# Teaching Systems Engineering During the First Semester

**Massood Towhidnejad**
*Embry-Riddle Aeronautical University*

## Abstract

The complexity of the systems being build are ever increasing, and the traditional engineering curriculum is having a hard time to keep up with preparing their graduates to work on such a complex systems. In recent years, most engineering and computing degrees are integrating capstone design project as part of their senior curriculum. Only part of these projects require students to work on a complex multidisciplinary projects, and some do not even require teamwork. Requiring students to work on a multidisciplinary project is a step in the right direction for preparing the graduates to enter the workforce. However, waiting until the last year to expose the students to such projects are just too late. This paper describes a multi-disciplinary introduction to engineering course that is offered during the first semester freshman year.

## Keywords

Systems Engineering, Team Project, Course Design

## Introduction

These days, we rarely find systems that are built by one or two person at a single location. In addition to the increasing complexity of the systems that are being built, these systems also require developers from the different disciplines, perhaps from around the world, to work on a large system that may be built out of multiple sub-systems. This is not a new phenomenon, building large systems started during the Persian and Roman empires, however as we added new technologies, the integration of these technologies introduced new challenges. These challenges became more pronounced with the invention of computers. As computers became smaller and faster, systems designers started to allocate more system functionalities to computers. This trend has been increasing exponentially in recent years. A quick review of the aviation industry reveals the exponential growth of computing both in the form of lines of code (Figure 1) [1], and number of functionalities (figure 2) [2].

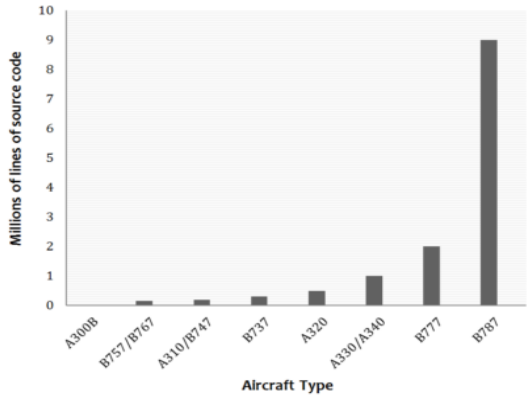



Figure 1. Growth of Software line of codes in Civilian Aircrafts

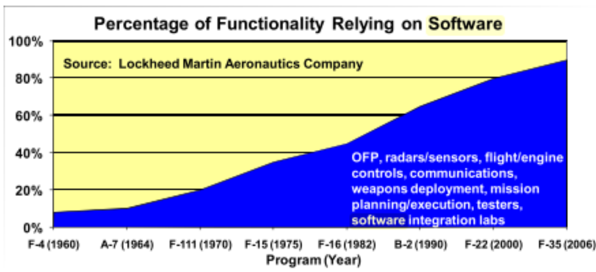



Figure 2. Growth of Functionality relying on software in military aircrafts

These complexity are not only related to the integration of computer. Systems as a whole have become more complex to deliver much challenging functionalities with much better accuracy and efficiency. For example, The Air Traffic Management (ATM) system is build out of 100s of components from diverse domains all working hand in hand to deliver safe and efficient air travel for its user. Some examples of the diverse systems that are uses in the ATM system, includes ground based radar system, global positioning systems, weather systems, air traffic control systems, airlines, etc. It is important to note that the systems mentioned before are all primary systems in the ATM, but each one of these primary systems by themselves may have a large number of supporting systems, that if one fails to deliver its functionalities, it will have a major effect on the overall operation of the ATM. For example, the airlines themselves have number of supporting systems such as maintenance organization, reservation systems, crew management systems, catering, etc. For example, on July 8$^{th}$ 2015, United airline had to issue a ground stop to all their flights for 90 minutes due to a "computer glitch in their reservation system". This resulted in 59 cancelled flights and over 800 flights delayed, and caused major stress on the ATM for the rest of that day [3].

It is obvious that the students who are graduating from undergraduate computer science and engineering degrees are required to work on much more complex multi-disciplinary systems than the students who have graduated twenty years ago. However, there has been no major revision to the undergraduate engineering and computer science curriculum in the last 20 years to address this change on the demand. One significant change in the undergraduate degree is the requirement of the capstone design as part of the curriculum, that is advocated by the ABET program criteria. Although this is a step in a right direction, this requirement still have not gone far enough to require the capstone design to be a multi-disciplinary team project, and in most cases, programs satisfy this requirement by introducing a single course (or a two sequence course) during the senior year. As a result, under the best case scenario, large number of students graduating with their bachelor degree will have their first experience to work on a team project in the industry, as they are getting ready to join the workforce.

This paper describes the first semester Introduction to Engineering course that is offered to all freshman students in the Computer Science, Electrical, Computer, and Software Engineering students. The remainder of this paper describe the course, and what we hope to achieve through this course.

**Introduction to Engineering Course**

As part of the freshman curriculum, all students in the engineering and computer science degree are required to take a two credits introduction to engineering class. Over the past ten years, this course has gone through number of redesigns. In its initial version, all the students in engineering degrees were randomly assigned to different sections, and all sections met one hour at the auditorium, where topics of interest were presented in the assembly. Some examples of these topics included introduction to each engineering program at the university, engineering student clubs, and industry guest speaker, ethics, etc. In addition to the general assembly, students also met one hour a week in a smaller sessions (up to 25 students), where they all work on some projects. Some of these projects were just paper and pencil, with a very limited technical challenges. In recent years, the role of big assembly class has been minimized, and more emphasis are given to the small classroom session. In recent years student met twice a week, where the class meetings are a combination of lectures and hands on activities.

The freshman students in the computer science, software engineering, electrical and computer engineering are registered in special sessions of the introduction to engineering course EGR-ECS. There are several reasons for having these students to all participate in the EGR-ECS class. Some of these include;

- The students in these four degree share the same curriculum in their freshman year.
- Students in the hardware track (EE and CE) share the same curriculum in their sophomore year. The same is true for the software track (CS and SE).
- All students in these four degree will be required to participate in the common capstone sequence course during their senior year. They will be all required to work on multi-disciplinary capstone project.
- Students in the computer science, computer engineering, and software engineering are required to participate in a multidisciplinary software engineering project during their sophomore/junior year. Students in the electrical engineering class are required to take introduction to systems engineering during their junior year. Both courses, emphasis on different phases of the development life cycle in addition to some coverage of the project management and process.

The major objective of the EGR-ECS class is to allow the students from the above mentioned degrees to work on a multidisciplinary project throughout the semester. In addition to working on a multi-disciplinary project, there are number of objectives for the EGR-ECS class. These are;

- Students to get exposed to all phases of the system development life cycle

- Students to appreciate the importance of documentations.
- Student experience working as part of the team
- Students experience technical presentation (demonstration)

## Class Organization

On the second meeting of the class, the students were divided to groups of three. Since this is the first semester that these students are attending the university, there is no prior knowledge related the student's capabilities; therefore it is very important to make sure teams are organized in a way, that as much as possible they have the same level of capabilities. In order to accomplish this, students were asked to complete a small survey describing their interest, previous experience in programming, working with a robot and/or any hardware component, extracurricular activities during high school, and any AP or advance courses in math and science. Given the result of the survey, and the program (hardware or software track) they pursue, students were organized into teams of three. We want to make sure each team end up with at least one student from hardware track, and another students from software track. We also wanted to have each team to have at least one person who have either previous programming background or robot work experience. As a result, out of 17 teams in the two sessions of EGR-ECS, all but two teams end up meeting all the requirements. Once the teams where organized (third class period), students were asked to seat with their team members the rest of the semester. This allow the students to bound with their teammates, and also allow the teams to spend time to work on their project during the workday sessions (over half of the class meeting times are assigned as work days).

## The Project

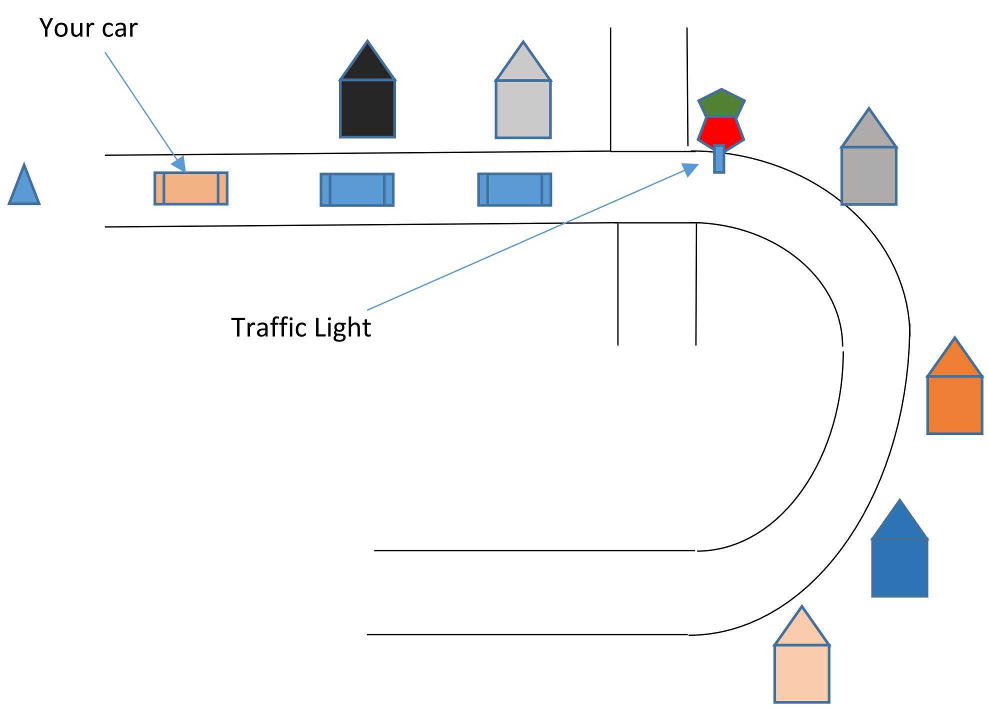
There are number of objectives for this course, but one of the key objective is to allow students to get some hands on experience as part of their first semester experience in the university. In previous offerings of this class, students were required to work on a team project(s), however, there was no real emphasis on the development life cycle. As a result, some of the teams would just go through a series of trial and error (sometimes referred to as hacking process) until they get their project completed. This semester, we introduced the concept of systems engineering (at a very abstract level), and discussed the different phase of the development life cycle. A special attention was given to the design and maintenance (adding features) which required additional attention to project documentation.

During the course of the semester, students were asked to work on a sequence of three projects, where they will be using a robot called "Boe-bot" by Parallax corp. [4] to complete series of tasks. Each projects is built on the capabilities that are delivered on the previous project, and it requires the student to work on both the hardware and software subsystems. Through this projects, students are exposed to different areas of systems engineering, with a major emphasis on the systems development life cycle, and project management. Given the fact that the follow-on projects are built on the previous project, major emphasis were given to the design, testing and maintenance phase. Fifty percent of the student's grade is associated with their project work. The following is the description of the final project.

**Project description:** The final project required students to develop an autonomous vehicle that drives on the road while satisfying the following requirements.

The roadway is painted white and the shoulders are identified by two solid black line (one on each side of the road). The minimum distance between the shoulders is 4 inches. The vehicle starts at the beginning of the road, and it will continue until it reaches the house that is painted blue. All the houses will be located on left side of the road (beyond the road shoulder). In addition to the blue house, there are number of additional houses but none of them are painted blue, green or red. As the vehicle travels the road, it may encounter traffic lights that could be either green or red. If the traffic light is red, car need to stop until the light turns to green before proceeding forward. In addition, there may be other cars on the road, which you need to avoid hitting them.

Following represent a potential roadway,  indicates start location.




In order to make sure students are able to deliver the final project, they were asked to deliver the functionality of driving the car between the lines as part of their project one. In order to do this, students were asked to first deliver their flow chart, in addition to the list of assumptions they made in their design. This document served as their design document, and was graded before they can start their implementation. Once they completed their first project, they asked to submit their project report, which included, their design, implementation/code, and test results. The same process was used for project two, which required the students to deliver the functionality of watching out for other cars, and making sure they do not hit them. Again students were asked to deliver their flow charts for avoiding hitting another car before they completed their project two. However, for project two report, students were asked to use the project one report and make the appropriate changes to the report, this included modification of their project one flow chart and code. For project three, students were asked to deliver all the necessary functionality for the final project. Again, students were required to update their project two document to represent the design and implementation of their project three.

## Semester Results

As previously mentioned, we started with 17 teams, where each team had three members. During the course of the semester two students dropped the class (both after project one), and one student stopped coming to class. Two out of the three students were from the same group, the remaining member of this group was asked to join the team that lost one student. As a result, we end up with 16 teams from project two and three. Out of the sixteen teams, four teams did not complete their project two on time. Two out of the four team were able to catchup and complete their project three on time. However, one of the twelve teams who completed their project two on time, did not complete their project three. Therefore, thirteen team completed all the required functionalities of project three, and three team partially completed the requirement.

## Conclusion

Student transition from high school to university is associated with a series of challenges, including the study habit adjustment, moving out the parent's house and leaving independently for the first time to just name a few. Exposing students to the team work, and working on a hands on multidisciplinary project during the first semester freshman year introduces additional challenges. Some of these challenges include;

- There is a relatively wide range of study habit between the freshman year students. Some of these differences are due to the student experiences in the high school, and the curriculum that is taught in their school (i.e., different states may have different graduation requirements, and different districts have different achievement level due to the population they serve).
- There is little to no opportunities for high school students to work on a complex, not well defined or open ended projects.
- There is very little opportunities for high school students to work as a member of a team. More than 65% of the students enrolled in the EGR-ECS class have no previous team work experience.

In addition to the above challenges, asking the students to follow some system engineering best practices such as design before implementation, required documentation, and formal testing introduce more challenges for the students. Therefore it is very important for the students to know why they are asked to deal with these challenges. In order to accomplish this, we spent several class meetings discussing the size and complexity of some of the systems they deal with on a day-to-day activities. One of the systems we discussed in the class include, Mars Rover, and the level of accuracy that is needed to make the travel between earth and Mars, and the pinpoint accuracy to land the rover in the Mars surface, another system was the drones that are being used by military. Once the students understood the magnitude of the challenges they face after the graduation, they better appreciated the reason behind what we were doing in this class.

The end of the semester course evaluation that was completed by the students have some interesting comments. The comments regarding the project were relatively positive. Almost all students like the idea of having an opportunity to work on a hands on project. Some students did not like the idea of having the follow on projects to be built on previous project capabilities. Number of students suggested having additional instructor and teaching assistant availability. This was a valid complain, as we had an unusual semester, where the TA was involved in a major unmanned system competition, and my travel during this semester was unusually high due to the fact that I was not initially assigned to teach this class.

## References

1 D. Cofer and D. Ph, "Taming the Complexity Beast," 2015.

2 Software: The brain behind the UD defense systems, https://www.atkearney.com/documents/10192/247932/Software-The_Brains_Behind_US_Defense_Systems.pdf/69129873-eecc-4ddc-b798-c198a8ff1026

3 United airline computer glitch, http://www.latimes.com/business/la-fi-united-flights-grounded-20150708-story.html

4 Board of Education shield bot robot, http://learn.parallax.com/tutorials/robot/shield-bot/robotics-board-education-shield-arduino

## Massood Towhidnejad

Massood Towhidnejad is Professor of Software Engineering in the department of Electrical, Computer, Software, and Systems Engineering at Embry-Riddle Aeronautical University. His research interest includes; software engineering, software quality assurance and testing, autonomous systems, and Air Traffic Management (NextGen). He has collaborated with various government agencies (i.e., NASA, FAA, and NOAA) and industries, securing about four million dollars in funding. The results of his research have been published in over eighty technical reports, journal articles, book chapters, and conference proceedings.