build a simple one dimensional model of a single car on a straight road with zero initial velocity speeding up and then cruising at a specified velocity. Because students are required to learn the relationship between position, velocity, and acceleration in the context of Newton's laws of motion they will develop systematic mathematical equations using the laws to compute the required values. Visual primitives are provided for students to compute the car's position and velocity using the car's current acceleration, velocity, and position. A library of mathematical functions allow students to model acceleration profiles, such as constant, linear, and square root. Students can us the visual environment to build and run these models and they can simultaneously observe the movement of the vehicle on a road as well as plots of position, velocity, and acceleration over time to facilitate their understanding of the concepts.

After completing the first module, students will move to modeling deceleration. Students can model a scenario where a single car moving with an initial velocity comes to a stop at a stop sign and then accelerates to its initial velocity on the road. If the new functions are used correctly, then the computed velocities will be correct for the entire trajectory, and the car will follow a deceleration profile, stop at the stop sign, and accelerate again.

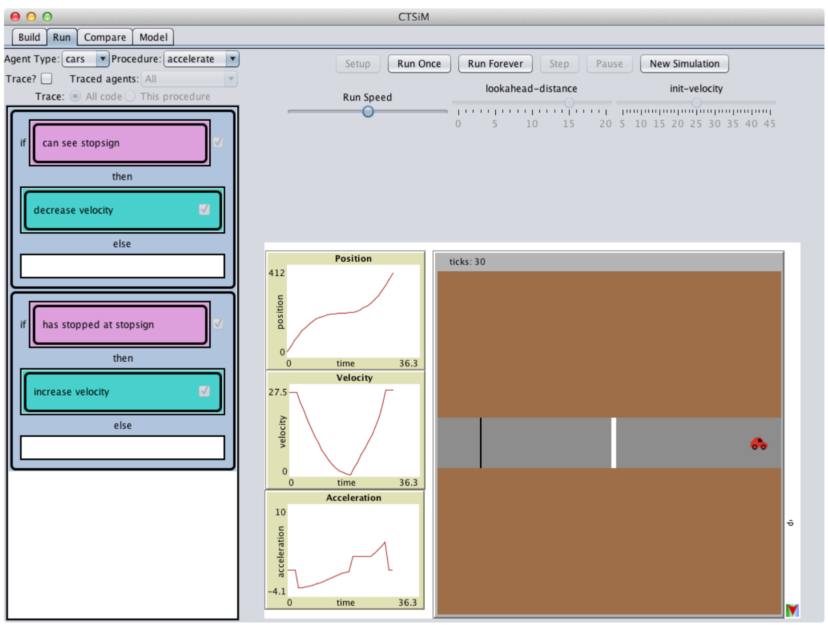

Fig. 2. Enactment world for stop sign model

After learning the basics of kinematics of motion, students move on to study driver behavior and interactions. The second module (enactment world shown in Fig. 2) introduces the parameters, such as *stopping sight distance*, the distance from a stop sign or another stationary object that the driver activates the stop behavior [19]. Students will set two parameters: *initial velocity* and *stopping sight distance* (with sliders visible in the upper portion of Fig. 2) to build two functions, accelerate and go. As any experienced driver knows, the faster the car's velocity, the earlier the driver has to initiate the deceleration function. Depending on the initial parameters, the acceleration plot (bottom plot in Fig. 2) may have a steep, uncomfortable drop rather than a smooth deceleration to stop.

3.2 Scaling up: Modeling Simple Traffic Flows

In this model, students will model multiple cars moving along roads with intersections that have stop signs and stop lights [20]. The road configurations will be provided to the students, and they may represent a traffic network in a section of the city. Like before, cars may enter the environment with an initial velocity or start from rest at different positions. The students' models will scale up to include multiple cars that operate together. As cars approach the stop sign or the end of the queue of cars waiting at the stop sign, they will have to slow down, and then take turns going through the intersection. Some cars may be able to go at the same time, such as two cars traveling in opposite directions on the same road or two cars on perpendicular roads both turning right. Because this module builds on previous ones, students will focus on the modeling issues associated with multiple interacting queues, such as waiting at a stop lights and signs, and following cars. Lane changing functions will be implemented as part of driver behavior, and students will study wave patterns, such as the propagation effects when a car suddenly slows down. In addition to basic physics, this module will introduce students to parameters, such as *average vehicle velocity* and *throughput*, allowing students to learn higher level mathematical analyses and methods for optimizing parameter values given constraints. A subsequent module may introduce *gap acceptance time*, and the *average turn time* at an intersection [21].

4 From Micro to Macro: Using SUMO

Another component of C3STEM is collaborative problem solving. Students will collaborate with other groups at their school and also groups at other schools to develop models of traffic flow for their entire city. Each group will model traffic flow in neighborhoods and connected city streets and highways near their school. Initially, they will interact with the researchers (the authors) and city traffic engineers to understand the traffic flow patterns and related parameters locally and globally (i.e., across the entire city). This study will help them formulate specific solutions to reduce traffic congestion.

For city-wide traffic simulation and analysis, students will interact with a simulation of traffic in the region around their school using web-based tools that display results of a back-end simulation run in the Simulation of Urban MObility (SUMO) environment, a continuous simulation for large road networks

[22]. A web interface using Google Maps will be used as the interface with the simulation server as it is a familiar interface for students. The map interface is easily recognizable and supports a drag-and-drop interface for manipulating the simulation parameters; cars, stop lights, and stop signs can be easily manipulated on the map.

5 Conclusions and Future Work

Computational thinking skills are an essential part of STEM-related careers, and the United States lacks qualified applicants in these disciplines. C3STEM seeks to improve students' STEM learning and problem solving skills using the traffic domain as a motivating example. Students start with the basic low-level physics simulations involving position, velocity, and acceleration in CTSiM. They then move on to model driver behaviors, such as gap acceptance time and stopping sight distance. After mastering the micro-level physics concepts, they move up to SUMO, to analyze and solve more complex analytic problems. At the macro-level in SUMO, students can manipulate intersections in many ways such as changing stop light timing or adding stop signs. Solving complex problems analytically requires students to think at multiple levels of abstraction and draw conclusions across the different layers. We will run experimental studies at two high schools in Chattanooga, Tennessee in the U.S. Students will be given a pretest to measure each student's understanding of physics principles and CT skills. After the program, students will be given a similar posttest to determine the efficacy of the program at promoting both CT and STEM skills in the students.

Some future work involves enhancing the collaboration aspect. Good collaboration requires good communication and this will be provided through a persistent online world. In this environment, students will be able to interact through text and video. They will be able to see the simulation as it is running and share ideas about how to improve it. Students working on adjacent neighborhoods will have to communicate often as they are the most likely to be affected by each other's changes, however, neighborhoods that are far apart may have to communicate as well and will be able to do so. We will also create additional CTSiM modules to simulate other aspects of driver behavior to students, such as lane changing and highway merging.

References

1. Fleischman, H.L., Hopstock, P.J., Pelczar, M.P., Shelley, B.E.: Highlights from PISA 2009: Performance of US 15-Year-Old Students in Reading, Mathematics, and Science Literacy in an International Context. NCES 2011-004. National Center for Education Statistics (2010)
2. Karoly, L., Panis, C.: The 21st Century at Work: Forces Shaping the Future Workforce and Workplace in the United States. RAND Labor and Population Technical Report 0-8330-3492-8 (2004)
3. Board, N.S.: Preparing the Next Generation of STEM Innovators: Identifying and Developing our Nation's Human Capital. Report NSB 10-33 (2010)

4. Schwartz, D.L., Brophy, S., Lin, X., Bransford, J.D.: Software for Managing Complex Learning: Examples from an Educational Psychology Course. Educational Technology Research and Development **47** (1999) 39–59
5. Wing, J.M.: Computational Thinking. Communications of the ACM **49** (2006) 33–35
6. Wing, J.M.: Computational Thinking: What and Why? Link Magazine (2010)
7. Council, N.R.: Taking Science to School: Learning and Teaching Science in Grades K-8. National Academy Press (2007)
8. Perkins, D.N., Simmons, R.: Patterns of Misunderstanding: An integrative model for science, math, and programming. Review of Educational Research **58** (1988) 303–326
9. Harel, I.E., Papert, S.E.: Constructionism. Ablex Publishing (1991)
10. Guzdial, M.: Software-Realized Scaffolding to Facilitate Programming for Science Learning. Interactive Learning Environments **4** (1994) 1–44
11. Sherin, B.L.: A Comparison of Programming Languages and Algebraic Notation as Expressive Languages for Physics. International Journal of Computers for Mathematical Learning **6** (2001) 1–61
12. Blikstein, P., Wilensky, U.: An Atom is Known by the Company it Keeps: A constructionist learning environment for materials science using agent-based modeling. International Journal of Computers for Mathematical Learning **14** (2009) 81–119
13. Jenkins, H., Klopfer, E., Squire, K., Tan, P.: Entering the Education Arcade. Computers in Entertainment (CIE) **1** (2003) 8
14. GlassLab: About SimCityEDU. http://www.simcityedu.org/about/ (2013)
15. Basu, S., Dickes, A., Kinnebrew, J.S., Sengupta, P., Biswas, G.: CTSiM: A Computational Thinking Environment for Learning Science through Simulation and Modeling. The 5th International Conference on Computer Supported Education (2013)
16. Sengupta, P., Kinnebrew, J.S., , Basu, S., Biswas, G., Clark, D.: Integrating Computational Thinking with K-12 Science Education using Agent-based Computation: A Theoretical Framework. Education and Information Technologies (2013)
17. Wilensky, U.: NetLogo. http://ccl.northwestern.edu/netlogo/. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL (1999)
18. Papert, S.: Mindstorms: Children, Computers, and Powerful ideas. Basic Books, Inc. (1980)
19. Fambro, D.B., Fitzpatrick, K., Koppa, R.J.: Determination of Stopping Sight Distances. Volume 400. Transportation Research Board (1997)
20. Horowitz, A.J.: Revised Queueing Model of Delay at All-Way Stop-Controlled Intersections. Transportation Research Record **1398** (1993) 49
21. Pollatschek, M.A., Polus, A., Livneh, M.: A Decision Model for Gap Acceptance and Capacity at Intersections. Transportation Research Part B: Methodological **36** (2002) 649–663
22. Behrisch, M., Bieker, L., Erdmann, J., Krajzewicz, D.: SUMO - Simulation of Urban MObility: An Overview. In: SIMUL 2011, The Third International Conference on Advances in System Simulation, Barcelona, Spain (2011) 63–68