Contents

This material is based on work supported by the National Science Foundation under DRK-12 research, Award Number 1418199, *Changing Culture in Robotics Classrooms. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation or any collaborator or partner named herein.*

1 CHANGING CULTURE IN ROBOTICS CLASSROOMS?

1.1 INTRODUCTION

Computer Science (CS) plays a key role in all innovation, including advancements across all science, technology, engineering and mathematics (STEM) fields, *yet nationally only 8 percent of schools offer Advanced Placement (AP) CS courses.* There are 26,407 public secondary schools and 10,693 private secondary schools in the United States[i], yet only 3,075 schools are accredited to teach AP CS.[ii] In today's modern economy the ability to think computationally is an absolute necessity, yet access to courses that engage students in Computational Thinking Practices[iii] (CTP) are limited and wide disparities in how these courses are staffed and taught exist between schools.[iv]

This paper reports on the progress of Carnegie Mellon's Changing Culture in Robotics Classrooms (CCRC) project: NSF 1418199. CCRC is a research and development project that designs robotic education tools to foreground and teach Computer Science Principles (CSP) identified as important in NSF's Computer Science Principles Project[v] by engaging students in CTP using CTP assessments identified by SRI International (formerly Stanford Research Institute).[vi] *The goal of the project is to move robotics classroom and competition activities from a mechanistic focus to a CTP focus where students are learning the types of CS competencies taught in the new AP CSP course*[vii]. This paper describes: the team, the technologies the team is using in their experiment, a description of how the team is using Model Eliciting Activities[viii] to motivate CS-STEM learning in robotics classrooms, and future R&D goals of the project.

1.2 WHY STUDY ROBOTICS?

Robotics serves education in many ways. The everyday relevance of robotic systems – smart phones, autonomous cars, Internet-connected appliances, telemedicine, and countless other applications – provides a natural hook for lessons. Given the disruptive effects automation has had on jobs and the economy[ix], robotics itself is also a valuable area of study. The process of developing robotic solutions provides a rich and meaningful context for engaging students in CT practices and CS content, including work-related 21st century skills[x]. Robotics scenarios can also be used to contextualize other STEM concepts.[xi] Ultimately, the "brains" of robotic systems are driven not only by the bits of steel, silicon, and data within them, but by the CS and Computational Thinking [xii] skills of the developers who design and program them – higher-order problem-solving skills that might be best developed when conducting authentic inquiry.[xiii]

2 KEY PARTNERS

2.1 THE TEAM

The project partners include Carnegie Mellon's Robotics Academy (CMU), the University of Pittsburgh's Learning Research and Development Center's (PITT), Robomatter Inc., and the Robotics Education and Competition Foundation (RECF); in this paper this group is referred to as "the Team". CMU develops CS-STEM training materials and makes iterative improvements to the materials based on testing. PITT is the project evaluator, they design the evaluation tools (pre and posttests), observe classroom implementations, conduct surveys, and work collaboratively with CMU to iteratively improve the educational tools. Robomatter is an educational solutions company that develops curriculum continuums, interactive programming tools such as Robot Virtual Worlds (RVW), and related software to help bring the curriculum concepts to life. The RECF promotes and facilitates robotics competitions to over 15,000 teams per year. The Team recruits students from local school districts and RECF competitions to participate in the CCRC research project.

2.2 CMU and Pitt's Observations About Robotics Competitions & Classrooms

According to robot competition sponsors there are over 30,000 middle and high school robotics teams in the United States.[xiv] Many robotics courses at schools are established as a direct result of the school's participation in a robotics competition. The classes themselves are frequently used to prepare for the competition. Teachers typically provide students with various forms of existing robotics curricular units and practice competition activities, with varying degrees of focus on programming skills. Students develop some basic programming skills (e.g., basic movement control commands and simple sensor-triggered behaviors), but there is little reason to believe that they proceed at any higher levels, especially given that the competition challenges generally neither require nor reward more sophisticated programming. Furthermore, the Team has observed that CS problems in robotics classes are solved by individuals rather than collaborative groups which is contrary to CSP practices. Most programming involves "one off" solutions with no attention paid to generalizable or algorithmic solutions, thus missing activities that involve iterative design, refinement, and reflection processes that are central to creativity as well as computational thinking.[xv] CCRC's goal is to prove that when using the correct pedagogy, teachers can significantly increase the level of CS concepts taught via robotics competitions and classrooms.

3 Key Technologies

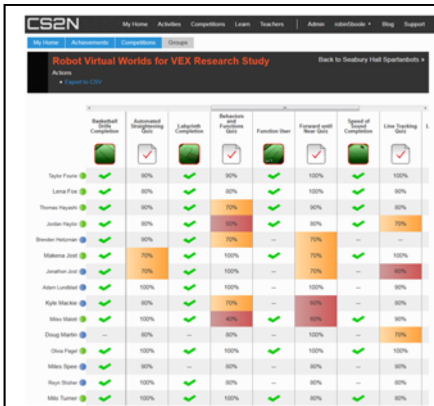
3.1 Robot Virtual Worlds (RVW)

RVWs are programmable simulation environments that the Team uses to engage students in CTP. RVW technologies provide students with access to ideal robots (i.e. perfectly working components such as motors, encoders, and sensors), advanced technologies not available in educational robotics (i.e. localized GPS with student defined waypoints, perfectly working

Left: Ruins of Atlantis programming game enables students to program submersibles. Middle: Simulations of RECF-promoted competitions. Right: Search and Rescue Challenge in which students build and test code with their virtual robot that also works on the physical robot.

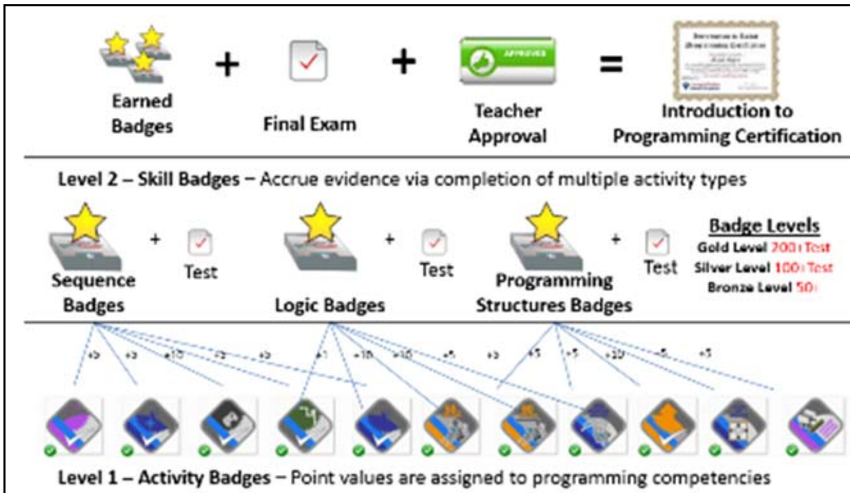
compass and gyro sensors, multi-robot communications), and access to otherwise impractical environments – underwater, in deep space, and driving vehicle-scale robots on city streets. The same technology can also be used to simulate existing educational competition environments, allowing students to build code before their physical robots are complete, or to extend these scenarios, e.g. by adding moving walls, floors, and randomized starting conditions. RVW affords classrooms with engaging environments that scaffold instruction, provides every student with a programmable robot, and has proven to save significant time when teaching robot programming.[xvi]

3.2 The Computer Science STEM Network



Picture 4 – A CS2N Group page

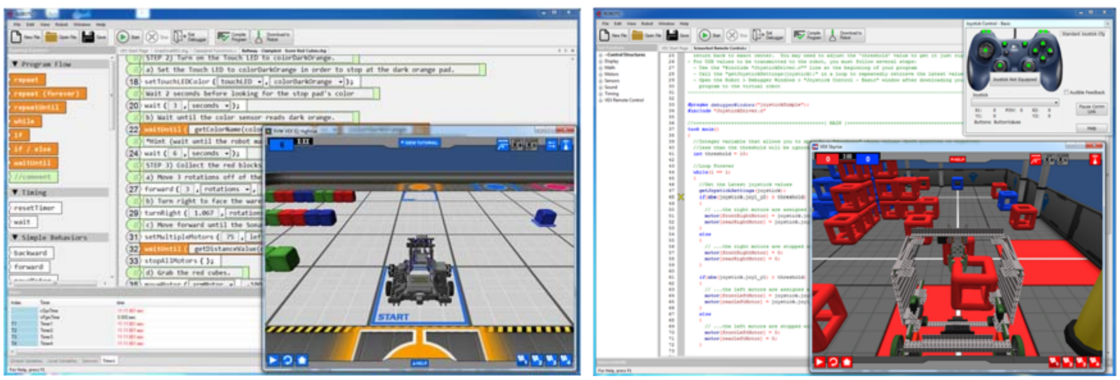
CMU and PITT designed an automated assessment and award system called the Computer Science STEM Network, (CS2N).[xvii] CS2N (www.cs2n.org) is the data aggregator for the project. CS2N uses cloud-based servers to deliver curricula, competitions, teacher training, team resources, and certifications. CS2N enables CMU and PITT to test and iteratively improve "mapped badged pathways" that lead to certifications. CMU and PITT are studying: the motivational and framing effects of badges and badging,[xviii] whether badges are accurate indicators of performance, and if the badges are easily understood by students, teachers, and other stakeholders. Additionally, the research team is interested in understanding the efficacy of CS2N automated assessment tools (badges, online tests, digital games,[xix] virtual competitions, and certifications) as formative and summative assessment tools (Picture 4). No one badged activity indicates that a student has mastered that concept; instead, accrual of multiple badges is evidence that the learner is moving towards mastery. To date, the system has awarded over 650,000 badges and certified over 4,000 students.



Picture 5 - An early prototype of a mapped badged pathway that leads to an Introduction to Programming Certification. In CS2N, RVW activities are assessed, badged, and given weighted values based on programming concepts like sequences, logic, and structures.

3.3 Scaffolded Software Solutions

The RVW software uses ROBOTC as its programming language and development environment. ROBOTC includes both standard text-based and graphical programming environments. ROBOTC Graphical (Picture 6) is designed for beginners and uses an intuitive drag and drop GUI to remove potential syntax errors, allowing the novice programmer to focus on programming logic. Text-based ROBOTC provides a simplified ANSI C programming environment that gives intermediate level programmers access to all of the tools that they need for more advanced programming with, e.g. variables, functions, and arrays. Beginning curriculum uses ROBOTC Graphical, transitioning to ROBOTC Text for intermediate and advanced topics.



Picture 6 – Left: ROBOTC Graphical. Graphical is designed for beginner programmers to learn basic program flow. Programming is done via user friendly set of drop down menus. Right: ROBOTC Text, based on ANSI C. Text is use in the intermediate and advanced lessons. Both pictures show the RVW simulation interface.

4 USING MODEL ELICITING ACTIVITIES TO TEACH COMPUTATIONAL THINKING

4.1.1 Model Eliciting Activities to Teach Computer Science Principles

Model Eliciting Activities (MEAs) have proven successful in creating rich classroom tasks for a diverse set of middle, high school, and college classrooms.[xx] Broadly construed, MEAs are a class of problems in which students must develop a "mental model" representing and incorporating key aspects of a given problem scenario in order to reason about it and produce a solution. The model is considered the product of the student work; it is not sufficient to find, e.g. the volume of water in a single oddly-shaped reservoir; students are required to produce a comprehensive (and comprehensible) general technique for finding the volume of ANY similarly-shaped reservoir. This framing shifts instructional emphasis to conceptual understanding and model-building rather than searching for the "right answer". Mental modeling is a critical component of mathematical thinking and learning[xxi] that has

Table 1. MEA Design Principle
Reality Principle – Can students can make sense of the problem based on prior experience?
Model Construction – Does the task need students to create a mental model of the solution?
Model Documentation – Will the response require students to explicitly reveal how they are thinking about the problem?
Self-Evaluation Does the statement of the problem strongly suggest criteria that enables students to judge when their response is complete?
Model Generalization Is the model not only good enough for the specific situation, but can be repurposed for other situations?
Simple Prototype Is the problem as simple as possible given the instructional goals?

also been shown to be critical to thinking and learning in science[xxii] and engineering. Groups of students solve MEA problems in teams over the course of several hours; the process typically requires them to express, test, and revise their models several times in order to solve the problem.

MEA scenarios are designed such that the process of developing the model leads students to engage with the desired learning content. A scenario in which students must keep track of free vs. occupied parking stalls in a lot, for example, would lead students to develop a system for representing free vs. occupied state of stalls using variables, as well as confront the need to organize those stalls according to some form of indexed data structure, e.g. an array. The task of assigning an arriving car to an unoccupied stall would require some form of search algorithm to find a stall, and the overall system of tracking vehicle comings and goings would require students to develop and implement their own algorithm to update the stall "occupied" state variables accordingly. The process of developing this solution would lean heavily upon students employing CT practices such as decomposition of the stall tracking problem into sub-problems, and developing algorithmic solutions that are generalizable to, e.g. garages with different numbers of stalls.

It would, of course, be completely infeasible to expect student groups to complete all these tasks independently of their own accord, without assistance. MEAs are sometimes said to "plow the field"[xxiii] by establishing the need for certain skills, techniques, and understandings in students' minds, allowing the seeds of more formal instruction to take root. MEA-based curriculum must also connect to (or provide) directed instructional resources on key topics for students to leverage as part of the learning and problem-solving processes. This implicit disposition toward independently seeking information as part of the problem-solving process (a form of self-regulated learning) can be considered a valuable outcome on its own.

It is important to note, however, that while there are clear principles that guide the development of effective MEA, there remains significant art to their design. For example, it is often difficult to come up with the right amount of information in the problem statement to enable immediate problem solving, without also removing opportunities for diverse solutions or decreasing conceptual difficulty. Sequencing activities after the introduction of the larger challenge can also be tricky. Consequently, developing a successful MEA will require empirical testing and refinement – what we present in this paper are ideas that we are in the process of testing.

4.1.2 Prior CMU/PITT Experience Developing MEAs

In the Robot Algebra project (NSF 1029342) CMU and PITT developed a series of effective MEA that used robotics activities to teach proportional reasoning; students showed significant mathematical gains after participating in the lessons.[xxiv] Prior research[xxv] found that students typically "guess and check" when they solve a robotics programming task. The Robot Algebra MEA forced students to construct mental models of their solution to the problem, test them, and present them to their peers. Additionally, students were required to develop generalized solutions that could be applied to an entire class of similar problems.

Picture 7 A set of robots with different diameter wheels that students are required to program to dance in synchrony in the RSD MEA.

Robot Synchronized Dancing (RSD)[xxvi] is one example of a Robot Algebra MEA. RSD targets proportional reasoning, a big idea in mathematics, as the learning goal. Proportional reasoning includes the mathematical concepts of scale, rate, and conversion of units; concepts that student need to know to be algebra ready. Proportional reasoning is the capstone of children's elementary school arithmetic, and the cornerstone of all that is to follow.[xxvii] In general, the nature of rolling robots lends themselves to interesting mathematical problems involving proportionality. The RSD Unit involved programming a set of robots with different size wheels (Picture 7) to dance in synchrony. All robots are required to travel the same distance at the same rate; this part of the challenge foregrounds direct proportional relationships. If the same set of robots have different length axles and the robot is required to turn the same number of degrees at the same speed, students will work with both direct and indirect proportional relationships. This MEA aligned with the design principles found in Table 1, the problem was easily understood by students, they had to develop a mental model of the problem, they had to document their solution to the problem, the challenge included rubrics that enabled self-evaluation (Table 2), the solution needed to be generalized so that it would work with any set of robots, and students were required to develop a simple prototype of their solution.

Performance Level	*Sharable or Reusable*	*Useful for this Specific Situation*	*Requires only Minor Editing*	*Requires Major Extensions or Refinements*	*Requires Redirection*
LETTER TO THE CLIENT Does the letter completely explain the solution, how/why it works, and how it could be adapted?	The letter provides enough detail for the client to implement the suggested solution, and it includes information about why it works, and how to alter the solution for different but similar circumstances.	The letter provides enough detail for the client to implement the suggested solution without additions or clarification.	The letter provides enough detail that the client could implement the procedure with only minor clarification.	The letter only describes the solution process generally. The client would be unable to implement the solution process simply from the information provided in the letter. The client would need clarification, more information, or help.	The letter describes very little of the solution process.

Table 2 One requirement of an MEA is that students document their solution. In the RSD MEA this requirement took the form of a letter to the client. The table above is the rubric students used to self-evaluate their work.

It is important to note that the development team had to iterate across a number of versions of the unit to find one that reliably improved mathematics skills. One version put too much emphasis on the task of designing an interesting dance rather than learning how to program it. A second version put too much emphasis on programing the dance for just one robot, so students did not attend sufficiently to the general mathematical patterns that are true of movement commands in general. The third version did not make enough use of the robotics context to add meaning to the mathematics patterns: students spent more time manipulating numerical patterns, but they did not spend much time thinking about why these mathematical patterns were meaningful. The fourth version got the balance just right: students spent most their time focusing on the mathematical patterns, but this time thinking about what features of the robots would lead to particular mathematical relationships (e.g., a multiplicative relationship rather than an additive relationship). Students in this version showed significant gains in general proportional reasoning, and the students were more likely to use the mathematics they learned in the MEA on another robotics problem.[xxviii] The current project integrates MEA into RVW and into the new CS Robotics Curriculum, and builds upon the Team's collective experience on developing MEA.

5 INTERMEDIATE AND ADVANCED CSP ROBOTICS UNITS

As noted in section 2.2 (Programming and Robotics Competitions), typical physical robotic competitions promote the development of engineering competencies and mechanics and not CS and CTP. This stands in stark contrast to the actual areas of robotics in which innovation is occurring today, such as machine vision and learning,[xxix] as well as those expected to be most valuable in the future.[xxx] Section 4 describes the tools and pedagogy that the Team is using in the Intermediate and Advanced Level CSP Robotics Units.

5.1 INTRODUCTION TO THE CSP ROBOTICS INTERMEDIATE LEVEL CURRICULUM

ASSUMPTIONS - The Intermediate Level (IL) curriculum assumes that students understand how to add comments to their code, how to decompose a problem through the creation of pseudocode, and have a basic understanding of motors, sensors, and the controller. Students are also expected to know how to download programs, how to develop mathematical solutions for basic robot movement, how conditional statements work, and behavior based program flow; these concepts were covered in the Getting Started curriculum.

BUILDING MODELS - Research shows that novice programmers tend to remain overly focused on superficial features of a program such as syntax and reserve words and not on the programming problem.[xxxi] Expert programmers, on the other hand, "chunk" code into larger conceptual blocks involving more generalized schematic representations for specific code functions.[xxxii] To succeed in the IL CS course, students will need to develop conceptual models of "what a computer is doing when it runs a program,[xxxiii]" rather than obsessing over the rules of the programming language. All students will have novice level experience building natural language programming solutions via pseudocode, but as problems become more complex research tells us that novice programmers become confused when their programming logic doesn't work.[xxxiv] This project's goal is to use conceptual modeling to help students develop strategies for solving computational problems. CCRC pedagogy emphasizes that students work collaboratively building a models of computational solutions before they begin coding. Students are encouraged to use abstraction to decompose a computational problem and break the problem into small parts. Once student teams build their conceptual models, they are required to present their models to the class. During class presentations, the class analyzes and optimizes the models before code development starts.

CODE DEVELOPMENT - Students learn intermediate level programming concepts like the use of variables, parameters, functions, and arrays. Now that the class has a conceptual understanding of the problem they are tasked to begin developing code. The decomposition process starts again and the conceptual model is broken into parts, patterns are identified, and programming assignments are given. Individual teams are assigned to develop solutions to small parts of the problem. When teams have working prototypes, they present their solution to the class. After all the small parts are solved, the entire class has access to all of the code and is challenged to stitch the code together. Throughout the process students are encouraged to share ideas and communicate their progress.

Intermediate level challenges require students to develop and explain how systems that collect, store, and use data work. The last assignment in each problem involves writing a letter to a potential client that explains how their solution works. This next section shows how the capstone activities in the new CSP Robotics Units will rely on use strategies found in MEA to teach CSP.

5.1.1 Intermediate Level Pipeline Explorer Challenge

Picture 8 shows example pipes along with a small robot designed to go inside a pipe.

The Pipeline Explorer challenge begins by introducing students to a hypothetical company named "Exploration Robotics International". Students are challenged to develop a working prototype of a robotic system that can travel into a pipe, identify where there are leaks in the pipe, record the location of the leaks, record how much of the pipe that the robot inspected, and then return to the beginning of the pipe and tell the operator the location of the leaks and the distance the robot traveled into the pipe.

Well-designed MEAs immediately engage students in high level thinking, and provide productive access to students from a wide range of prior knowledge. Table 3 maps MEA design principles to the CTP that students will engage with in the Pipeline Explorer Challenge.

Table 3. Model Eliciting Activity Design Principles

MEA Design Principle	**Pipeline Challenge Alignment with MEA/CTP**
Reality Principle – Can students can make sense of the problem based on prior experience?	The physical shape of a pipe is well-known to students, as is the idea that a crack can occur within one.
Model Construction – Does the task need students to create a mental model of the solution?	Constructing a solution to this problem requires abstractly representing a distance between two key locations, i.e. "the distance from the pipe entrance to the crack", even though the value of that distance may change. This maps directly onto the CS concept of variables, and the CT concept of abstraction.
Model Documentation – Will the response require students to explicitly reveal how they are thinking about the problem?	Students will reveal their thinking at three key points: when communicating about the solution within and between their groups, when submitting their pseudocode or flowcharts to the instructor, and when implementing their strategy using variables in code.
Self-Evaluation Does the statement of the problem strongly suggest criteria that enables students to judge when their response is complete?	The challenge itself implicitly requires students to account for the variability in distances. Student teams are provided with a client driven rubric that enables them to determine what a successful solution should include. Teachers are prompted in teacher-support

	materials to ask students to communicate this information while facilitating the class.
Model Generalization Is the model not only good enough for the specific situation, but can be repurposed for other situations?	The location of the crack in the pipe is understood to be unknown from the problem context, and is randomized at run-time, so that students have to (and know they have to) develop solutions that handle variability.
Simple Prototype Is the problem as simple as possible given the instructional goals?	The challenge focuses attention on the required element (the distance to the crack in the pipe), the completion requirement of displaying this distance explicitly requires the distance to be modeled, and the problem contains very few distractors.

5.1.2 Intermediate Level Bar Code Scanner Challenge

In this lesson, students are working as a subcontractor for Container Systems Unlimited (CSU) to develop a container tracking system for a new state of the art container yard in Baltimore, Maryland. Container yards are huge and if a container is lost it can take days to find it. CSU is looking at variety of solutions, but CSU's CEO seams set on using a barcode tracking system. The team's job is to develop a working prototype of a container handling barcode storage and retrieval system. The client is looking for innovators and successful teams will present how they have optimized their solution during development.

In order to prepare students to solve the problem they are assigned to research how barcodes readers work (there are many resources readily available on the Internet) and then develop a flowchart presentation of how their container storage and retrieval barcode system will work. This lesson uses the same MEA format as the Pipeline lesson; students begin by developing, sharing, and optimizing computational models of their solution before they develop any code. As they develop their conceptual models they will recognize that they need more efficient ways to store variables which the Team hopes to use to prime the discussion around arrays. This project requires students to learn about: the binary numbering system, arrays, how to store and retrieve elements in an array, how to use Boolean variables when programming, how to

```
bool lineMarker[5];
int barcodeNumber;

char letters[27] =
{'_','a','b','c','d',
  'e','f','g','h','i',
  'j','k','l','m','n',
  'o','p','q','r','s',
  't','u','v','w','x',
  'y','z'};

int binaryConvert()
{
  int binarySum = 0;

  binarySum = (lineMarker[0] * 1);
  binarySum = binarySum + (lineMarker[1] * 2);
```

Picture 9 Shows a picture of a container yard. At the right is code that students are given to being the problem.

utilize a for-loop to efficiently store data in an array, how to use the counter variable that is part of the "for" loop to iterate through and store information sequentially in an array, how to convert the array of Boolean values into binary numbers, how to convert the Boolean number algorithm into alphanumeric characters, and how to display values onto the LCD screen.

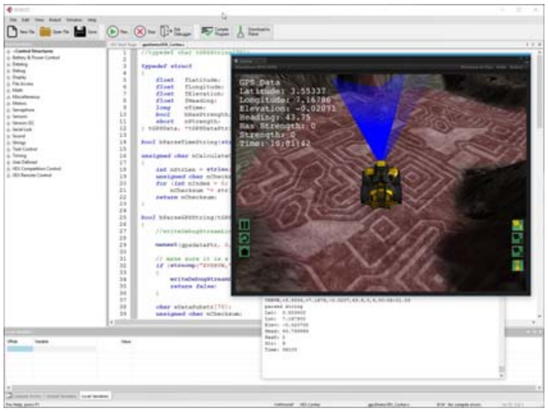
5.1.3 Intermediate Level Automated Garage Storage System Challenge

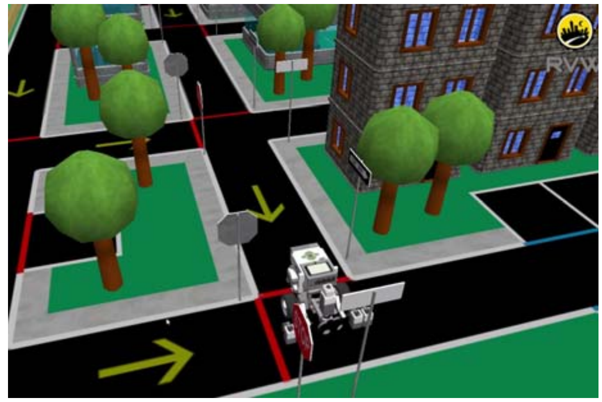
This problem begins with a client that needs a working prototype of an Automated Parking Garage. Students are required to develop a scale model working prototype of an Automated Garage Storage System. The prototype must be structurally sound and contain at least four parking spaces. The system must be able to identify how many cars and available parking spaces are currently in the garage, the location of each car, and use a color-coded system to access and deliver the car back to the owner. Successful solutions will require the use of at least two sensing systems (or input devices). An LCD screen will be used for communications between the automated system and the driver. The LCD screen will ask the driver to indicate whether the car is entering or the car is exiting. The driver needs to be able to scan a colored card to identify which parking space the driver wants to access. The robotic system should then move the car to the correct available space. While moving, an alert will sound to warn others of the automated parking system's movements. The miniature cars can be manually picked up and placed in spaces as needed during the prototype demonstration. This culminating challenge provides an iterative research and design problem that includes open-ended problem solving and opportunities for students to engage in CTPs and apply intermediate level programming concepts.

5.2 Current Technology & Future Curriculum Development – Advanced Level Curriculum

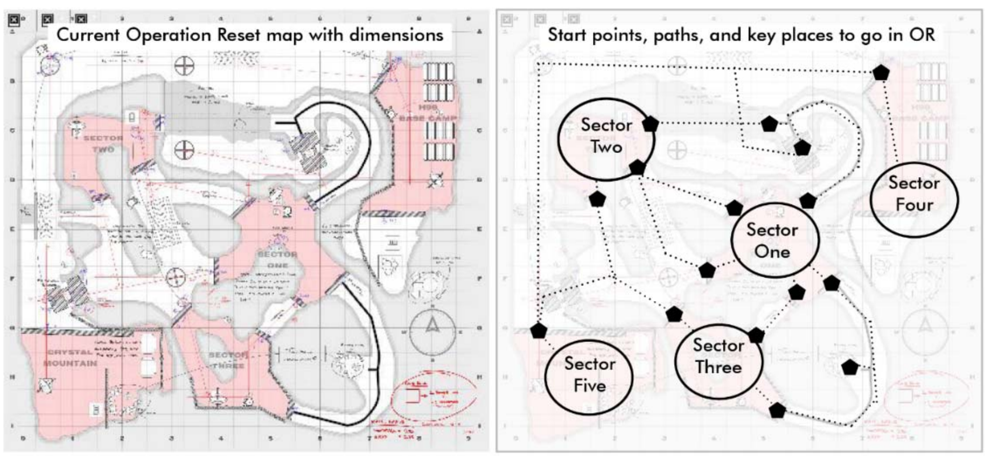
The goals of the Advanced Level Units are to teach multi-robot coordination and communications, variability, and advanced sensing techniques. The Team developed new technologies that provides students access to the types of sensing systems being used in today's real-world robotic systems. Picture 10 shows the integration of a Virtual Global Positioning System (V-GPS) into the RVW simulation environment. The V-GPS provides student access to the Longitude, Latitude, Elevation, Heading, and Speed of the robot, along with additional world parameters including Satellite Strength and World Time. With this data, rather than relying on a linear set of commands and a static world for robot locomotion, students can program their robots to move to a specific location within the virtual world, even if their starting position changes or the robot gets off course.

The Team also introduced environmental randomization into the RVW system (i.e. obstacles and conditions that randomly appear) promoting the development of more sophisticated algorithms by students. The Mini-Urban Challenge (Picture 11) and Operation Reset (Picture 12) are two example worlds that we are currently experimenting with. The combination of V-GPS sensor data with information from sensors such as the Gyro Sensor (an angular velocity sensor) enables students to know the robot's heading and location. If they also know the world's map, they can develop working prototypes for challenges like Search and Rescue or the travelling salesman problem.





Picture 10 on the left shows how the team integrated a programmable Virtual GPS sensor into the RVW system. Picture 11 shows on the right shows a version of a mini Urban Challenge that is available today for students to practice advanced programming concepts with.



Picture 12 Operation Reset World Map In this challenge students must develop a generalized algorithm that enables them to find the quickest route when one of the paths is blocked.

6 IN SUMMATION

The project Team's goal is to design a set of scaffolded problems that engage students in CTP that promote students ability to develop computational models of solutions. The pedagogy of the curricula is designed to encourage student communications and collaboration and place students in projects where they "figure things out together".[xxxv] The Team is developing and testing rubrics that reward the development, documentation, and evaluation of algorithmic models. In sum, the Team is using strategies that have proven to be successful in engineering and mathematics education[xxxvi] to scaffold CS in robotics classrooms. MEA's target model construction, documentation, communications, and collaboration skills, these activities align with the Team's curricular goals of teaching CSP[xxxvii] and foregrounding CTP in robotics classrooms.

7 REFERENCES

[i] Archived: High school facts at a glance – US Department of Education, http://www2.ed.gov/about/offices/list/ovae/pi/hs/hsfacts.html

[ii] College Board, AP Course Audit (Computer Science and Calculus, 2013-2014, United States; accessed April 27, 2015), https://apcourseaudit.epiconline.org/ledger/search.php; http://www2.itif.org/2013-guestworkers-high-skill-labor-market.pdf

[iii] International Society for Technology in Education (ISTE) and the Computer Science Teachers Association (CSTA). This material is based upon work supported by the National Science Foundation under Grant No. CNS-1030054

[iv] "CSTA-Oracle Academy 2014 U.S. High School CS Survey: The State of Computer Science in U.S. High Schools: an Administrator's Perspective," Computer Science Teachers Association,/ http://csta.acm.org/Research/sub/Projects/OracleSurvey_DataSummary.pdf .

[v] Astrachan, O., Briggs, A., Diaz, L., (2009-13) CS Principles, http://www.csprinciples.org/ *NSF Special Projects O938336*

[vi] Bienkowski, M., Snow, E., Rutstein, D. W., & Grover, S. (2015). *Assessment design patterns for computational thinking practices in secondary computer science: A first look (SRI technical report).* Menlo Park, CA: SRI International.

[vii] College Board, 2016, *AP Computer Science Principles Course and Exam Description Effective Fall 2016*, https://secure-media.collegeboard.org/digitalServices/pdf/ap/ap-computer-science-principles-course-and-exam-description.pdf

[viii] Lesh, R., Hoover, M., Hole, B., Kelly, A., & Post, T. (2000). Principles for developing thought-revealing activities for students and teachers. In A. E. Kelly & R. Lesh (Eds.), *Handbook of Research Design in Mathematics and Science Education* (pp. 591-646). Mahwah, NJ: Lawrence Erlbaum Associates.
Reid, K., & Floyd, C. (2007). *Details of Implementation of an Interdisciplinary Pre-Engineering Activity in a Middle School Curriculum*. Paper presented at the Frontiers In Engineering.

[ix] Manyika, James, et al. *Disruptive technologies: Advances that will transform life, business, and the global economy*. Vol. 12. San Francisco, CA: McKinsey Global Institute, 2013.

[x] Ravitz, J., Hixson, N., English, M., & Mergendoller, J. (2012, April). Using project based learning to teach 21st century skills: Findings from a statewide initiative. In *Annual Meetings of the American Educational Research Association. Vancouver, BC. Retrieved from http://www. bie. org/research/study/PBL_21CS_WV*.

[xi] Silk, E. M., Higashi, R., Shoop, R., & Schunn, C. D. (2009). Designing Technology Activities that Teach Mathematics. *Technology Teacher*, *69*(4), 21–27. Retrieved from http://search.ebscohost.com/login.aspx?direct=true&db=eric&AN=EJ867950&site=ehost-live\nhttp://www.iteaconnect.org/Publications/ttt.htm\nhttp://proxy.library.umkc.edu/login?url=http://search.proquest.com/docview/61821086?accountid=14589\nhttp://www.iteaconnect.org/Publications/ttt.htm LA - English

[xii] Guzdial, Mark, and Daniel Reed. "Securing the future of computer science; reconsidering analog computing." *Communications of the ACM* 56.4 (2013): 12-13.
Wing, Jeannette M. "Computational thinking," *CACM Viewpoint,* March (2006): 33-35.
Lazer, D., Pentland, A. S., Adamic, L., Aral, S., Barabasi, A. L., Brewer, D., & Van Alstyne, M. (2009). Life in the network: the coming age of computational social science. *Science (New York, NY)*, *323*(5915), 721.

[xiii] Kuhn, D. (2007). Is direct instruction an answer to the right question?. *Educational psychologist*, *42*(2), 109-113.

[xiv] FIRST. (2013). http://www.usfirst.org/aboutus/first-at-a-glance reports that they have 32,650 teams with 350,000 students participating and over 120,000 adult volunteers.
BEST Robotics Annual Report – 2012 Report states that they had over 800 teams all from schools http://www.roboticseducation.org/wp-content/uploads/2013/05/RECF_AnnualReport2013.pdf
VEX Robotics 2013 Annual Report – http://www.roboticseducation.org/wp-content/uploads/2013/05/RECF_AnnualReport2013.pdf - Reports that they have 7,336 teams compete in the 2012-13 VEX Robotics Game.

[xv] Resnick, M. (2007, June). All I really need to know (about creative thinking) I learned (by studying how children learn) in kindergarten. In *Proceedings of the 6th ACM SIGCHI conference on Creativity & cognition* (pp. 1-6). ACM.
[xvi] Liu, A., Schunn, C. D., Flot, J., & Shoop, R. (2013c). The role of physicality in rich programming environments. *Computer Science Education*, 23(4), 315-331. 10.1080/08993408.2013.847165
[xvii] CS-STEM (2011) Computer Science Student Network Badge System, Digital Media Learning Competition. http://dmlcompetition.net/Competition/4/badges-projects.php?id=2607
[xviii]Abramovich, S., Schunn, C.D., Higashi, R. (2013) *Are Badges Useful in Education? It Depends Upon the Type of Badge and the Expertise of the Learner.* Educational Technology Research & Development, March 2013 DOI: 10.1007/s11423-013-9289-2.
Higashi, R., Abramovich, S., Shoop, R., Schunn, C.D. (2012, June) *The Roles of Badges in the Computer Science Student Network.* 2012 GLS Conference
[xix]Alfieri, L., Higashi, R., Shoop, R., Schunn, C.D. (2015, February) *Case Studies of Robot-Based Game to Shape Interests and Hone Proportional Reasoning Skills*. International Journal of STEM Education.
[xx] Lesh, R., Hoover, M., Hole, B., Kelly, A., & Post, T. (2000*). Principles for developing thought-revealing activities for students and teachers*. In A. E. Kelly & R. Lesh (Eds.), Handbook of Research Design in Mathematics and Science Education (pp. 591-646). Mahwah, NJ: Lawrence Erlbaum Associates.
Reid, K., & Floyd, C. (2007). *Details of Implementation of an Interdisciplinary Pre-Engineering Activity in a Middle School Curriculum*. Paper presented at the Frontiers In Engineering.
[xxi] Lehrer, R., Schauble, L., Carpenter, S., & Penner, D. E. (2000). The interrelated development of inscriptions and conceptual understanding. In P. Cobb (Ed.), *Symbolizing and communicating in mathematics classrooms: Perspectives on discourse, tools, and instructional design* (pp. 325-360). Mahwah, NJ: Lawrence Erlbaum Associates.
[xxii] Schwartz, D. L., Martin, T., & Pfaffman, J. (2005). How mathematics propels the development of physical knowledge. *Journal of Cognition and Development, 6*, 65-88.
Schwarz, C. V., Reiser, B. J., Davis, E. A., Kenyon, L., Acher, A., Fortus, D., et al. (2009). Developing a Learning Progression for Scientific Modeling: Making Scientific Modeling Accessible and Meaningful for Learners. *Journal of Research in Science Teaching, 46*(6), 632-654.
[xxiii] How to Teach with Model Eliciting Activities, Pedagogy in Action, http://serc.carleton.edu/sp/library/mea/how.html
[xxiv] Stein, M. K., Boston, M., & Kessler, A. (under review). Conceptualizing Teacher's Practices in Supporting Students' Mathematical Learning in Computer-Directed Learning Environments. Paper submitted to the International Conference of the Learning Sciences.
[xxv] Silk, E. M. (2010, August 25). Can math help in LEGO robotics competitions? [4-partweb logpost]. Retrieved fromhttp://robotics-academy.org/blog/?p=356[Part 1] [Part 2] [Part 3] [Part 4]
[xxvi] Silk, E. M., Higashi, R., Shoop, R., & Schunn, C. D. (2010). Designing technology activities that teach mathematics. *The Technology Teacher, 69*(4), 21-27.
[xxvii] Lesh, R., Post, T., & Behr, M. (1988). Proportional reasoning. In J. Hiebert & M. Behr (Eds.), *Number Concepts and Operations in the Middle Grades* (Vol. 2, pp. 93-118). Reston, VA: National Council of Teachers of Mathematics.
[xxviii] Schunn, C. D., & Silk, E. M. (2011, April). Resources for learning robots: Facilitating the incorporation of mathematical models in students' engineering design strategies. Paper to be presented at the Annual Meeting of the American Educational Research Association, New Orleans, LA, USA.
[xxix] Guizzo, E., & Deyle, T. (2012). Robotics Trends for 2012. *IEEE Robotics & Automation Magazine*, *19*(1), 119-123.
[xxx] Robotics Virtual Organization. (2013). *A Roadmap for U.S. Robotics—From Internet to Robotics (2013)*. Retrieved from http://robotics-vo.us/sites/default/files/2013%20Robotics%20Roadmap-rs.pdf
[xxxi] Kurland, D. M., Pea, R., Clement, C., Mawby, R. (1986) *A study of the development of programming ability and thinking skills in high school students,* Journal of Educational Computing Research, 2 (4) (1986), pp. 429–458
Catrambone, R. (1998). The subgoal learning model: Creating better examples so that students can solve novel problems. Journal of Experimental Psychology: General, 127(4), 355-376.

[xxxii] Robins, A., Rountree, J., & Rountree, N. (2003). Learning and Teaching Programming: A Review and Discussion. Computer Science Education, 13(2), 137–172. http://doi.org/10.1076/csed.13.2.137.14200
[xxxiii] Du Boulay, B., O'Shea, T., Monk, J., (1981) *The black box inside the glass box: presenting computing concepts to novices*. International Journal of Man-Machine Studies, 14(3):237–249
[xxxiv] Robins, A., Rountree, J., & Rountree, N. (2003). Learning and Teaching Programming: A Review and Discussion. *Computer Science Education*, *13*(2), 137–172. http://doi.org/10.1076/csed.13.2.137.14200
Ambrosio, A. P., Almeida, L. S., Macedo, J., & Franco, A. (2014). Exploring core cognitive skills of Computational Thinking. *Psychology of Programming Interest Group Annual Conference 2014 (PPIG 2014)*, 25–34. Retrieved from http://repositorium.sdum.uminho.pt/bitstream/1822/30076/4/PPIGproceedings.pdf
[xxxv] 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.
[xxxvi] Lesh, R., Hoover, M., Hole, B., Kelly, A., & Post, T. (2000). Principles for developing thought-revealing activities for students and teachers. In A. E. Kelly & R. Lesh (Eds.), *Handbook of Research Design in Mathematics and Science Education* (pp. 591-646). Mahwah, NJ: Lawrence Erlbaum Associates.
[xxxvii] Astrachan, O., Briggs, A., Diaz, L., (2009-13) CS Principles, http://www.csprinciples.org/ *NSF Special Projects O938336*
Bienkowski, M., Snow, E., Rutstein, D. W., & Grover, S. (2015). *Assessment design patterns for computational thinking practices in secondary computer science: A first look (SRI technical report).* Menlo Park, CA: SRI International.