Pre-college Students' Use of Systems Engineering Methods in Design

Aaron W. Johnson*, Sara Willner-Giwerc*, Paul T. Grogan†, and Ethan E. Danahy*

*Center for Engineering Education and Outreach, Tufts University, Medford, MA

{aaron.johnson, sara.willner_giwerc, ethan.danahy}@tufts.edu

†School of Systems and Enterprises, Stevens Institute of Technology, Hoboken, NJ

pgrogan@stevens.edu

***Abstract*—Systems engineering activities provide an opportunity for pre-college students to engage in sophisticated engineering design practices beyond commonly-accepted limits of their capability. This research qualitatively investigates how pre-college students engaged in a systems engineering activity using LEGO EV3 robotics with three learning objectives: 1) a system can be decomposed into subsystems, 2) designers coordinate activities by communicating requirements and interfaces, and 3) testing reveals problems and changes propagate to other components. The analysis was based on methods of qualitative content analysis and employed video data produced by students to describe their robot designs as well as video data collected by the researchers. While students did not use the language of systems engineering, examples show students productively engaging in systems engineering methods that relate to each learning objective. Results suggest pre-college students are capable of understanding and implementing basic systems engineering concepts representing a more sophisticated approach to engineering design.**

Keywords—discourse analysis; P-12; problem based learning; systems thinking

I. Introduction

Systems engineering is a structured process to realize complex products in large organizations through coordinated efforts across multiple disciplines [1]. Core systems engineering activities manage interdependencies between entities to define requirements, decompose a system into its components, design and implement the components, and recompose components into the integrated system while verifying and validating requirements. The International Council on System Engineering's vision for systems engineering in 2025 highlights a need to educate and train the future workforce with rigorous and pragmatic methods and tools built on foundational elements of mathematics and information science, systems science and theory, social and human sciences, and natural science [2]. These topics contribute to the broad set of skills in effective systems engineers including core math, science, and engineering fundamentals; mastery of systems engineering practices; systems thinking; interpersonal skills; and technical leadership [3].

Motivation for systems education aligns with recent discussion of integrated approaches to pre-college science, technology, engineering, and math education [4]. While students must establish competency in specific disciplines, systems engineering activities require a representational fluency (e.g. [5]) to transition between domains. Early education focusing on systems thinking trains individuals to understand the broader impacts of a decision on others and, correspondingly, how others' decisions affect and influence their own. Developing these skills requires a more abstract thought process similar to chunking in cognitive science [6]. Thinking about an engineering problem in terms of systems engineering concepts—such as requirements, functional decomposition, and integration—is a more sophisticated approach to design beyond what some research claims pre-college students are capable of as beginning designers [7].

An understanding of systems engineering promotes broad-based technological literacy and enables all students—not just those who become future systems engineers—to understand how the technological, human, and natural components of a system affect the others in both positive and negative ways. The International Technology Educators Association's Standards for Technological Literacy specifically identify systems engineering as a focus for grades K-12 [8]. For example, students in grades 6-8 should learn that "systems thinking involves considering how every part relates to others," and "requirements are the parameters placed on the development of a product or system" (p. 39).

Despite the potential benefits of teaching elements of systems engineering at the pre-college level, systems engineering education has traditionally been focused on master's degrees and continuing education [9]. Some organizations have developed systems engineering-oriented educational programs for pre-college students and teachers [10,11], but assessment of learning outcomes has been varied. Jain et al. quantitatively assessed students' learning through pre- and post-activity multiple-choice tests [12,13]. While the authors were able to show statistically significant increases in learning, this assessment method is unable to show how students think about systems engineering concepts during the course of the activity. Bartus and Fisher qualitatively assessed how teachers engaged in systems engineering methods like systems integration and verification and validation in a wind farm design activity [11].

This paper fills the research gap by providing a qualitative analysis of middle school students engaged in a systems engineering activity using LEGO EV3 robotics. The activity is

This research was funded by the CEEO Innovates program and Dr. E's Lab at Tufts University.

designed as an introduction to systems engineering concepts emphasizing the role of interdependencies between subsystems in design. To this end, the workshop features three distinct learning objectives [14]:

1. *A system can be decomposed into subsystems.* This learning objective encourages students to view an entity as a set of interrelated components. Through the process of decomposing a system into its constituent subsystems, students also assign functional attributes to alternate physical forms.

2. *Designers coordinate activities by communicating requirements and interfaces.* This learning objective introduces students to two key concepts in systems engineering: requirements and interfaces. Requirements dictate the design of lower-level subsystems so that systems objectives are achieved and enforce constraints imposed by architectural decisions. Interfaces, a specific type of requirement, define how subsystems connect to each other with physical links (in the case of this activity).

3. *Testing reveals problems and changes propagate to other components.* This learning objective emphasizes that systems design often requires an iterative process. It highlights how testing can uncover incorrect or incomplete requirements that cause problems in systems design, and how changes to one subsystem have a tendency to propagate to other subsystems.

In this paper, we analyze student discourse in video data captured during the three workshops to answer our research question:

What elements of systems engineering methods do pre-college students use in design?

Specifically, we looked for evidence of systems engineering methods that related to our three learning objectives: 1) abstraction methods (decomposition and integration), 2) coordination mechanisms (requirements and interfaces), and 3) design processes (testing and iterative design).

II. Study Context

We held three nearly-identical systems engineering workshops in January and April 2016. These workshops were the result of seven months of development and iteration, which is described in detail in [14]. Each two-hour workshop enrolled between nine and eleven students in grades 5-9, with most in middle school (grades 6-8). Thirty students participated in the study across all workshops—eleven females and nineteen males. Our recruitment e-mail noted the systems engineering focus of the workshop and requested participants to have experience with LEGO robotics (either the NXT or EV3 sets). While we assume all students had some experience with LEGO robotics, we did not test students' abilities at the workshop and assume that their experience levels likely varied.

After brief introductions, the workshop began with a thirty-minute interactive presentation based on the NASA Systems Engineering Handbook [15]. Using the NASA Space Transportation System (i.e. the Space Shuttle) as an overarching example, this presentation defined systems engineering, discussed the cycle of decomposition-implementation-integration, and gave examples of functional and interface requirements. This content of the presentation supported our three learning objectives but did not explicitly outline them. After this presentation, students had approximately ninety minutes for the activity: building, testing, and iterating upon a "toxic waste disposal" robot.

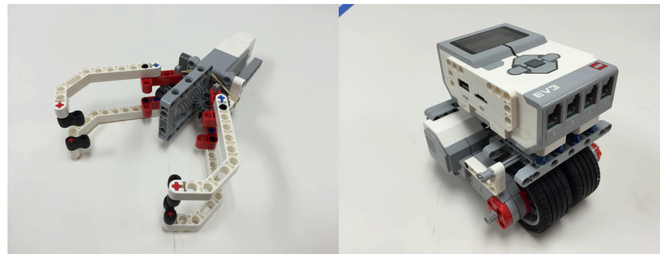

Fig. 1. Students are provided with a pre-built arm with a claw manipulator at the end (left), and a pre-built and pre-programmed chassis (right).

Because our systems engineering learning objectives focus on students' abilities to integrate and redesign subsystems and not to construct a robot from the ground up, in our activity we provide students with two pre-built subsystems: an arm with a claw manipulator at the end, and a chassis (Figure 1). These subsystems are designed to be functional as presented to students but sub-optimal in their operation [16]. The robot is pre-programmed to travel a defined distance, grab a soda can (representing the "toxic waste"), and move the can into a boxed area representing a "safe disposal container" (Figure 2). Students were divided into groups of one or two and asked to use the remaining parts from a LEGO EV3 kit to modify the individual subsystems as they saw fit and to integrate them.

This activity features two separate parts. In the first part, the robot travels 36 inches to the platform, picks up an empty soda can off the front of the 3-inch tall platform, travels backwards to the start line, and releases the can. We encouraged students to check partially-constructed robots against the test setup to gather feedback (referred to as a "check" in this paper), but challenged students to successfully complete the task during the first dynamic test (i.e. running the robot's program) to encourage thoughtful design instead of

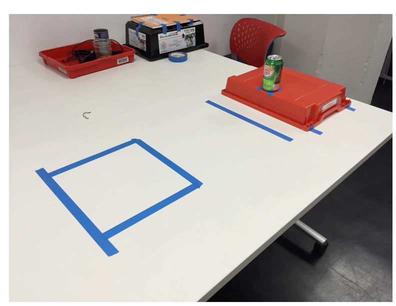

Fig. 2. The test setup. Robots start at the left of the test setup, retrieve the can from the platform on the right, and deposit the can in the box on the left. The lines to the left of the taped box and platform serve as guides to help students position their robot.

constant guess-and-checking.

The second part of this activity consists of two more-difficult challenges requiring students to modify their robot subsystems and/or interface. First, we place the empty soda can on a higher platform (6.5 inches) and further back from the edge (7.5 inches). The robot employs the same programming for this challenge. Once students complete this challenge, we increase the weight of the soda can by filling it with pennies to create a larger tipping moment on the robot and require students to further modify their robot. In the first and second workshops, students went from the empty can to a can with 35 or 25 pennies, respectively. While several groups successfully completed this challenges, none were successful on the first try. In the third workshop, we added only 10 pennies to start, and added pennies in increments of 5 as groups were successful with the prior weight. The incremental addition of weight gave students a better idea of how their robot performed, rather than a binary assessment of success or failure with 35 pennies.

Each group was given an iPad to document their building process and the factors that they considered while making their robot. Students in the first workshop, for example, received the following instructions from the first author:

> "What we want you guys to do is– when we're done– essentially I want you guys to take video of your robot and tell us a little bit about why you're building. So say, like, 'I'm putting the arm, you know, in this position for this reason,' or, 'I'm connecting them together this way for this reason.' Just tell us a little bit about why you're making the decisions that you're making."

Students in the first and third workshops performed this documentation by making short videos, and students in the second workshop made a comic strip with the LEGO StoryVisualizer app. These videos and comic strips were intended to capture the students' thinking during the activity and provide insight into what factors (i.e. requirements) influenced the design of their robot. To supplement this student-produced documentation, we also captured video of the entire workshop room, the table where students tested their robots, and carried a video camera around to record student conversations.

III. Data Analysis

The primary data for our analysis were the student-produced videos from the first and third workshops, which gave us insight into their thinking about the activity. We also reviewed the researcher-produced videos to give a complete record of each group's activities during the workshop, and to place their self-produced videos in the larger context. The second group's LEGO StoryVisualizer comic strips were excluded from the analysis because this documentation format required more effort from the students, and as a result their discussions of the robot designs were briefer and less descriptive. In total, we reviewed data from nine dyads (five females and thirteen males) and one female student working alone.

Our analysis followed a systematic and iterative process based on methods of qualitative content analysis [17] and open and axial coding [18] in order to identify examples of how students engaged systems engineering methods that correlated to our learning objectives: 1) abstraction methods (decomposition and integration), 2) coordination mechanisms (requirements and interfaces), and 3) design processes (testing and iterative design). To begin our analysis, the first and second authors independently reviewed the video data and noted examples of student discourse—both words and gestures—that involved one of these three systems engineering methods. After this independent coding was complete, we compared notes and discussed any examples with disagreement on the intent of students' discourse. Next, we re-analyzed the videos to draw out similarities and differences in the ways that students' discourse showed evidence of the three systems engineering methods. We then combined the examples for each method together into a "flight" of data [19]. We reviewed and discussed the flights in order to select the best examples of each systems engineering method, which are presented in this paper. Finally, the third author, an expert on systems engineering who did not participate in these workshops or the data analysis, reviewed the selected examples to independently evaluate how they addressed the research question.

IV. Findings

Students rarely used the language of systems engineering discussed in the presentation at the beginning of the workshop. They did not specifically call out "decomposition," "requirements," or the "interface." However, we argue the videos show students discussing systems engineering methods in a more familiar language. In this section, we use examples from student groups to show how students engaged systems engineering methods that correlated to our learning objectives: 1) abstraction methods (decomposition and integration), 2) coordination mechanisms (requirements and interfaces), and 3) design processes (testing and iterative design).

A. *Abstraction Methods: Decomposition and Integration*

The first learning objective of our workshop encouraged students to approach their robot with a systems thinking mindset. We looked for evidence of students viewing the robot as having two distinct subsystems that operate together to complete the task, even after the arm and chassis were integrated. We also looked for evidence of students performing "functional decomposition" to break the overall task into smaller individual subtasks, assigning subtasks to specific parts of the robot, and considering alternative physical forms that still complete each subtasks.

It is difficult to tell from students' videos whether they considered the robot as one complete unit or two integrated subsystems. Students often discussed features of the claw or the chassis on their robot, but this does not give clear insight into their deeper thinking. Without specifically using the language of systems engineering and discussing subsystems and interfaces, we cannot be sure how students were thinking of their robot.

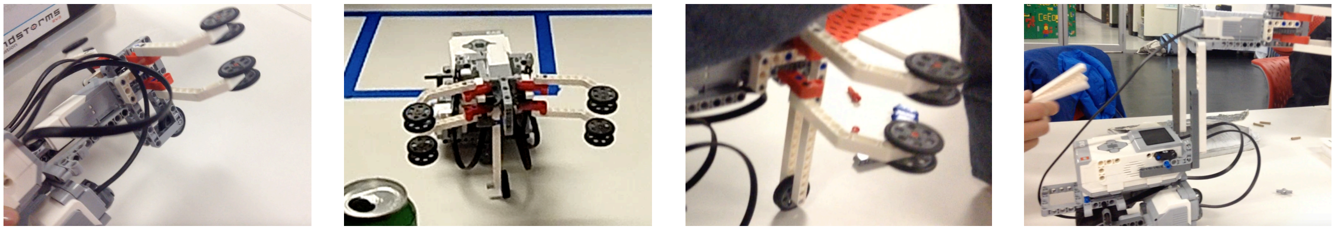

Fig. 3. Evolution of the design for Kylie and Andrei's arm support in the first part of the activity from a short rectangle in the first iteration (left) to a single long beam in the second iteration (center left) and, finally, to two long beams better connected to the arm in the third iteration (center right). In the second part of the activity they used the connector cable for the claw's motor to provide tension, supportin the arm (right).

The actions of one dyad, Kylie and Andrei,[1] suggest that they may have been considering the robot as integrated subsystems by the way they made incremental modifications to their robot that focused on one component at a time. Kylie and Andrei's first robot design had the arm connected to the front of the chassis (Figure 3, left). The design also featured a rolling wheel under the arm intended to provide, in their words, "more stability under here [the arm] because of all this weight." Kylie and Andrei made a video of their robot and then checked it against the test setup, seeing their arm would not clear the top of the platform. They then disconnected the arm from the chassis and built a new, taller interface. This taller interface also required them to make the support under the arm longer (Figure 3, center left). Kylie and Andrei tested this robot design, but found that the single-beam support wobbled and made the robot turn slightly to the left while traveling forward. After this failed test, they modified aspects of their robot's arm while leaving the chassis and interface alone. They added a second beam to the arm support and exchanged the wheel for one that would roll better (Figure 3, center right). They also saw other teams were having trouble with the arm's gearbox as the round gears would occasionally pop out of the worm gear when the claw closed around the can. In order to solve this problem, Kylie and Andrei added a beam across the gearbox to hold it together. Kylie and Andrei made a video explaining these modifications and, as soon as they were finished, Andrei disconnected the arm from the chassis and began re-building the interface. The modified interface connected the arm to the side of the chassis rather than the front. In their third video, Andrei explained they changed the interface because it "was loose and would move up and down." This robot design was ultimately successful at the first challenge. By making separate videos for the arm modifications and the interface modifications, Kylie and Andrei may be indicating that they considered the robot as two subsystems and an interface, and not as one integrated robot. However, this intent is not explicit, and is developed through our interpretation of the video data. Furthermore, we cannot know to what extent Kylie and Andrei's thinking was influenced by us giving them the arm and chassis as separate components at the beginning of the activity.

We argue that Kylie and Andrei's modification of their robot also shows evidence of functional decomposition and considering alternate physical forms that satisfy particular functions. As they modified their robot, each design had some component that supported the weight of the arm (Figure 3). However, this function was satisfied by multiple physical forms that evolved as Kylie and Andrei discovered new requirements. During the first challenge the support evolved from a short rectangle to a single long beam and, finally, to two long beams better connected to the arm. All of these also featured a rolling wheel in contact with the ground. When Kylie and Andrei moved to the second challenge, they supported the weight of the arm in an entirely different manner by wrapping the connector cable for the claw's motor around the chassis, providing tension to keep the taller interface from tipping over (Figure 3, right). The evolution of Kylie and Andrei's arm support suggests that they were able to perform a functional decomposition on their robot and consider each of the necessary functions, such as supporting the weight of the arm, individually.

B. *Coordination Methods: Requirements and Interfaces*

The second learning objective of our workshop encouraged students to consider requirements and interfaces, both of which are critical concepts to communicate in the development of large-scale systems. Instead of just addressing issues with their robot as they cropped up, or taking a purely guess-and-check approach to design, we looked for evidence of students assessing the challenge and developing a mental list of requirements their robot had to satisfy. These requirements could address the *constraints* of the test setup, pre-built subsystems, programming, or LEGO EV3 pieces; the *functions* the robot had to provide; or the *interface* the students had to construct to unite the arm and the chassis.

Students' videos discussed a wide variety of requirements that they considered in designing their robot. Table 1 shows the requirements that were mentioned across all videos, along with an example of the language that students used.

Students identified that their arm had to be a certain height above the table in order to clear the platform and a certain length in order to reach the can. Both of these requirements derived from the challenge given to students. Students also identified two functional requirements linked to deficiencies in the design of the pre-built arm: 1) the claw needed to securely grab the can but there was little friction between the provided claw and the can; and 2) the gears needed to stay meshed but could come apart when gripping the can.

[1] Note: all student names in this paper are pseudonyms chosen by the researchers.

The videos also show students across all three workshops discussed design features that made parts of the robot “sturdy,” “stable,” or “secure” (Table 2). Conversations with other engineering education researchers at Tufts University indicated students, particularly elementary students, sometimes use these words to express an unknown problem with a design. When students fail a test and they can’t figure out why, they add more pieces onto their design in an effort to make it more “stable.” To the contrary, we find evidence that the students in our workshops were considering stability in an effective and productive manner. They had an inherent sense of how much the arm should be able to move up and down, likely from their experience building LEGO robots, and they formed this into a requirement for their interface. The video data shows how students took this requirement, communicated it to their partner, and designed their interface to satisfy the requirement. We also see numerous examples where students—such as Kylie and Andrei, and Tim and Ollie—iterated on their interface to satisfy this requirement.

C. *Design Processes: Testing and Iterative Design*

The third learning objective of our workshop showed how incorrect or incomplete requirements can cause problems when a system is operated and how these issues can be discovered in iterative testing and re-design. We looked for students checking their robot against the test setup (without operating it), testing their robot (by running the program), and making changes to their design in response to the results. We

TABLE I. REQUIREMENTS DISCUSSED BY STUDENTS

Requirement	Example
Arm must be tall enough to clear the top of the platform.	*Student brought robot to test setup:* Researcher: “Do you think it’s going to work?” Thomas: “No” Researcher: “No, what do you think is not going to work?” Thomas: “Um, it’s not tall enough I think.”
Arm must be long enough to reach the can.	Will: "So now we have a new challenge where the soda can is raised up higher and further away. What we've done is we extended the reach by moving it this way [points to the left of chassis where claw is]. First it was about here… and now we've moved it up there extending the reach and making it higher."
Robot must not tip over when transporting heavy can with pennies in it in the second challenge.	Will: “We tried the heavier can challenge and what happened unfortunately is, like, when it grabbed the can it grabbed the can but then when we took it off the platform… it toppled right over so we’re probably either going to counter weight it here [touches back of chassis] or put down some stakes with wheels here [motions below motor on arm] to hopefully stabilize this.”
Claw must securely grip soda can such that it does not fall out while being transported.	Will: “So we tested our first prototype. It did grab the can but the can slipped since we didn’t have any friction to hold it in place so we added these gummy little pieces right here [squeezes black gripper pieces on claw] to hold the can in place.”
Arm gears must remain meshed while claw is gripping can.	Kamele: “I noticed the claw wasn’t really closing and opening right so we added a rubber band right here just to keep it together more so that the gears would stay on the springs, er, the spinners [pointing to worm gears] and I also added the grips.”
Arm must be secure and not move up and down.	*See Table II*

TABLE II. WAYS IN WHICH STUDENTS DISCUSSED THE REQUIREMENT THAT THE ARM MUST BE SECURE AND NOT MOVE UP AND DOWN.

Students	Discourse about Stability of Arm
Will and Chris	Will: “Surprisingly it’s stable [wiggling arm up and down]; I wouldn’t expect that. But hopefully it’ll work.”
Kamele	“I’m probably gonna need more support in the front.”
Kylie and Andrei	Andrei: “We added some supports here [points to pieces on side of chassis] because, uh, this piece [grabs the top of the interface] was loose and would move up and down like that so we added supports.”
Alex and Esther	Alex: “Maybe that’s not sturdy enough.” Esther: “Do you wanna reinforce it? Like… [indicates holes where they could add another connection between the arm and chassis].” Alex: “Sure.”
Tim and Ollie	Ollie: “We added a little bit- we added more- one of these [a connector between the arm and the chassis] to increase the stability because this [shakes the arm up and down] is moving around too much.”
Natalia and Aanya	Natalia: “We stabilized it [the arm] using a couple of blocks [points to pieces of the interface] so it wouldn’t fall over. Yeah, so, it’s pretty sturdy [wiggles arm up and down]. It doesn’t fall.”

specifically looked for students discovering incorrect, incomplete, or new and yet-unfulfilled requirements, and not just making reactive, ad hoc changes. The third learning objective was also intended to demonstrate how design changes made to one subsystem can propagate through the system. We looked for students considering the effect of each modification on the interface and other subsystem.

The video of the test setup shows students frequently coming over to check some aspect of their robot. While we encouraged groups to see if they could complete each challenge on the first test, not all groups did. If students were not successful on the first test they continued to iterate and re-test their robot until they were successful or ran out of time. In their videos, students also reflected on their checks and tests and the modifications they made in response. Kylie and Andrei, who were followed in the analysis of Learning Objective 1, demonstrate this in their third video. Checking their robot against the test setup brought up the requirement that the arm must be tall enough to clear the top of the platform and they modified their robot in response. Later on, their first test added a sub-requirement to a requirement they had previously considered. Their support was built in response to the requirement that the arm must not move up and down. However, it had the unintended consequence of making their robot veer to the left. This made them discover the sub-requirement that their support be sturdy and well-constructed.

Other groups also discussed how checking and testing their robot led to changes. Will and Chris's first test led them to discover the claw was not filling the requirement to securely grip the soda can to avoid dropping it during transport. As they discussed in their video:

> Will: So we tested our first prototype. It did grab the can but the can slipped since we didn't have any friction to hold it in place so we added these gummy little pieces right here [squeezes black gripper pieces on claw] to hold the can in place.

Through testing, a number of groups found an unsatisfied and previously-unknown requirement that the arm gears must remain meshed while claw is gripping can. In his video, Michael stated, "And, uh, the motor was making these pop out– these, the claw pop out. And, yeah, we made a change." The change involved adding an axle piece across the arm gearbox to keep it together. Kamele discussed both the requirements of gripping the can and keeping the gears meshed in her video:

> Kamele: "I noticed the claw wasn't really closing and opening right so we added a rubber band right here just to keep it together more so that the gears would stay on the springs, er, the spinners [pointing to worm gears] and I also added the grips."

Students also discussed ways in which the integration or design changes made to one subsystem propagated to other subsystems. Multiple groups made modifications to the chassis after they had integrated the arm. Will and Chris were one such group:

> Will: "So, yeah, now we're building the base of it… and supporting the claw. And we're gonna add some wheels at the bottom to make it more stable 'cause right now it's a bit wobbly."

Other groups also recognized that the robot was unsteady without modifications being made to the pre-built chassis.

The second part of the activity was designed to promote changes that propagated through the interface and subsystems, and many students observed and responded to this. In order to reach the soda can when it was further away from the platform edge, Tim and Ollie extended the length of their interface (Figure 4). However, when they went to test their robot they discovered the weight of the arm enacted a moment on their robot and lifted the wheels off the ground. They discussed this problem and their solution in a video:

> Ollie: "So, our finalized design made this [the interface and arm] much longer 'cause there's a new challenge and we also added more weight to the back because this– the wheels kept on lifting up off the ground. So we added the weights because this [the interface and arm] was so heavy. So now it actually works."

> Tim [whispering]: "Say the thing about the iPad. iPad"

> Ollie: "And we also added an iPad on the top to increase the weight, otherwise it still went like that [lifts the back of the chassis up so the wheels aren't touching the table]."

Tim and Ollie modified their interface and arm to satisfy a new requirement of the second challenge, but later realized that this caused unforeseen consequences and required modifications to another subsystem. They added the heaviest LEGO pieces they could find to the back of the chassis to serve as a counterweight. This did not completely solve the problem in the time provided, so they resorted to putting an iPad on top of their chassis above the wheels. A number of other groups encountered similar problems during the second part of the activity, particularly when attempting to retrieve the heavy soda can filled with pennies. Will and Chris' robot, for example, tipped forward with the weight of the can, also motivating them to add a counterweight to the chassis.

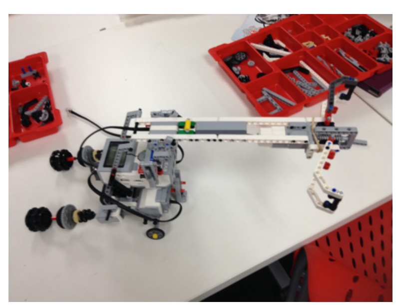

Fig. 4. Tim and Ollie's final robot design, which included counterweights on the back of the chassis in order to balance out the moment enacted by their long interface and arm.

V. DISCUSSION

In looking at the qualitative video data, we found evidence showing students engaging in systems engineering methods that related to the three learning objectives of our activity. These findings suggest with the right activity pre-college students are capable of understanding and implementing basic systems engineering concepts. Performing functional decomposition, considering requirements, and understanding how changes to a subsystem propagate throughout a system represent a more sophisticated approach to engineering design. Rather than reacting to each individual issue as it arose, the students in our workshop appear to have kept the overall task in mind. When they discovered an issue through checking or testing their robot, they made modifications that went beyond the subsystem directly affected by the issue. Students often modified the interface or another subsystem in order to respond to a new requirement created by the modification (e.g. Tim and Ollie adding counterweights to the chassis to account for the longer arm needed in the second challenge) or to ensure that a previously-existing requirement continued to be satisfied (e.g. Kylie and Andrei continually changing the support for their robot arm).

Teachers should work to model, identify, and encourage these productive behaviors during all engineering design projects. They can use the language of systems engineering, such as "requirements," "interfaces," or "subsystems" when setting up a design challenge to model how students should be discussing and considering their work. Furthermore, they can identify when students are engaging in these productive systems engineering behaviors and encourage them to continue or use them as an example for the class. If these students are not using the language of systems engineering, the teacher can name the specific concept at this instance, tying it to students' personal experiences.

VI. CONCLUSION

In this research study we qualitatively analyzed video data from two systems engineering workshops to find examples of how students engaged in systems engineering methods related to three learning objectives: 1) a system can be decomposed into subsystems, 2) designers coordinate activities by communicating requirements and interfaces, and 3) testing reveals problems and changes propagate to other components. Students did not use the language of systems engineering during the workshop, but we found evidence of all three learning objectives in their discourse. This included discussion with their partners and with the researchers during the workshop, and in short iPad videos explaining their LEGO EV3 robot designs. While it is difficult to say with certainty, some students appeared to demonstrate systems thinking by considering their robot as two integrated subsystems—an arm and a chassis—instead of two isolated components. The same students also appeared to show evidence of performing functional decomposition and considering alternate physical forms for accomplishing each function. Students identified many requirements driving their design. One often-discussed requirement was that the arm be "stable" and not move up and down too much. Groups across both workshops developed this requirement on their own, likely from their prior experience with LEGO robotics. Lastly, students often checked their robot against the test setup and dynamically tested their robot by running the program, both of which indicated incorrect or incomplete requirements. When students made modifications based on these issues, they were able to consider how these modifications affected other subsystems.

There are a number of future directions for this research. In subsequent workshops we would like to record and analyze video of students working with their partners in order to have more insight into their natural discourse when building their robot. This would serve as a complement to the self-produced videos in which they explain their design to the researcher. We do not believe this is the case, but it is possible that the students analyzed in this paper made design decisions in the moment without much thought, and then created justifications later when they made their videos. Taking data of students working with their partner would allow us to assess if this is occurring. We are also particularly interested in how students begin to design their robots: do they jump right in or do they spend time planning with their partner? In future analyses we are also interested in how students' actions and discourse change over the duration of the workshop. For example, do they discuss the requirements of their robot differently after they have had some time building and testing? Does the full-group reflection between the first and second parts of the activity have an influence? Future workshops could also assess students' ability to engage in systems thinking by providing them with a complete pre-built robot and asking them to decomponse it into subsystems on their own. Finally, future workshops could give students pre- and post-workshop questionnaires to quantitatively assess their learning. These results could be compared to our qualitative analyses to better validate both approaches in assessing students' learning of systems engineering principles.

ACKNOWLEDGMENT

The authors would like to thank Prof. Kristen Wendell, Chelsea Andrews, and Dr. Tejaswini Dalvi for their assistance with the data analysis. They would also like to thank Jen Scinto, Barbara Bratzel, and Michael Edegware for their assistance in running the workshops.

REFERENCES

[1] International Council on Systems Engineering, "Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities. C. Haskins, Ed. INCOSE-TP-2003-002-03.2.2, October 2011.

[2] International Council on Systems Engineering, "A World in Motion – Systems Engineering Vision 2025," 2014.

[3] A. Pyster, D. Henry, N. Hutchison, C. Jauregui, and M. Clifford, "Atlas: The Theory of Effective Systems Engineers: Version 0.5," Systems Engineering Research Center. SERC-2015-TR-108, December 2015.

[4] National Research Council, "STEM Integration in K-12 Education: Status, Prospects, and an Agenda for Research," M. Honey, G. Pearson, and H. Schweingruber, Eds. Washington, D.C.: National Academies Press, 2014.

[5] T.J. Moore, R.L. Miller, R.A. Lesh, M.S. Stohlmann, and Y.R. Kim, "Modeling in Engineering: The Role of Representational Fluency in Students' Conceptual Understanding," J. Engr. Education, vol. 102, no. 1, January 2013.

[6] J. Moss, K. Kotovsky, and J. Cagan, "The Roke of Functionality in the Mental Representations of Engineering Students: Some Differences in

the Early Stages of Expertise," Cognitive Sci., vol. 30, no. 1, pp. 65-93, 2006.

[7] D.P. Crismond and R.S. Adams, "The Informed Design Teaching and Learning Matrix," *Journal of Engineering Education*, vol. 101, no. 4, 738-797, 2012.

[8] Technology for All Americans Project, and International Technology Education Association, *Standards for technological literacy: Content for the study of technology*, 3rd edition, International Technology Education Association, 2007.

[9] A. Kossiakoff, and W.N. Sweet, *Systems Engineering Principles and Practice*. Hoboken: John Wiley and Sons, pp. 19-24, 2003.

[10] B. Erwin, "K-12 Education and Systems Engineering: A New Perspective," *Proceedings of the 1998 American Society of Engineering Education National Conference*. Seattle, Washington, 1998.

[11] G. Bartus, and F.T. Fisher, "Outcomes of a Systems Engineering Project for K-12 Teachers," *Proceedings of the 122nd ASEE Annual Conference & Exposition*. Seattle, Washington, 2015.

[12] R. Jain, K. Sheppard, E. McGrath, and B. Gallois, "Promoting Systems Thinking in Engineering and Pre-engineering Students," *Proceedings of the 2009 American Society for Engineering Education Annual Conference and Exposition*. Austin, Texas, 2009.

[13] R. Jain, M. McKay, B. McGrath, and D. Brockway, "Translating Systems Engineering for High School Teachers and Students: An Exploratory Study of Implementing Some Initial SE Concepts." *International Journal of Intelligent Defence Support Systems* 2(3):143-156, 2009.

[14] A.W. Johnson, S. Willner-Giwerc, and P. Grogan, "Developing a Systems Engineering Activity for Middle School Students using LEGO Robotics," *Proceedings of the 123rd ASEE Annual Conference & Exposition.* New Orleans, Louisiana, 2016.

[15] National Aeronautics and Space Administration, *NASA Systems Engineering Handbook*, NASA/SP-2007-6105. Washington, DC: Nataional Aeronautics and Space Administration, 2007.

[16] P.M. Sadler, H.P. Coyle, and M. Schwartz, "Engineering competitions in the middle school classroom: Key elements in developing effective design challenges," The Journal of the Learning Sciences, vol. 9, no. 3, pp. 299-327, 2000.

[17] M. Schreier, "Qualitative Content Analysis," In *The SAGE Handbook of Qualitative Data Analysis*, Ed. Uwe Flick. London: SAGE Publications, pp. 170-184, 2014.

[18] A. Strauss, J. Corbin, and others, *Basics of Qualitative Research*. Newbury Park, CA: Sage, vol. 15, 1990.

[19] A.T. Jeffers, A.G. Safferman, and S.I. Safferman, "Understanding K-12 Engineering Outreach Programs," *Journal of Professional Issues in Engineering Education and Practice* 130(2):95-108, 2004.