Engaging High School Students in Modeling and Simulation through Educational Media

David R. Musicant
Carleton College
1 N. College Street
Northfield, MN 55057
dmusican@carleton.edu

S. Selcen Guzey
Purdue University
100 N. University Street
West Lafayette, IN 47907
sguzey@purdue.edu

ABSTRACT

The new AP CS Principles curriculum has a significant component regarding modeling and simulation, and many teachers will need to figure out how to accomplish this in their classroom. We have developed a new publicly available media-enhanced approach for teaching modeling and simulation, designated as *TrafficJam.* The approach consists of two core activities, both of which involve students optimizing traffic signals (also known as traffic lights, or stoplights). TrafficJam is likely best distinguished from other classroom simulation exercises in that these activities are introduced, motivated, and demonstrated in a video created by Twin Cities Public Television (TPT). The video uses an inquiry-driven format to feature four high school students who take on the task of improving signal timing in their own neighborhood. A pilot study of TrafficJam in four schools indicates that students find the video engaging, the activities relevant and interesting, and that they gain understanding of modeling and simulation from the experience.

Categories and Subject Descriptors

K.3.2 [**Computer and Information Science Education**]: Computer science education; I.6.0 [**Simulation and Modeling**]: General

General Terms

Algorithms, Experimentation

Keywords

Modeling; simulation; AP CS Principles; media-enhanced

1. INTRODUCTION

Modeling and simulation are dynamic and interactive approaches that can be quite successful at introducing students to computational thinking [4]. The new Advanced Placement (AP) CS Principles course integrates these approaches directly into the "Abstraction" portion of its curriculum, requiring students to use models and simulations to "represent phenomena," as well as to "formulate, refine, and test hypotheses" [8].

The imminent arrival of AP CS Principles charges a large number of teachers to become fluent with a variety of new ideas in computer science [1]. This may be challenging to accomplish. Many existing AP computer science teachers have been teaching the AP CS A course, whose content focuses on object-oriented programming and algorithm analysis, and they will have to learn considerable additional content if they begin to teach AP CS Principles. Many more teachers may begin teaching AP CS Principles who have never taught computer science at all. There is a considerable need for material for this new course, especially content that can be leveraged en masse.

The need for simulation and modeling in high school curricula, and the challenge in bringing it to large numbers of classrooms quickly, served as our main source of motivation in developing a new publicly available media-enhanced approach designated as *TrafficJam.* The approach consists of two core activities, both of which involve students optimizing traffic signals (also known as traffic lights, or stoplights). The activities involve optimizing timing for a single traffic signal via a spreadsheet model, and using a graphical simulation to reduce traffic delay time for an entire neighborhood.

TrafficJam is likely best distinguished from other classroom simulation exercises in that these activities are introduced, motivated, and demonstrated in a video created by Twin Cities Public Television (TPT), an experienced professional nationally-recognized television production team with experience in communicating complex content in ways that engage youth. The video uses an inquiry-driven format to feature four high school students who take on the task of trying to improve signal timing in their own neighborhood. These students in the video learn about traffic signals, collect data, work with a spreadsheet model, experiment with a graphical simulation, and see their ideas implemented. This video is entertaining for students to watch and helps engage them in the activity, but perhaps more importantly, the students in the video serve as role models and guides for real-world students participating in the activity. Furthermore, the video helps teachers to adopt this activity for their classroom. The video is broken up into segments, designed to be shown in-class over multiple days to accompany the work that the class would actually do.

TrafficJam is targeted at students in grades 9–12 who are in their first semester of an introductory computing

SIGCSE'15, March 4–7, 2015, Kansas City, MO, USA.
ACM 978-1-4503-2966-8/15/03.
http://dx.doi.org/10.1145/2676723.2677237.

course. The AP CS Principles Draft Curriculum Framework includes using models and simulation among its learning objectives. We achieve this specifically in Core Activities 1 and 2, which we describe below. TrafficJam is potentially quite useful, however, beyond AP CS Principles. We believe that TrafficJam would serve well in any introductory level computing course where modeling and simulation is a component to be explored. Furthermore, the video can stand alone without the core activities and could be used as a recruiting or motivational tool in order to draw more students – particularly underserved students — to computer science.

TrafficJam is freely available online.[1] It consists of the following contributions:

- A novel approach for optimizing traffic signals that simplifies it appropriately for high school audiences, yet still reasonably accounts for four-way intersections, turning cars, and data collected at such intersections.
- A sample spreadsheet model intended to be used by a teacher to learn about our approach. It is used for optimizing a traffic signal at a single intersection.
- A graphical simulation built in NetLogo for optimizing a neighborhood of traffic lights, which students use to optimize a collection of traffic signals at once.
- Video segments to accompany the classroom activities that engage and guide the students and the teachers.
- A detailed teacher's guide that explains how to implement the exercises and involve students in data collection.

The remainder of this paper is organized as follows. After discussing related work, we present in detail the modeling and simulation core activities. We then describe the creation of and production details for the associated video. This is followed by a description of a pilot study that we ran, and we discuss quantitative and qualitative results from that study. Finally, we conclude and consider future work.

2. RELATED WORK

Previous efforts have used television programs to teach science, with a specific goal of inspiring and engaging kids. One particular approach features students on screen, pursuing their own investigations as they work to solve problems. This approach has been used in different ways in PBS series such as the classic "3-2-1 Contact," and more recent PBS series including "Fetch," "Design Squad," "DragonflyTV," and "SciGirls." Shows such as these feature kids of approximately the same age as the expected audience. These on-screen characters solve scientific problems in much the same way that the audience would, thereby modeling successful behavior. Research has shown that seeing successful behavior modeled can have an significant effect on one's own self-efficacy, i.e, belief in one's own ability to succeed [5]. Television programs with this particular format may act as a source of *vicarious experience* for viewers [5]. Vicarious experience can be quite effective at boosting self-efficacy [12], especially when the models observed are perceived to be similar to the observer [11]. For TrafficJam, we combine this form of television with classroom projects and assignments appropriate for high school students.

The choice that we made to focus on modeling and simulation was motivated in part by the AP CS Principles Draft Curriculum Framework [8], which addresses these topics within the context of "Big Idea 2: Abstraction." Specifically, the framework indicates that one of the enduring understandings conveyed is that "Models and simulations use abstraction to generate new understanding and knowledge."

The specific projects that we develop place these models and simulations in the context of infrastructure engineering, or more specifically, traffic management. Infrastructure engineering is an activity that students are generally not exposed to in high schools within the United States, despite a lack of attention to infrastructure within the US in recent decades [3, 6]. However, most students are aware of traffic problems, and solving a problem in their local neighborhood makes the corresponding activities relevant and meaningful. Traffic engineering has been done previously in high school and/or computer science contexts, though seemingly not very frequently. An Android game was developed to simulate a single intersection in a fairly realistic manner, with the goal of using crowdsourcing to improve timing [7]. This project contrasts with our approach in that we model a single intersection mathematically via a spreadsheet, using data gathered by the students themselves, and then we use a graphical simulation to handle multiple intersections. Our graphical simulation was built with the simulation environment NetLogo [9]. NetLogo comes with a sample simulation of traffic in a grid of intersections [10], which we used as a starting point for our own simulation project. We extended the capability of this simulation considerably by adding two-way streets and turning cars, but much more importantly, we changed the logic within the simulation engine to incorporate user-entered real-world traffic data.

One of our major challenges in developing this project was in creating a traffic signal optimization procedure that was meaningful and interesting, but not too complex for a high school audience. We based our techniques heavily on procedures which we learned from the infrastructure engineering community. Typically, one begins by optimizing the parameters within a mathematical model for each traffic signal in isolation from the rest, based on traffic data collected on site. This optimization is typically done directly and numerically [2]. However, each traffic signal alters traffic flow in a way that affects the rest of the neighborhood around it, and so once all traffic signals are individually optimized their parameters are used as initial conditions for a traffic simulation. Professional traffic management typically employs commercial software, which traffic engineers experiment with to produce an efficient set of timings. This process, which we simplified appropriately, became the framework for the two core activities for TrafficJam. We describe these in the next two sections.

3. MODELING A SINGLE INTERSECTION

Core Activity 1 engages students by having them build a model in order to optimize timing for a single intersection. Specifically, they implement the following steps:

Data Collection (optional). If students wish, they can collect their own data from an intersection near their home or school, as modeled in the project video. The TrafficJam approach assumes a standard simple four-way intersection, with no turn-lanes; specifically, the traffic signal has a single green, yellow, and red light in each direction. The model requires as input data, for a particular time of day, counts

[1] `http://tpt.pbslearningmedia.org/collection/emacs`

of the number of cars per hour that approach the intersection from each direction and how many of them turn left, turn right, and go straight. This data collection activity can be fun as well as empowering for the students, and can help them experience ownership of the activity. Alternatively, we provide such data for students to use if they or their teacher wish to skip this step.

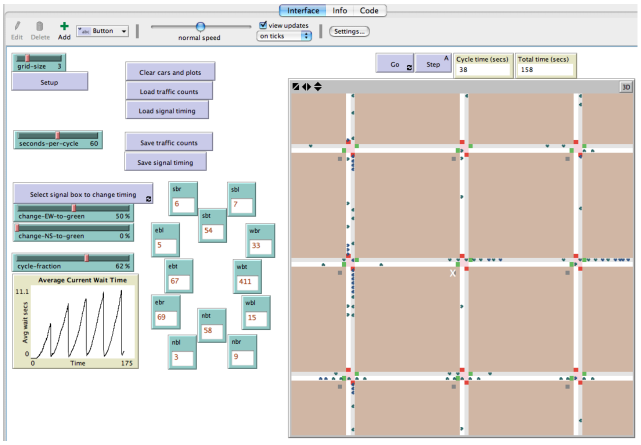
Model construction. This portion of the task requires the students to build a model that uses as inputs both the data described above, as well as user-entered parameters describing how long the traffic signal should be red, yellow, and green in each direction. (The optimization step, described in the next subsection, allows the students to determine which timing numbers are most desirable.) The students then implement calculations to produce an estimate of average delay time per car, which is defined as the average amount of time a car waits at a red light before it can proceed. Building this model requires some significant effort: it requires students to engage in thinking through how to convert number of cars per hour in a particular direction to time delayed per car, based on the input parameters. The TrafficJam materials describe this process to teachers in considerable detail, and also provide a complete solution for a teacher to use in coming up to speed. We expect that a teacher would spend time with the students in the classroom discussing these ideas before students implement the actual task.

There is much flexibility in what sort of tool a teacher might choose for students to use in implementing this model. It requires students to design a somewhat complex model and implement a variety of calculations, but it does not require programming logic such as conditionals or loops. Our video shows students using a spreadsheet to implement the model, which we chose in part because of its simplicity and visual appeal. That said, the same model could alternatively be done in any textual or visual programming environment.

Model optimization. After students have built a model that outputs average delay time for a particular set of inputs, they determine what traffic signal timings, used as inputs, will result in an optimal average delay time. The materials that we provide explain that while a short delay time is good, making it too short would not give pedestrians time to cross the street. Moreover, this is where the collected data plays a major role; the time for red vs. green lights needs to be appropriately balanced across the different directions.

The approach that the students need to implement is well-known to computer scientists: given a black box model that takes inputs and returns an output, find the set of inputs that produces the optimal output values. Teachers implementing this exercise can make this as simple or as fancy as they like. When shooting our video, we allowed the students to experiment themselves with different input values, and to informally develop their own procedure for doing so. They began with a guess-and-check sort of procedure, but as they continued, they discovered patterns within and eventually followed a hypothesis generation process that hints at something like binary search. We use it as a teaching moment in the video to briefly consider the nature of an algorithm.

At the completion of this task, students have their best estimates of the optimal timing for a single traffic signal. They then use this as an input for the next task, which is a neighborhood simulation.



☑ view updates
on ticks
grid-size 3
seconds-per-cycle 60
change-EW-to-green 50 %
change-NS-to-green 0 %
cycle-fraction 62 %
Cycle time (secs) 38
Total time (secs) 158
sbr 6
sbl 7
sbt 54
ebl 5
wbr 33
ebt 67
wbt 411
ebr 69
wbl 15
nbt 58
nbl 3
nbr 9

Figure 1: TrafficJam NetLogo simulation: students optimize timing parameters for each traffic signal separately.

4. SIMULATING A NEIGHBORHOOD

Core Activity 2 invites students to find signal timings for an entire neighborhood, which requires a different approach than optimizing each intersection individually. Changing the timing of the traffic signal on one intersection changes the patterns by which vehicles appear at adjacent intersections. Furthermore, the number of parameters that need to be optimized is multiplied by the number of intersections in the neighborhood. Traffic experts agree that this is too complex to solve via a direct model, and so it is typically done via simulation [2]. For TrafficJam, we chose to implement our own neighborhood simulation software. Most existing tools were quite complicated as well as expensive, and so this gave us the opportunity to create a simulation that fit the simplified traffic optimization approach that we constructed. Furthermore, the freely-available simulation environment NetLogo [9] already had a traffic simulation model [10], though we had to modify it considerably to handle car-count data, different timings for different signals, two-directional traffic, and cars turning. We discussed having students implement their own simulation in NetLogo, but concluded that this task is too difficult for the target audience. However, doing NetLogo programming to add capabilities is something that advanced students may wish to pursue, and our teacher's guide lists some possible extensions. For our expected usage, students do not need to know any NetLogo programming.

The simulation that we provide is for a 3 x 3 grid of intersections, resulting in a total of 9 traffic signals to be optimized (Figure 1). Each one of these intersections requires its own set of input data: specifically, the user specifies for each intersection the number of cars that approach it in each direction, and how many turn left, turn right, or proceed forward. This data is perhaps intractable for students to collect from nine real intersections in their city, though they would be welcome to do so. (There is a wonderful opportunity here to have a side-discussion about the value of crowdsourcing.) Many major metropolitan areas have government websites with car-count data, which could potentially be explored. Classes are otherwise welcome to use the sample car-count data which we supply in our downloadable materials, which reflects actual data from our metropolitan area.

Figure 2: Excerpt from TrafficJam video showing the students meeting with a traffic engineer at the city traffic control center.

Figure 3: Excerpt from TrafficJam video showing students collecting data at their intersection.

Our materials describe in detail the task that the students perform. They first use their results from Core Activity 1 as initial conditions for the center traffic signal, then they adjust a variety of timing parameters associated with each traffic signal to influence the location and severity of traffic jams. Data is displayed which shows on average how long every car in the grid waits at a red light. Students strive to reduce that statistic, as well as achieve a general sense of good flow by looking at the graphical simulation.

5. VIDEO PRODUCTION

The TrafficJam video was produced as a collaboration between TPT and academic computing educators. It was decided early on that we would use an inquiry-driven format featuring real high school students solving problems on-screen. (See Section 2 for more details on the benefits of such an approach.) Casting was done by visiting high schools in our metropolitan area and holding auditions. Since there were no AP CS Principles pilot courses being taught in the schools in which we auditioned, we did not recruit students directly from that course. Instead, we cast students who were smart, enthusiastic, had good presence on-screen, and who did not know very much about computer science. One of our goals was to have the students in the video serve as role models for those watching the video, and so we also strove to cast students with varying ethnicities and genders.

The students that we cast were not professional actors, and this was by design. The goal was to show them honestly experiencing on-screen the process of solving problems and the excitement of success. Therefore, most of the video was not scripted word-for-word, as this would never appear natural. Rather, most of the video was outlined by a professional writer, with input from the rest of the team. This outline indicated what activities would happen in each scene, and what concepts were intended to emerge. Most of the actual work that students do in the video is genuine footage of them working to solve problems. To help tie everything together, we did construct a fictional meta-story describing the students as a group of friends who respond to an invitation to work with their city in optimizing timing for a traffic signal close to their home, and end up with their new timing parameters being implemented on that traffic signal. These fictional portions are only outlined as well, and not scripted. For each segment, the general setup was explained to the students with some suggestions that they implemented via role-playing.

The complete 18-minute video that we created consists of the following sections, which are designed to be shown in portions to students as they work on associated activities:

1. “Traffic Jam”: This first segment introduces the four students, and shows them responding to an invitation to work with the city to improve timing on a traffic signal near their home. See Figure 2.
2. “Straight to the Source”: The students visit their city’s traffic management facility, tour the main traffic signal control room, and meet with a traffic engineer who answers some of their questions.
3. “Brainstorm!”: The students describe their task to a computer science professor, who works with them to figure out what data they need to collect.
4. “Take the Numbers”: The students go to the intersection of interest and collect the data they need. See Figure 3.
5. “Make a Model”: The professor from the “Brainstorm!” segment visits with the students again, introducing them to modeling. They experiment with a spreadsheet model in order to optimize the timing for a single traffic signal.
6. “Run a Simulation”: The students learn how to use a NetLogo simulation of traffic in an entire neighborhood, optimize timing for 9 traffic signals, and reflect on results.
7. “Mission Accomplished”: The new timing for the traffic signal near their home is put in place, and the students celebrate their success on location at the intersection.

Classroom teachers might choose to show these segments in a classroom in differing ways. Our recommendation, which is described in detail in our associated materials, breaks this up into a four day task. On the first day, a class would watch the first four portions of the video, discuss what they see with their teacher, and lay out a plan for data collection. The second day is optional, and involves the class going to an intersection to collect their own data. The third day involves showing the “Make a Model” segment, followed by the class actually building their own models. On the last day, the class watches the “Run a Simulation” segment, experiments themselves with the NetLogo simulation, and then watches the “Mission Accomplished” wrapup. An alternative approach is to use a flipped-classroom arrangement where students watch the videos on their own, either in daily chunks or all at once. And, as mentioned earlier, the video can also be seen as a standalone engagement tool.

Constructing the portion of the video that shows the students working with the spreadsheet model was particularly challenging, and was a learning experience for us. We had originally hoped to show genuine footage of the students building the spreadsheet model from scratch. However, we had chosen the cast members to be students who did not have much computer science experience, as their reactions and behavior would best model students who might be near the beginning of a course such as AP CS Principles. It was always our assumption that in an actual course, once a teacher had chosen a tool for the students to use for modeling (spreadsheet, or other programming environment), the students would have the opportunity to work on more basic assignments to learn how to use that tool. There was no time or funding for us to do all that with our four cast members. In the end, we pointed our students to an online tutorial on spreadsheets before we shot the video, and then had them work on video with a spreadsheet that we had already prepared in advance for them. This approach worked well for us in that our students were still able to experiment with the model we created in order to perform optimization, and there were many valuable discovery moments for the students as part of that process.

6. PILOT STUDY

We chose to apply a quasi-experimental research design to investigate the change on students' engagement in, attitudes towards, and understanding of computer science. We focused our attention on the following research questions:

- Do students find the video engaging?
- Do students find the lesson engaging?
- Does the video increase interest in and confidence in CS-related activities?
- What aspects of the video do students like most/ least?

6.1 Study design

We found four high school teachers that were willing to have their students participate in the study. Two of these teachers agreed to both show our video and spend a number of full class days with their students working through the associated classroom activities. The other two teachers agreed to show the video only, as that was easier and quicker to implement. To make the distinction clear going forward, we refer to the first group as the "whole-project" group, and the second group as the "video-only" group.

Group demographics: The whole-project group consisted of 39 students, and the video-only group included 37 students. A large group of students in the study (whole-project, 57% and video-only 66%) were White/Caucasian. In both groups, approximately half the students were in grades 9–10, with the other half in grades 11–12. All students were enrolled in some sort of computing-related class.

Data collection: The study used a two-group pre-test and post-test non-randomized experimental design. Whole-project students filled out the pre-viewing survey, viewed the video and completed the classroom activities. Afterwards, students were interviewed regarding their opinions about the video and classroom activities. Each focus group interview took about 30 minutes. Video-only group students filled out the pre-viewing survey, viewed the video and then completed the post-viewing survey. Video-only students were also interviewed regarding their opinions about the video.

Survey design: The pre- and post-surveys were not identical. The pre-survey was designed to measure student attitudes towards and interest in computer science and to collect demographic information. The post-survey for the whole-project group was divided into three parts. The first part, designed to measure engagement in the project, asked about how much students liked the video and the classroom activities. The second part, which was intended to measure student attitudes, asked about student interest in computer science in general. Finally, the third part was designed to measure understanding, and tested student learning of targeted computer science concepts. The post-survey for the video-only group was very similar, though it left out all content related to the classroom activities. In addition to these surveys, focus group interviews were conducted mainly for feedback regarding the TrafficJam video.

6.2 Study Results: Quantitative Data

Engagement: Both groups of students rated the video engaging and interesting. One of the main reasons that students were so engaged by the video was its applicability. Over 75% of the whole-project students and over 85% of the video-only students commented that they liked the video because it was a real-world scenario. Students felt that the challenge in the video was personally relevant and engaging. Furthermore, over 50% of both groups indicated that they liked each section of the video by rating it with a 3 or 4 on a scale of 1 (not at all) to 4 (like it a lot), although students tended to like some sections more than others (e.g., they liked the "straight to the source" considerably). Finally, the whole-project students found the classroom activities engaging. They particularly liked working on NetLogo and working in groups. A large number of students wanted to watch more videos like it.

Attitudes: Pre-survey results showed that students in both groups had very a high level of positive attitudes towards and interest in computer science before viewing the video. Due to this high level of initial motivation, the pre-survey and post-survey results for both groups of students showed similar results. Furthermore, no substantial difference was observed between the whole-project students' and video-only students' interest in computer science.

Content Understanding: 83% of the whole-project students indicated via self-reflection that they learned from the TrafficJam video and classroom activities while 72% of the video-only students indicated that they learned from the video. In addition to the self-reflection questions, the post-survey included several multiple choice and true/ false questions about the spreadsheet, model, and simulation in the video. Since whole-project students participated in classroom activities, they were better in particular at questions that were written to measure learning about the spreadsheet and models. The classroom activities that whole-project students participated in helped them better understand certain targeted computer science concepts (such as optimization and graphical simulation).

6.3 Study results: Qualitative Data

As noted earlier, focus group interviews were conducted with both groups of students. Open ended questions gave students the chance to reflect on the video or classroom activities and their learning. The majority of the students expressed that they liked the video, since the challenge within the video is a real-world problem. Students also expressed that it is encouraging to see and learn that they can solve real-world problems. The following are several representative responses by students to the open-ended question "What did you like about the video?" These are direct quotes, and so grammatical errors have not been corrected:

- "I like the fact that it was a real case scenario."
- "The actors were funny and the problem was common."
- "How easy it is to solve daily problems in life and how hardworking the kids were."
- "I like it was in [my own state]."
- "I like how the kids were able to positively affect their neighborhood via simulation."
- "Data collection"
- "It really utilized the structured teamwork and engineering problem-solving to help make the community better!"
- "I like that they actually were able to make a difference with computers."
- "I like how the kids went all the way to the main source to fix the problem that they had."
- "Interesting, informative, direct. Not too wacky or crazy."

While many students liked the video, a few students did not. Those who do not like the video generally thought traffic control was uninteresting. Some of those students also did not like the acting of the students in the video, finding it "cheesy." This was useful feedback; some of the techniques for producing videos such as this one that are successful for general science broadcast programming may be perceived differently for productions to be shown in school, and we would hope to consider this in detail for future work.

In summary, the pilot study gathered data from 76 students in grades 9–12. A majority of the students found the TrafficJam video and classroom activities very engaging. The whole-project students were overwhelmingly positive about the classroom activities associated with the video. While acknowledging some issues such as the acting performance of the students in the video, most agreed that they found the experience very valuable.

7. CONCLUSIONS

The new AP CS Principles curriculum has a significant component regarding modeling and simulation, and many teachers will need to figure out how to accomplish this in their classrooms as the course is adopted by more schools. We have developed a set of materials, which could be used in AP CS Principles or in other contexts, to introduce modeling and simulation. Our work consists of an approach for making this level-appropriate for a high school setting, a set of activities for students to work on, and a set of professionally made videos designed to introduce the ideas and engage the students. A pilot study indicated that students mostly found the activities relevant and interesting, and that they gained a greater understanding of modeling and simulation from the experience.

Looking forward, we hope to evaluate the effectiveness of TrafficJam on student success directly in an actual AP CS Principles course. This will be more easily accomplished as more schools begin to adopt the curriculum. We also plan to develop new media-enhanced projects, using the format of this one, to help improve other aspects of computer science education. We suspect that our approach of having computer science activities accompanied by videos showing inquiry-driven explorations might be particularly effective at engaging and motivating new students into the field, especially those who do not already have a high level of interest. We hope to research this further and produce additional materials aimed at this group of students.

8. ACKNOWLEDGMENTS

This work is supported by NSF CNS grant 1339104. In addition to NSF, we thank many people and organizations who contributed to this project. At TPT, our co-PIs Richard Hudson and Kathleen O'Donnell provided leadership and vision, Marie Domingo and Emily Stevens managed production and logistics, and Angie Prindle handled story design and field production. Traffic engineers Nickolas Van Gunst from the City of Minneapolis and JoNette Kuhnau from Kimley-Horn and Associates provided considerable insight and domain knowledge. Evelyn Beyer of Trafficware provided support on understanding professional traffic management software. Finally, PBS LearningMedia is hosting the website offering EMACS materials for download.

9. REFERENCES

[1] STEM-C Partnerships: Computing education for the 21st century (STEM-CP: CE21). Program Solicitation NSF 14-523. Accessed: 2014-08-25.

[2] *Highway Capacity Manual (HCM)*. TRB Publications, fifth edition, 2010.

[3] Investing in infrastructure: A question of trust. *The Economist*, May 12, 2012.

[4] C. Alvarado. Member Spotlight: Robert Panoff. *ACM SIGCSE Bulletin*, 46(1):6–7, January 2014.

[5] A. Bandura. Self-efficacy. volume 4 of *Encyclopedia of Human Behavior*, pages 71–81, 1994.

[6] E. Lotterman. Campaigns ignore infrastructure, public good. *St. Paul Pioneer Press*, Sept 8, 2012.

[7] D. Riley, G. Nellen, R. Barrera, and J. Quevedo. Crowdsourcing traffic simulation to improve signal timing. *J. Comp. Sci. Coll.*, 29(5):112–118, May 2014.

[8] The College Board. AP Computer Science Principles draft curriculum framework. June 2014.

[9] U. Wilensky. Netlogo. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL, 1999. `http://ccl.northwestern.edu/netlogo/`.

[10] U. Wilensky. Netlogo Traffic Grid model. Center for Connected Learning and Computer-Based Modeling, Northwestern U., Evanston, IL, 2003. `http://ccl.northwestern.edu/netlogo/models/TrafficGrid`.

[11] A. L. Zeldin, S. L. Britner, and F. Pajares. A comparative study of the self-efficacy beliefs of successful men and women in mathematics, science, and technology careers. *Journal of Research in Science Teaching*, 45(9):1036–1058, 2008.

[12] A. L. Zeldin and F. Pajares. Against the odds: Self-efficacy beliefs of women in mathematical, scientific, and technological careers. *American Educational Research Journal*, 37(1):215–246, 2000.