

# Cultivating professional technical skills and understanding through hands-on online learning experiences

Julia Mullen

*Lincoln Laboratory Supercomputing Center  
MIT Lincoln Laboratory  
Lexington, MA, USA  
jsm@ll.mit.edu*

Lauren Milechin

*Earth, Atmospheric and Planetary Sciences  
Massachusetts Institute of Technology  
Cambridge, MA USA  
lauren.milechin@mit.edu*

Albert Reuther

*Lincoln Laboratory Supercomputing Center  
MIT Lincoln Laboratory  
Lexington, MA, USA  
reuther@ll.mit.edu*

**Abstract**—Life-long learning is necessary for all professions because the technologies, tools and skills required for success over the course of a career expand and change. Professionals in science, technology, engineering and mathematics (STEM) fields face particular challenges as new multi-disciplinary methods, e.g. Machine Learning and Artificial Intelligence, mature to replace those learned in undergraduate or graduate programs. Traditionally, industry, professional societies and university programs have provided professional development. While these provide opportunities to develop deeper understanding in STEM specialties and stay current with new techniques, the constraints on formal classes and workshops preclude the possibility of Just-In-Time Mastery Learning, particularly for new domains.

The MIT Lincoln Laboratory Supercomputing Center (LLSC) and MIT Supercloud teams have developed online course offerings specifically designed to provide a way for adult learners to build their own educational path based on their immediate needs, problems and schedules. To satisfy adult learners, the courses are formulated as a series of challenges and strategies. Using this perspective, the courses incorporate targeted theory supported by hands-on practice. The focus of this paper is the design of Mastery, Just-in-Time MOOC courses that address the full space of hands-on learning requirements, from digital to analog. The discussion centers on the design of project-based exercises for professional technical education courses. The case studies highlight examples from courses that incorporate practice ranging from the construction of a small radar used for real world data collection and processing to the development of high performance computing applications.

**Index Terms**—MOOC, hands-on laboratories, project-based, STEM, andragogy, self-paced, professional education

DISTRIBUTION STATEMENT A. Approved for public release. Distribution is unlimited. This material is based upon work supported by the United States Air Force under Air Force Contract No. FA8702-15-D-0001. Any opinions, findings, conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Air Force.

## I. INTRODUCTION

It is well understood that professionals need to continue learning in order to keep pace with advances in their field. However, traditional avenues for professional education tend to bifurcate into a training track focused on how to use new technologies and a formal degree track focused on theory. Among the promises of Massively Open Online Courses (MOOCs) is the potential to merge theory and practice in an educational environment designed from the key andragogical principles that define adult learners. Knowles [1] described the characteristics of adult learners as learners who:

- are self-directed,
- bring significant experience to their learning,
- are interested in learning subjects that have immediate impact on their lives,
- are driven by problem solving rather than content and
- are internally motivated.

Furthermore, technical professionals recognize that there is no single solution to a problem, and therefore look for practical applications of theoretical content. They aim to understand the full system and its components in order to effectively evaluate the trade-offs between solution strategies.

By design, MOOCs are capable of supporting these characteristics. The open nature of MOOC courses allows learners to decide whether or not their background is sufficient for a given course, lets them select topics of immediate interest, work through the material in their own time and place and elect whether or not to register for credit. Self-paced courses, in particular, provide the Just-In-Time learning options that adults require in order to balance personal and professional lives while developing new skills and understanding. However in our experience, the course must be designed around problem

solving, troubleshooting and evaluation, moving a residential course online is not sufficient.

For adults, the key to mastery learning is to merge theory with practice using real-world problems. In traditional, formal education, practice is achieved through worked examples, review questions, quizzes, tests and laboratory experiments. In the professional context this translates to real-world design and/or development using software, hardware or physical construction. The solutions include the creation of an architecture, prototype, product or set of strategies created by an individual or team. Traditional academic practice formats are easily transferred online using existing tools and components. What is less clear is how to effectively convert real-world design forms of practice into online course modules. The focus of this paper is methods of addressing this challenge. The next section describes case studies (courses) that use software and physical construction to motivate the challenge. Section III presents the lessons learned building and delivering the courses followed by the summary in Section IV.

## II. CASE STUDIES

### A. Overview

Learning Management Systems (LMS), both traditional e.g. Canvas [2] and MOOC platforms such as Coursera, edX, and FutureLearn [3] – [6], have long incorporated digital hands-on exercises, e.g. essays, blogs, and programming exercises, into coursework. Less common are exercises that require using or building physical systems. Meanwhile, there has been an explosion in the development of science kits designed for blended learning environments. Such exercises provide engineering and science students with important challenges and experiences that cannot be simulated in a digital environment [7]– [9]. Since the goal is to prepare learners to develop solution strategies for their applications, and intuition about the systems they work with, the courses described in our case studies incorporate hands-on assignments. The two programming centric courses described here, *Advanced Database Technologies* and *High Performance Computing Workflows and How to Exploit Them*, use real world applications for programming and computational practice.

While hands-on programming assignments are common, the overlap of MOOCs and hands-on laboratories is rare [10], [11]. Examples of MOOCs with hands-on laboratories using physical devices include *Embedded Systems–Shape the World: Microcontroller Input–Output* and *Embedded Systems–Shape the World: Multi-Threaded Interfacing* from the University of Austin [12], [13], now archived, and MIT Lincoln Laboratory’s *Build a Radar* [14], [15]. The *Build a Radar* course is unique in that the goal is to create a self-paced asynchronous course that includes both physical construction and the collection and analysis of real world, non-idealized data.

The courses described here all contain traditional auto-graded exercises, e.g. multiple choice, fill in the blank, matching, but the focus of this paper is the development of digital and physical laboratory activities. The laboratory exercises, drawn from years of supporting and teaching professional

engineers and researchers, represent real problems in each domain. The courses described here are informal courses offered by a national laboratory to provide deeper insight into engineering topics not generally covered in academic courses so there is no provision for collecting homework or providing grades. However, the modules forming the courses were designed to be easily incorporated into formal academic courses and have been included in academic courses using a blended format.

While rarely explicitly stated, traditional academic laboratory exercises are designed to provide students with the opportunity to address engineering tasks and build skills while giving instructors an opportunity to model proper engineering practice toward problem solving. In particular, laboratories offer a means for instructors to teach students to:

- predict outcomes prior to performing tests or experiments, guided by the questions
  - What do you expect to see?
  - What data do you need to collect?
- develop appropriate tests
  - To locate the source of the problem.
  - To evaluate the fix.
- analyze results and data as part of troubleshooting
  - What do the test results indicate?
  - Does the correction require a minor fix or a major redesign?
- develop solutions
  - How do you resolve the problem?
  - What unexpected outcomes might that produce?
- develop perseverance and confidence
  - Engineering is an iterative process.
  - You are not wrong, you just haven’t found all the bugs and solutions yet.
  - Enhance constructive self-evaluation.

These skills define the learning objectives for hands-on and project-based learning activities and were used as the design principles for the hands-on activities included in the case studies described below.

Good professors and experienced laboratory instructors achieve these goals intuitively, guiding the student as questions arise during a hands-on activity. Conversely, self-taught learners using web-based resources, quickly discover that online videos, Do-It-Yourself resources and commercial companies rarely provide enough information or guidance to help students develop an effective process for troubleshooting code or devices. Where instructor guidance can lead to the development of an engineering mindset, the process of careful testing and deliberation to find a solution, the lack of guidance too often leads to students hacking a solution that is not robust. For self-paced STEM disciplines this is a missing piece and key challenge. Careful thought and design is required to create exercises that build an engineering mindset.

The design of laboratory exercises described in the case studies below addressed this challenge, using the skills listed

above as a guide. The methods vary by discipline and laboratory type, but generally include a mixture of partial credit and sanity checks to guide learners as they craft their solution and develop their intuition. Furthermore, these checks and troubleshooting tips help professionals develop formative self-evaluation skills that can be transferred to their work environment.

The following sections describe three courses, *High Performance Computing Workflows and How to Exploit Them*, *Advanced Database Technologies* and *Build a Radar*. For each case, the design decisions and challenges associated with crafting hands-on laboratory exercises are discussed.

### B. A. High Performance Computing Workflows and How to Exploit Them

Most High Performance Computing (HPC) centers recognize the need to provide their users with HPC training; however, the current educational structure and accessibility prevents many scientists and engineers who need HPC knowledge and skills from becoming HPC practitioners. Starting from a successful approach centered on one-on-one mini-tutorials to jump start new HPC practitioners, the *HPC Workflows and How to Exploit Them* MOOC course is designed to teach learners how to map their serial or small scale application onto a set of common HPC workflows, e.g. high throughput, MapReduce, Leader-Worker and full MPI based parallel applications [16], [17]. Once students understand where their applications fit within a set of canonical workflows, they can focus on techniques to effectively refactor or extend their application to make it suitable for an HPC system. As students transition into practitioners, they recognize the potential of HPC resources and begin to expand their applications in size and complexity. This scaling often requires re-mapping the application to a different workflow in order to maintain or gain efficiency. A key understanding is recognizing when to refactor the parallel computing framework rather than making a series of complicated modifications to the parallel model selected initially. This trade-off is crucial to providing efficient solutions and for proper use of a shared HPC system.

The HPC course begins with an overview of scientific computing and its role in solving and simulating real-world models. Using this scientific computing application space as motivation for HPC, the course continues with an introduction to HPC and HPC systems, including architectural components that are new to students, such as the resource manager (scheduler), high performance networks and the shared file system. Building on these fundamentals, the course focuses on the canonical set of HPC workflows and the way in which HPC systems interact with and support these workflows.

For each of the common workflows a set of teaching examples has been created to provide practice in the conversion of serial to concurrent workflows. The teaching examples are drawn from real applications and form the parallel programming and HPC solution design laboratories. In keeping with the learning goals of laboratories the HPC strategy exercises challenge the students to map applications of increasing com-

plexity to appropriate HPC workflows. Not all examples have an obvious or single correct strategy so students must learn to understand and articulate the trades-offs when making design decisions.

The teaching examples are provided in multiple languages via Jupyter Notebooks[18], as shown in Figure 1, application code and job submission scripts. The programming languages include MATLAB<sup>®</sup>/Octave, Julia, Python and C and are available via GitHub [19]. The Jupyter Notebook environment also provides a terminal window and a text editor in the browser, which creates an approachable Linux learning environment. Additionally, the self-contained nature of the Jupyter Notebook code cells provides a way to present common programming errors that students can resolve in an easy troubleshooting environment. To facilitate self-paced asynchronous learning, instructions are provided on how to run the examples on both a supercomputing system and a multi-core laptop for those students who do not have access to a supercomputer. The instructions also include required software packages, installation instructions and data where appropriate. Methods of extending the teaching examples for batch processing and scaling on a supercomputer are included to prepare students for a time when they gain access to supercomputing resources.

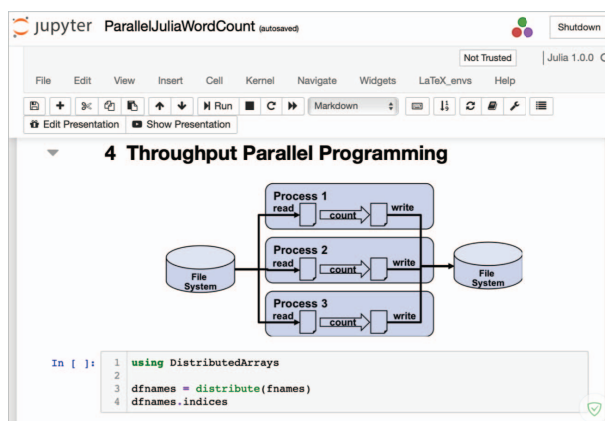


Fig. 1. Jupyter Notebook example for High Throughput Computing.

### C. Advanced Database Technologies

The *Advanced Database Technologies* course was designed to address the challenges of processing and exploiting Big Data for decision making. By construction, the content is presented within the context of a systems approach with a focus on how to pick the right tools and techniques (including emerging database technologies) required to design a system that takes raw data and converts it to a form that an analyst can explore and analyze. The course is organized around challenges that occur in designing a Big Data System: those introduced by the data, those introduced by the computing system (hardware and software), and those introduced by the goals of the analyst. In this way we navigate the trade-space and design decisions that are introduced by each set of challenges through a series of videos, text, and knowledge-check questions.

There are many roles to play in designing and implementing a Big Data System, or Pipeline. Requirements at the downstream end of the Pipeline affect requirements upstream, and decisions made at the start of the Pipeline affect the capabilities at the end. Every member of a team is responsible for a stage of the Pipeline but needs to develop intuition about the other stages in order to create an efficient, effective end-to-end system. In particular, the ability to predict what data is needed, craft tests and understand the results is necessary for modifying the Pipeline when new analytics or capabilities are required. To develop these skills, the course includes hands-on programming laboratories and a final exam where student create the full Big Data Pipeline. The goal for the laboratory exercises is to give students some awareness of the challenges, requirements, and design decisions that must be made in each of these roles. This will help them reason about their own roles in their work, how they affect and are affected by others, and how to operate more effectively as part of a Big Data System.

The final exam/project walks the student through building a Big Data system for a particular problem and use case. Questions about the final state of the data provide a sanity check for students that they have created a working system. The labs, problems, and exam are evaluated using automatically graded questions, such that the student can confirm that they understood and completed the problem.

The course was designed to be integrated with the MIT LLSC and MIT Supercloud High Performance Computing system and Accumulo [20] and SciDB [21] Databases. Since these are rare resources offering a MOOC version of the course required modifying the laboratories while still meeting the laboratory learning goals. Setting up databases can be a complex process not within the scope of the course, so problems are created that can be completed with a backend database or text files, supported by a discussion on when to use which. The exercises use Julia, Matlab<sup>®</sup> and D4M [22]. We provide careful instructions for installing any other required software or languages, as obtaining the required tools can also be a challenge. The examples and problems are provided in a public git repository [19] and laboratories are in the process of being refactored to use datasets that are public and easy to obtain.

#### D. Build a Radar

The pathfinder MOOC driving the convergence of digital theory with physical construction is the *Build A Radar* course. The initial in-person course was lecture based, but the primary component involved physically constructing, testing, and tuning a small radar, shown in Figure 2, and using it to collect data in a variety of radar modes. A key learning objective was to enable students to enter a public space and select a Region of Interest for data collection that was appropriate to the radar capabilities and operational mode. These experiments raise radar collection and processing concerns through the use of real world collection points that return messy, potentially incomplete data. Converting the course content for use in a self-paced asynchronous course followed standard practices,

e.g., the creation of targeted concept videos, crafting review questions and auto-graded exercises that reinforce the theory. The challenge centered on addressing the issues associated with supporting the radar construction, data collection and analysis of real, non-ideal data.

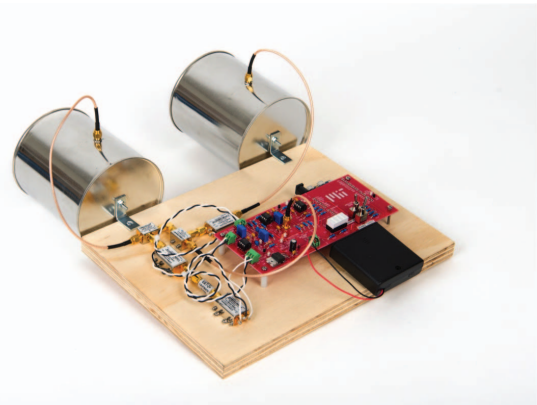


Fig. 2. Small Radar.

To address this challenge the conversion of the in-person course to a self-paced asynchronous course was completed in stages. The initial stage involved the creation of a blended course from the initial in-person model. Using feedback from the in-person course experience with professional engineers and university engineering students the Build Module of the course was designed to include a set of carefully crafted and annotated construction steps. In addition to the annotation, the instructors included a set of construction tips on what to do and what not to do for steps that had proven particularly confusing for during the in-person course. The result of this stage was to move all lectures, the radar construction and initial radar tuning online and hold in person meetings for the analysis of results stemming from data collection and processing. In the blended version both the professional and university cohorts were able to build a working radar using only the online instructions. However, many were either unable to troubleshoot on their own, weren't confident that they had done it properly, misunderstood how to test the device or attempted to collect data in a manner incompatible with the capabilities of the radar. The carefully documented build steps and tips provided guidance for construction but were not enough.

Based on the feedback from the blended course, creating a self-paced asynchronous version required designing additional exercises and examples to guide the student through the troubleshooting, scene selection for data collection and actual data collection activities. In practice this translates to extending the trouble-shooting activities and developing engaging exercises that motivate students to evaluate expected outcomes and adjust data collection strategies prior to collection. This was accomplished by adding detailed troubleshooting checks and tips and extra tuning experiments to the Build Module of the course. Each section of the Build Module ends with a check to confirm that the step was completed successfully.

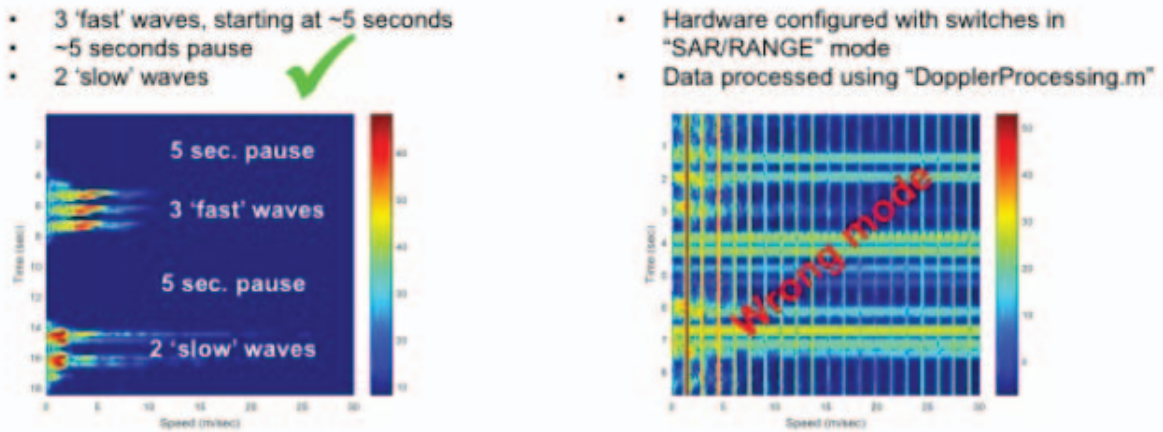


Fig. 3. Functional Testing Process for Build a Radar.

If the student was successful they receive credit, if not, they are provided with a set of troubleshooting steps to resolve the issue. These checks illustrate how the instructors created tests and developed solutions, two of the learning objectives of laboratories outlined in Section II-A and provide the student with confidence that they are building a working radar. The final sections in the Build Module provide functional tuning tests, with expected outcomes, to guide the learner through the process of tuning the radar for the three operating modes. An example of an extra tuning exercise is shown in Figure 3 where the images are provided for results obtained when the radar settings and scene are matched (left) and when there is a mismatch (right). The goal is to prompt students to predict their expected result, understand how to analyze and recognize unexpected results, think about testing to uncover and resolve the issue.

This small radar, running in the ISM band, is capable of capturing data in Doppler, range and Synthetic Aperture Radar (SAR) modes but is limited by the transmit power. During the blended course, the instructors observed that students misunderstood the capabilities of the radar, how to collect data in each operating mode and how to analyze the processed images. In some cases addressing the misconceptions was as simple as providing the detailed functional tests to highlight the steps involved in data collection for each operating mode. A more subtle issue was guiding students in the selection of the scenes for data collection, for example the radar cannot track an insect or small bird and its range is limited compared to the radar in a car. Initially the guidance for students was a list of recommendations, but instructors continued to see students attempt to capture scenes beyond the capabilities of the radar and the recommendations were deemed insufficient. To address this gap new exercises were added to each of the experiment sections, (Doppler, range and SAR). For guidance on scene selection there are galleries of image data comparing data collection locations with the resulting radar images, including videos where appropriate. To help students develop skill in

analyzing the return images, there are collections of exercises describing scenes and requiring the student to select the image that is generated from the scene. Though we expect to make future improvements, these additions provide the guidance required to achieve the objectives of laboratory activities described in Section II-A.

### III. LESSONS LEARNED

The case studies presented here started as in-person courses where the instructor team was able to provide the necessary resources and assistance to support hands-on projects using real world applications and data. Scaling the courses required careful consideration of the ways that students learn to troubleshoot projects in order to provide guidance required to enable students to succeed on their own. During the restructuring process, the instructor team learned the following key lessons:

- Clear, explicit troubleshooting steps must be included in the hands-on work, including:
  - Test or debug points with references to solutions and
  - Controlled tests with known results to confirm correctness at points in the process;
- Software strategies must be compatible with a laptop but easily extensible to a target system to be used by the students;
- Tools must be
  - Easy to obtain
  - Easy to use and
  - Inexpensive;
- It is important to highlight the extensions required to expand from the course project to a real world engineering system;
- Providing context is essential so that learners understand how and where tools and technologies fit into the larger system; and
- It is necessary to highlight the trade-offs (benefits and drawbacks) associated with the design choices.

The instructor teams continue to review, adapt and extend the scaffolding around hands-on activities, particularly those units that guide students through troubleshooting steps.

#### IV. SUMMARY

In depth examples help students develop the intuition needed to evaluate results and determine whether or not they make sense. Thorough troubleshooting guides illustrate how to systematically uncover the source of a problem. Trade-space analyses help students transfer their learning to their workplace and profession. These components are part of rich project-based learning experiences and with careful design can be successfully incorporated into self-paced online courseware.

The courses described here have only recently shifted from a blended to an asynchronous mode, so the analysis and discussion is based on descriptive rather than quantitative results. The Open edX platform [3], on which the courses run, offers the opportunity to collect a great deal of course activity data which can be correlated with auto-graded assessments, but much of the laboratories work is performed outside of the course site. The auto-graded questions related to the programming examples are captured, so some judgement can be made regarding student understanding, but currently there is no way to capture the solution or troubleshooting process beyond following posts to the discussion boards. Future work focuses on developing methods to capture and assess the external laboratory work. This includes determining which data should be captured, and how to gather it. Additionally, new quantitative assessment strategies for work performed outside of the course site will be developed and deployed. These new methods will provide insight regarding data requirement needs.

#### ACKNOWLEDGMENT

The authors would like to acknowledge the MIT Lincoln Laboratory Supercomputing Center and MIT Supercloud teams who built and support our unified educational environment: an Open edX instance and the supercomputing system used for HPC practice for the MIT MOOCs.

#### REFERENCES

- [1] M.S.Knowles, *The Modern Practice of Adult Education, From Pedagogy to Andragogy*, 2nd Edition, Cambridge Books, 1980.
- [2] Canvas, <https://www.instructure.com/canvas/>, accessed on 6-01-2019.
- [3] Open edX, <http://www.open.edx.org>, accessed on 6-01-2019.
- [4] edX, <https://edx.org>, accessed on 6-01-2019.
- [5] Coursera, <https://coursera.org>, accessed on 6-01-2019.
- [6] FutureLearn, <https://www.futurelearn.com>, accessed on 6-01-2019.
- [7] J. Garcia-Guzman, F. H. Villa-Lopez, J.A. Velez-Enriquez, L.A. Garcia-Mathey, "Remote Laboratories for Teaching and Training in Engineering", DOI:10.5772/67459, *www.intechopen.com*, January 2017.
- [8] T. De Jong, MC Linn, ZC Zacharia, "Physical and virtual laboratories in science and engineering education", *Science*, Vol 340, *www.sciencemap.org*, April 19, 2013.
- [9] V. Handur, P. Kalwad, M. Patil, V. Garagad, N. Yeligar, P.Pattar, D. Mehta, P.Baligar and G.H.Joshi, "Integrating Class and Laboratory with Hands-on Programming: Its Benefits and Challenges", *IEEE 4th International Conference on MOOCs, Innovation and Technology in Education*, December 2016, DOI:10.1109/MITE.2016.041
- [10] W.Halimi, C. Salzmann and D.Gillet, "Access to Massive Open Online Labs through a MOOC", *Learning Scale*, Cambridge, MA 2017.

- [11] F. Moosvi, S.A. Reinsberg and G.W. Rieger, "Can a Hands-On Physics Project Lab be Delivered Effectively as a Distance Lab?", *International Review of Research in Open and Distributed Learning*, vol 20, pp. 21–42, February 2019.
- [12] J.Valvano and R.Yerraballi, "Embedded Systems–Shape the World: Microcontroller Input–Output", *utaustinx*, <https://www.edx.org/course/embedded-systems-shape-world-utaustinx-ut-6-20x>, 2017.
- [13] J.Valvano and R. Yerraballi, "Embedded Systems–Shape the World: Multi-Threaded Interfacing", *utaustinx*, <https://www.edx.org/course/embedded-systems-shape-the-world-multi-threaded-interfacing>, 2017.
- [14] J.Mullen, L.Milechin, M.Houle, P.Bell, A.Fenn, K.Kolodziej, J.Meklenburg, J.Nguyen, B.Perry and A. Reuther, "Bringing Physical Construction and Real-World Data Collection into a Massively Open Online Course (MOOC)", *IEEE Frontiers in Education Conference Proceedings*, October 2017.
- [15] K. Kolodziej, P. Bell, A. Fenn, E. Kowalski, J. Meklenburg, W. Moulder, J. Mullen and B. Perry, "Build-a-Radar Self-Paced Massive Open Online Course (MOOC)", *IEEE Radar Conference 2019*, Boston, MA April 2019.
- [16] J.Mullen, C.Byun, V. Gadepally, S. Samsi, A. Reuther and J.Kepner, "Learning by Doing, High Performance Computing Education in the MOOC Era", *Journal of Parallel and Distributed Computing*, DOI:10.1016/j.jpdc.2017.01.015
- [17] J.Mullen, W. Filinger, L.Milechin, D.Henty, "The Impact of MOOC Methodology on the Scalability, Accessibility and Development of HPC Education and Training", *Fifth SC Workshop on Best Practices for HPC Training and Education at SC18*, November 2018, published in *Journal of Computational Science Education*.
- [18] Jupyter, <https://jupyter.org/>, accessed on 6-01-2019
- [19] HPC Teaching Examples, <https://github.com/llsc-supercloud/>, accessed 6-21-2019.
- [20] Accumulo, <https://accumulo.apache.org>, accessed 8-30-2019.
- [21] SciDB, <https://www.paradigm4.com/technology>, accessed 8-30-2019.
- [22] D4M, <https://www.mit.edu/~kepner/D4M>, accessed 8-30-2019.