Optimizing with Uncertainty: Cross Disciplinary Problem Solving

Grand Challenge Scholars Program Portfolio

Theme: Joy of Living

Luke Metz, Class of 2015, Engineering: Computing

This portfolio is submitted in partial fulfillment of the Olin Grand Challenge Scholars Program requirements

Abstract:

Engineering problems vary and can be approached through a number of different processes. Selecting the best process to fit the problem is a key part of making improvements and reaching the desired solution. This portfolio examines some of the various approaches available to an engineer and develops a framework for problem solving that draws inspiration from machine learning, design process, entrepreneurship, and many other fields. Using the best process to fit the task allows an engineer to make efficient and effective progress.

Portfolio:

I got into engineering because I wanted to solve problems. There are a number of different ways to think about what it means to “solve” a problem, but for me, the goal is to get the optimal solution. It’s easy enough to create something that works, but progress and innovation does not come from sub-par solutions. To get these optimal solutions, however, requires using the best process for the job. Though this may seem simple in principle, in practice, it’s far more complicated. As part of my engineering education, I have been exposed to a number of different fields, each of which problem solves in a different way. Mechanical engineering might use a lot of CAD and simulations, where electrical engineering might make use of fast prototyping via breadboards. Design strives to work with the environment or the users, where math solutions, on the other hand, hardly pose any environmental or user elements as they are just equations and facts. Each field is vast and complex, with no right way to solve problems. The power of thinking about problem solving this way is that one can mix and match processes from different fields to best suit the problem or requirements.

As an illustration of some of the complexity involved in problem solving, take, for example, the purportedly simple challenge of deciding what to eat for dinner. The computer scientist in me approaches this challenge by distilling the problem into distinct pieces: the information I know about the problem, the potential solution, and the results I learn when trying out the solution. Despite sounding simple, optimizing my dinner choices can be tough. How do I decide between ice cream or kale? How do I balance my desire to eat tasty food yet still be healthy? Internally, I am weighing the different factors and deciding on a solution that matches my objective. Sometimes, I make the wrong choice. A particular noodle dish might look good, but turns out to be bland and oily. Was it worth the risk of trying it or should I have just stuck with what I knew was good? Our mental model can never match the reality of a situation, as we need actual facts, not our hopes or expectations. There can be indicators, like what other people think, or what similar foods taste like, or what my experience was the last time I tried a certain dish, but one can almost never predict the actual result without taking the action to try it.

Reduced to these simple terms, the above example gets at the heart of what it means to solve problems. It's a very complicated process; further, the approach one takes to solving the challenge is a key part to the success of finding the optimal solution.

This way of thinking about problems in terms of optimal results as the measure of success stems from the field of Machine Learning, a subject in which I am very interested. Given some amount of information or knowledge about a situation, machine learning's goal is to predict something meaningful. This could be anything from classifying types of plankton in images, to getting a robot to walk stably, to predicting hotel rankings. The goal is always the same: to get the optimal result. Developers build models based on data and the information garnered from how the world reacts and validates these models. Much like in the dinner example, these models are complex, but unlike choosing the optimal food, they are completely computable given sufficient time. This is nice because one can find good solutions and can experiment. One subset of the field that I am incredibly excited about is Bayesian optimization. The goal is simple: find an optimal value of some function as quickly as possible. This could be finding the best ratios to use for ingredients when making cookies to finding the best model to recognize if there is a cat in a photo. It takes a while to make a batch of cookies or run a cat detection model, so learning about the problem space as quickly as possible and minimizing the number of runs needed is advantageous.

Not all problems, however, can be handled by means of this formalism. It's easy to change the amounts of flour or water in cookies, but it's quite difficult to use the same problem solving framework when trying to write an essay or design a product. The concepts just don't mesh perfectly, but the ideas behind the method are applicable: one wants to perform actions to increase knowledge and find a good solution using the most efficient process. Processes vary, however, and a good problem solver selects and employs the best approach to the challenge. In the next few paragraphs, I will show examples of different problem spaces and the corresponding problem solving techniques that I was exposed to in my time at Olin.

In one of my classes called User Oriented Collaborative Design, UOCD, for example, I learned how to design a product for a specific group of people--in my case, farmers. UOCD's approach to it was both incredibly clever but also quite simple. It was an iterative process, where we constantly evaluated our thoughts with our users to better focus our efforts. We first talked to the farmers, got a sense of how they think, and what they valued. When we started the project, I thought farmers were averse to technology and change, but what we learned upon interviewing our user group was that they farmed because they saw our society out of balance. They wanted to change the world and bring their community back to a kind of balance with the earth. Next, we used this information to ideate. Instead of guessing wildly as to what our user group of farmers might need, we used the information we gathered to guide and inform ideation. Instead of focusing on ideas such as a new shovel or better watering gadgets, we ideated around how to get the community more involved, which was something they really valued. To better attune our ideas with our users, we brought our concepts to them and ideated together. By working back and forth with our users, and employing low resolution prototypes such as blocks of foam and post-it notes, we were able to produce a quick succession of possible solutions ranging from educational gloves to carved vegetables. Instead of focusing in on one idea or another, we explored many thoughts about the problem so as to not miss any potential areas. While we identified community involvement as a big area of farmer concern, we deemed it too large in concept and felt that our designs would not be likely to make a large impact. We didn't start to specialize until after we reviewed many possibilities. This approach allowed us to take an idea like "a wearable to keep a farmer safe" to a fully realizable product with a form, a shape, specific user applications, and so on. For me, it was a lesson in how even a big and ill-defined problem space can be solved with the right process.

In Fundamentals of Business and Entrepreneurship, FBE, we learned how to solve problems that fell more into the entrepreneurship space. While similar in uncertainty and scope to design challenges, the entrepreneurship community's drive to get out there and sell a product as fast as possible has informed its problem solving process. The slower it takes to get a product to market, the more money is needed. Instead of the slower pace of getting to know the users as my team employed in UOCD, we jumped right into brainstorming products, figuring out who might buy them, and how we might bring the ideas directly to the potential market. I brainstormed many product ideas from decorative night lights, to cardboard furniture, to wooden wallets and tested interest. I learned that people weren't excited about lights or furniture, but liked the idea of a new sleek wallet. Next, we made prototypes early in our design process and learned that the wallet was too thick, that a smaller wallet would be needed, and one made of a different material as well. When a piece of the process ended up not working, we pivoted and we were off and running in another direction. When, for example, our laser cutter based manufacturing turned out to be time consuming and produced low quality products, we switched to outsourcing production to a local company. In the end, my team successfully Kickstarted and delivered an ultrathin wallet made out of space age materials -- and exceeded our Kickstarter goal. This was made possible by the rapid iterations we used. The key was to keep moving. We learned that nothing should ever stop an entrepreneur from making progress on the creation of product. If a problem comes up, instead of getting demotivated and halting progress, shift the direction or vision of the project and start working on that in the hopes that the new approach will work out better.

Aspects of UOCD and FBE lessons learned can also carry over to more “traditional” engineering, as I came to understand. Such challenges are, to my mind, the most specific and thus the easiest to evaluate. Before Olin, when I thought about “engineering” problems, this style of characterizing a problem in terms of its concrete, scientific elements came to mind. While such factors are important, however, they are not the whole picture. In Principles of Engineering (POE), for example, our team was tasked to build a cardboard quadcopter. Our finished quadcopter design and construction was the solution we desired. How well the quadcopter performed was the result. For this task, a straightforward engineering challenge, as we thought of it, our internal understanding of mechanics, software, and electronics ruled. Creating and testing full prototypes, however, is expensive and time consuming. Instead, we tested small pieces via software or quick prototypes. We also made extensive use of tools that were meant to emulate the product and help us design in the real world, such as CAD. The only reason this worked out (and where the POE challenges differ from those of

UOCD and FBE) is because the world of materials and construction behaved generally as we expected, which is never the case when the problem solving requires creating for people. With a final project near completion, we shifted our efforts from big picture design to debugging and making the thing fly. In some respects, "traditional" engineering problems are straightforward and are what we have been trained to do most of our lives. Designs are easier to test, and the results are easier to measure. The whole process is easy to inspect and to gauge progress. Still, drawing from other problem solving processes, we were able to reach our most efficient solution relatively quickly.

The number of processes I've been exposed these past four years has allowed me to expand my thinking as to how to approach a given challenge. I have been striving to incorporate more of this cross disciplinary problem solving in my activities, with good results. One project in particular, for example, involved creating a program to classify different kinds of plankton. I spent the beginning of the project looking and learning, scouring the Internet, and reading dozens of papers on image classification. Instead of diving right in to development, much like in design problems, I concluded that my time was better spent educating myself and learning about the space. When I had a good understanding of how the various pieces of the project worked, I then made quick progress based on my well informed mental model. As I started to learn more about the challenge, I then experimented with different techniques ranging from convolutional neural networks to autoencoders. Each run I did, I strived to increase my understanding of the underlying system. If something didn't work, I moved on. I treated it like another datapoint, instead of a defeat. I also started to run tests in parallel to try to speed up my progress, as I wanted to be able to learn and improve my model as fast as possible. This cross disciplinary approach, then, allowed me to make rapid, informed progress and reach my desired goal.

Going forward, this model of drawing techniques from several problem solving approaches seems particularly useful to problem solving in my area of concentration: machine learning. To me, the most fascinating thing about machine learning is the unknowns. The models used are so complex that we have ceased to be able to understand them. As such, computer science problem spaces have become more similar to design spaces and entrepreneurship spaces, where the result is not easily forecast. Unlike these spaces, however, the speed at which improvement and iteration can be made is determined by the hardware and not from interacting with users. Taking the most applicable elements of a cross-disciplinary approach will allow real progress in my problem solving techniques.

In some respects, my education at Olin has been about exposure to the various processes available to an engineer and learning how to select the right approach, or elements from several approaches, to solve problems effectively. By being a project based curriculum, I've had ample time to experiment with various different techniques. In a class called Design Nature, we were tasked to build a toy for 4th graders. Before this, I hadn't talked to anyone that young in many years, let alone designed and built a toy that they would enjoy. Still, we learned through an examination of real world chipmunks and created something mechanical that moved in a pleasing manner. In Modeling and Controls, I learned how to work with electronics: how to conceive and create circuits, how to prototype on breadboards, and how to move that to a more stable solution. In UOCD, I engaged in user dialogue. In FBE, I balanced market research against the realities of product production. In my plankton research, I combined the most relevant elements of all the foregoing processes to come up with the most efficient approach to the problem. What I've learned from exposure to these various approaches is that there's always a way to go at a problem, even challenges that are beyond initial comprehension, such as those presented in machine learning. I've also learned that the real value of problem solving lies in an intelligent use of process and not in the quick grab at a solution. The world always changes. My interests move on and alter. My environment changes. By looking at problem solving in terms of a process to maximize a final result, I can make progress regardless of my understanding or skills in an area. I'm encouraged by this understanding and feel that my Olin education has well equipped me to reach the optimal solution of whatever problem I undertake. This, to me, and as stated at the outset of this reflection, is the goal of engineering practiced at its highest, most efficient level.

At the end of the day, however, figuring out how to solve problems is in itself a process, the depths of which are still unfolding.. I recognize that though I've been exposed to a number of ways to approach a challenge, not everything I may seek to resolve will fit the foregoing problem solving frameworks. Fields of study evolve and the tools and approaches adapt accordingly. Still, having a place to begin and something to work off of helps and reveals where I could learn more. As I go off to work in the “real world,” I am interested to see what techniques have been built and how work gets done. My expectation is that I’m well prepared to meet the challenges and even to advance the processes for getting optimal results.

Links:

Personal Site: http://lukemetz.com

Quadcopter Instructable:
http://www.instructables.com/id/Autonomous-Cardboard-Rasberry-Pi-Controlled-Quad/

UOCD website: http://design.olin.edu/courses/uocd/

Design Nature website: http://design.olin.edu/courses/dn/

Addendum:

The foregoing reflection touches upon many of five pillars necessary to becoming a Grand Challenge Scholar. This addendum identifies and amplifies these areas of investigation as experienced as a student, research assistant, and summer intern.

Interdisciplinary Curriculum:

- Course: Stuff of History
    - Studied how materials and technologies shaped history from a historical and a material science perspective.
- Course: Modeling and Simulation
    - Examined how to go about modeling systems. The course consisted of a mixture of physics, mathematics, biology, and computer science.

Entrepreneurship:

- Course: Foundations of Business and Entrepreneurship
    - Learned the basics of entrepreneurship as well as how to create a small business. My team designed an ultra thin wallet, funded it through a successful Kickstarter campaign, and produced and delivered it to our customers.
- Two summer internships with Onshape, Inc., an early stage CAD startup.

Global Perspective:

- Course: Human Connection

  - Explored different perspectives on life in this anthropology class, studying a range of people from workers at a needle factory to Hindu monks.
- Course: Environmental Documentaries
  - Watched and discussed documentary films from across the globe and analyzed them from a media perspective as well as from an environmental point of view.

Service Learning:

- Course: User Oriented Collaborative Design
  - Worked with and designed products for Community Supported Agriculture farmers.
- Course: Design Nature
  - Designed a game for local 4th graders.

Large Projects:

- Artificial Intelligence Writing Assistance
  - Examined recent advances in machine learning to improve the writing and editing processes.
  - This research was conducted as a part of the GCSP Student Summit program.
- Designed and Built a Normal Map Editor for 2D Mobile Games
  - Normal maps can be used to add graphical effect to 2D games. Creating them can be hard and time consuming. My team and I created a product to make the process easier.
- Research at Olin Robotics and Bioinspiration Lab
  - In this multi year project, I used motion capture to evaluate robot run performance, and machine learning to develop controls.