Transitioning M&S Courses from a M&S Engineering Degree to a Major Under a Computer Engineering Degree

James F. Leathrum, Jr., Yuzhong Shen, Masha Sosonkina, Michel A. Audette
Department of Computational Modeling and Simulation Engineering
Old Dominion University
Norfolk, Virginia
jleathru@odu.edu, yshen@odu.edu, msosonki@odu.edu, maudette@odu.edu

ABSTRACT

Recent circumstances forced Old Dominion University to reconsider the practical support of an undergraduate degree in Modeling & Simulation Engineering (M&SE). Due to large support for the program, in particular from industry constituents, it was determined that the best course of action to provide stability for the program was to migrate it to be a major under the Computer Engineering (CpE) degree program. The new major became available in Fall 2021 with the old degree program being phased out by spring 2025. This resulted in a major restructuring of the curriculum to fit within a computer engineering degree. The resulting curriculum (1) reduces the number of core M&SE courses to accommodate required CpE courses, (2) addresses university requirements previously covered by lost courses, and (3) integrates the M&SE content with core CpE content. This paper focuses on the remaining core courses. It examines how the courses are migrating from their old content and structure to the new requirements, primarily in the context of core M&SE knowledge and how to tie it into a CpE degree. Obviously being under a CpE degree program, the new courses emphasize the computational side of M&SE, while retaining fundamental M&SE concepts. The basic curriculum is presented as the context for the individual courses. Then each course modification is discussed from the perspective of the instructor responsible for the transition. The challenges encountered are highlighted as well as the benefits of students having a Computer Engineering background.

ABOUT THE AUTHORS

James Leathrum is an Associate Professor and Chief Departmental Advisor in the Department of Computational Modeling and Simulation Engineering at Old Dominion University. He earned the Ph.D. in Electrical Engineering from Duke University. His research interests include simulation software design, simulation-based test and evaluation of autonomous systems, distributed simulation, and simulation education. His e-mail address is jleathru@odu.edu.

Yuzhong Shen is a Professor and Chair in the Department of Computational Modeling and Simulation Engineering at Old Dominion University. He earned the Ph.D. in Electrical Engineering from the University of Delaware. His research interests include virtual reality, augmented reality, visualization and computer graphics, transportation modeling and simulation, general modeling and simulation and signal and image processing. His e-mail address is yshen@odu.edu.

Masha Sosonkina is a professor of Modeling and Simulation Engineering at Old Dominion University. She received her Ph.D. in Computer Science and Applications from Virginia Tech. Dr. Sosonkina's research interests include high-performance computing, large-scale simulations, parallel numerical algorithms, performance modeling and analysis.

Michel Audette is an Associate Professor in the Department of Computational Modeling and Simulation Engineering at Old Dominion University and Graduate Program Director in Biomedical Engineering. He earned the Ph.D. in Biomedical Engineering at McGill University. His research interests include medical simulation and therapy planning, with clinical applications spanning neurological, orthopedic, thoracic, and cardiac surgery, musculoskeletal simulation applied to geriatrics and obstetrics, and physiological simulation to potentiate warfighter and pilot performance. His e-mail address is maudette@odu.edu.

Transitioning M&S Courses from a M&S Engineering Degree to a Major Under a Computer Engineering Degree

James F. Leathrum, Jr., Yuzhong Shen, Masha Sosonkina, Michel A. Audette
Department of Computational Modeling and Simulation Engineering
Old Dominion University
Norfolk, Virginia
jleathru@odu.edu, yshen@odu.edu, msosonki@odu.edu, maudette@odu.edu

INTRODUCTION

The current academic climate makes it hard to sustain niche programs. Universities are anticipating a large drop in enrollment starting in 2025 due to drastic drops in birth rates following the recession in 2007 (Kline 2019). Numerous universities have started to announce the elimination or consolidation of degrees and departments, starting prior to COVID-19. For the sustainability of such programs, no matter the strength or merits of the program, it becomes beneficial to align these programs with well-established programs. This allows the sharing of resources to include personnel and available course offerings. Modeling and simulation (M&S) engineering as an undergraduate program falls under this category.

Old Dominion University (ODU) chose to create the first undergraduate program in 2010. The program has graduated very high-quality students who have gone onto top tier graduate programs and high paying careers both in industry and government. Employers have greatly valued the characteristics of the graduates making them highly sought after, often receiving multiple job offers very early in their senior year. However, the program has suffered the common pitfalls of niche programs, primarily in the form of insufficient advertising/recruitment to properly educate the community about the program. This results in constantly worrying about state requirements for a program, in this case the State Council of Higher Education for Virginia (SCHEV) requires enrollment of 36 full-time equivalents (FTEs) and 9 graduates per year (the existing MSIM program at ODU is 32.9 and 9.2) (State Council of Higher Education for Virginia 2021).

But the university recognized the merits of the program and requested a proposal to enable continuance in a viable form. A "home" for M&S engineering was selected within the existing computer engineering degree at Old Dominion University, an established program that already exceeds the SCHEV requirements, removing the SCHEV concerns. This takes advantage of the strong computational engineering focus of the existing program to benefit the computer engineering program while providing the M&S engineering students a stronger grounding in engineering with a clear application domain. The result is that students get a computer engineering degree with a M&S engineering major. The curriculum was then developed so that graduates satisfy computer engineering requirements while providing a depth of knowledge in M&S. Students entering the university in the fall of 2021 may take advantage of the new curriculum. A description of the curriculum and how it fits within a Computer Engineering degree program is presented in (Leathrum, et al., 2021).

This paper takes a step deeper, discussing the decisions made in developing the content of the four core M&S courses retained in the new major. The challenges of meeting university requirements, computer engineering requirements (both imposed by ABET and by ODU's department) and meeting the basic ToK requirements of the M&S community are addressed by the individual core courses. For each course, the M&S content removed and added is presented with a rationale. In addition, the placeholders to address external university and computer engineering requirements are identified. The overall curriculum is first presented followed by a discussion of the development of each course. This is followed by discussing how other content can be covered by elective courses.

M&S ENGINEERING MAJOR CURRICULUM WITHIN A CpE DEGREE PROGRAM

Whereas the original M&S engineering curriculum provided a strong balance between M&S, mathematics, and software, the new curriculum must balance more components. In addition to the previous content areas, computer engineering and electrical engineering are now included. **Error! Reference source not found.** presents the core for both computer engineering and M&S engineering, highlighting the commonality. The course topics are color and symbol coded to illustrate the balance. Common engineering courses are not shown where mathematics (Calculus I and II and Differential Equations) are a focus along with the sciences (Physics and Chemistry). The courses are heavily computationally focused to meet the needs of a computer engineering program and to support Virginia's Tech Initiative Program (News @ ODU, 2019).

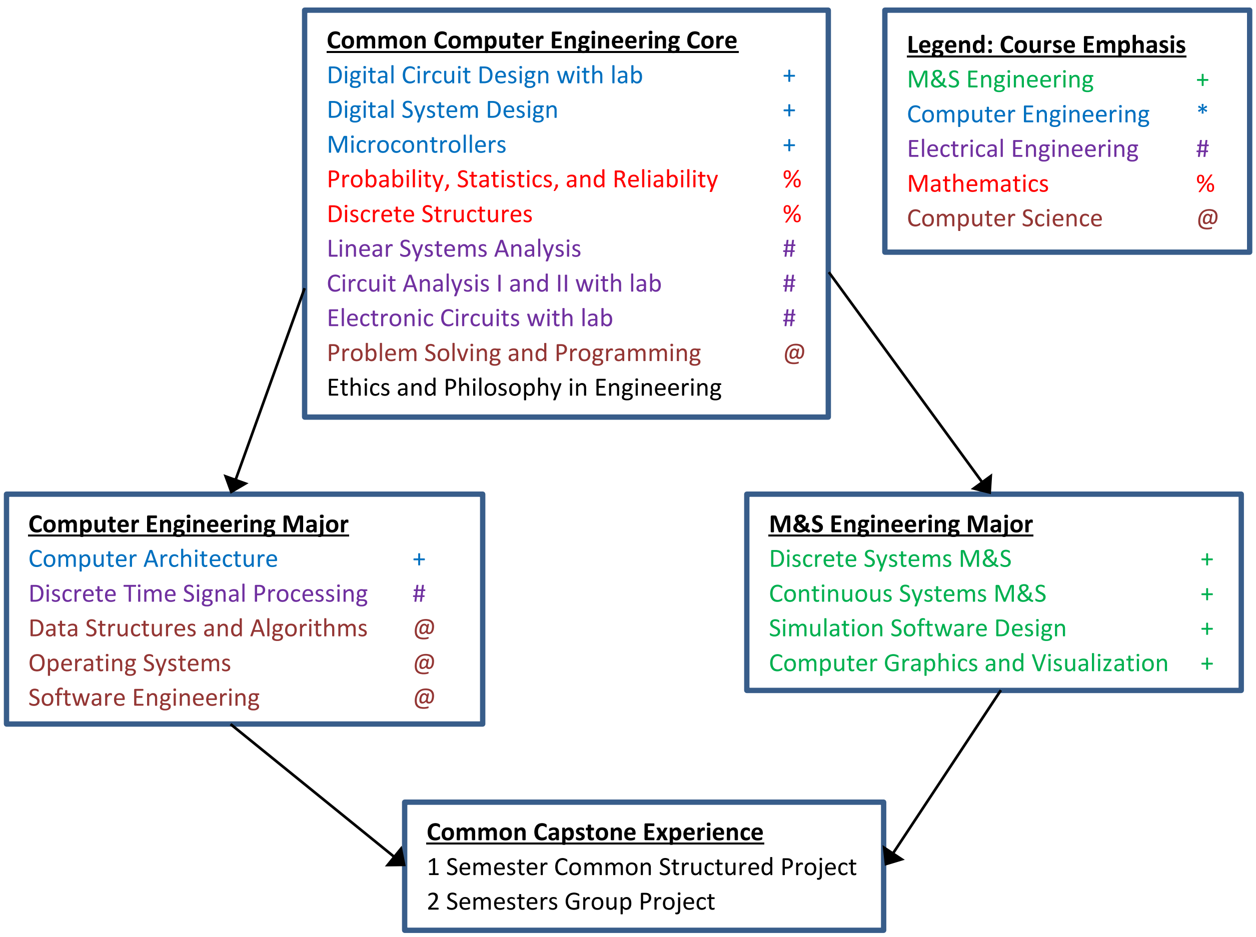

Figure 1. Structure of CpE and M&S Engineering majors.

Of note compared to the previous M&S based curriculum is the elimination of courses, an M&S intro course, a modeling course, and a simulation analysis course. The latter two will be offered as elective courses. The only content lost from the introduction course is Monte Carlo simulation. As the courses are further developed, space to include Monte Carlo simulation will be found as it is unacceptable for graduates not to experience the three main paradigms, discrete, continuous, and Monte Carlo. Basic discrete and continuous modeling lost from the modeling course will need to be included in the Discrete and Continuous Systems courses. In addition, the Discrete Systems course will require more analysis than previously covered to ensure sufficient coverage in the core curriculum.

Compression of the curriculum to fit within a computer engineering curriculum required elimination of three M&S labs, namely for the Discrete Systems, Continuous Systems and the Simulation Software Design courses. The Computer Graphics and Visualization course previously could have justified a lab component, so all these courses have a significant hands-on component that will need to be integrated into a three credit course.

A benefit to the M&SE major is the inclusion of several courses and concepts from the CpE program. Course additions of note include Linear Systems and Microcontrollers. Linear systems has long been considered missing in the previous curriculum and is welcome in the new program. Microcontrollers offers a hardware component not available previously. However, students frequently found themselves building hardware, especially in the senior capstone course, where they could encounter medical manikins, autonomous vehicles, and flight simulators. In addition, students will be introduced to some modeling concepts prior to their M&SE courses, such as finite state automata in Digital Circuit Design and Y-Chart modeling concepts in Digital System Design. They will also have an in-depth exposure to computer simulations in the form of circuit simulators for both digital and analog circuits.

The curriculum for Modeling and Simulation Engineering major under Computer Engineering was presented to the Industrial Advisory Board (IAB) of the Department of Computational Modeling and Simulation Engineering at its meeting in Spring 2021. The IAB members expressed disappointment on the upcoming discontinuation of the Modeling and Simulation Engineering degree that serves the needs of various industries locally and nationally and conveyed their concerns about the new major’s ability to meet industry needs. In particular, they were very concerned about the conversion of the analysis course from a required course to an elective course, since analytical skills are critical and highly sought-after in many industries. They understood the tight restrictions on the program, that core analysis content will be moved into other courses and that students will be encouraged to take the course as an elective. But they felt this content was a unique and valued skillset of our undergraduates. The IAB reviewed the proposed curriculum and provided feedback, primarily in terms of desired analysis content, to the ODU M&S team prior to the design of new course content in July 2021. This feedback has been integrated into the individual course development presented below.

NEW M&S ENGINEERING COURSE DEVELOPMENT

The four new courses in the M&S Engineering major discussed here all had previous versions in the M&S Engineering degree program. However, each course requires some significant modification to accommodate:

- Required material from the courses lost from the original degree program,
- Lost laboratory experience,
- Knowledge from the CpE courses the students are taking, for instance the addition of linear systems,
- New direction in course examples to take advantage of CpE prerequisite knowledge, and
- Addition of university requirements that were previously covered by courses the students are no longer taking.

The last bullet reduces the time available to cover M&S content, and thus has a negative effect on the core content. The Simulation Software Design course will be required students to analyze the broader impacts of simulation in a global, economic and societal context, albeit it will be applied to M&S, it will be on the periphery of the course content. The Computer Graphics and Visualization course will now have added content on communication and information literacy. There is hope that this content can be integrated into their capstone experience (where it was covered in the previous degree), but that approval will require time.

Each course will be discussed as to how the above modifications impact the course.

Discrete Systems M&S

Adaptations to the previous Discrete Event Simulation course for fusion of the program with CpE as a Discrete Systems M&S course fall under four categories:

- A consolidation of both the 3-credit classroom course and 1-credit into a single 3-credit classroom course.

- Further discussion on discrete event modeling. This ensures curriculum coverage of material lost from the modeling course, but also allows discussion of models the students are already familiar with from CpE courses, in particular finite state automata.
- A renewed emphasis on analytical aspects of M&S to ensure adequate coverage in the curriculum core. Students will be encouraged to take an elective course on M&S analytics to provide greater depth.
- An adaptation of the course content to emphasize use-cases that resonate with CpE students, including digital circuit simulation.

As a result of dropping the lab, the new course necessarily encompasses lab exercises as homework assignments. The current lab course currently offers a series of practical assignments based on commercial software tools such as Arena (Rockwell Automation, 2022) and Simio (Simio, 2022), involving implementation of common modeling constructs (queues applications such as Printed Circuit Board assembly lines and emergency departments), input data modeling, simulation experiment design, output analysis, as well as verification and validation. Many of these lab assignments will be folded into the new integrated course. Secondly, the discussion on discrete event modeling requires modification. The current course emphasizes queuing models as a high-level model of a system, an introduction to state machines to illustrate the changes of state in the system, and significant time on event graphs as an appropriate model of the event scheduling worldview leading nicely into a software model used in the Simulation Software Design course. The third refinement will center on providing students with expanded analysis content. This enhanced section will borrow from the current analysis course. The implications include strengthening of sections on input modeling, random number generation, output analysis, variance reduction, experimental design as well as a discussion of verification & validation. Fourth, in-class examples and assignments will build on DES applications that are a precise fit for students with a background in Electrical Engineering (EE) or CpE. These EE/CpE-centric scenarios include digital circuits, semiconductor fabrication, and computer networks.

A compelling use-case is a *digital circuit* comprised of gates, either combinational (NOT, AND, OR, etc.) or sequential (latches and flip-flops). Gates are characterized by internal state changes, driven by logical events, simulated using discrete events. When modeling the timing of combinational circuits, each gate requires an associated delay model. A simple event graph model (Buss, 1996; Schruben, 1983) for an AND gate is illustrated in Figure A. Creation of individual gate models and then interconnecting into a combinational circuit provides students an understanding of digital circuit simulation software utilized in prior courses. Extending to sequential circuits requires more sophisticated digital modeling as a result of the inclusion of clock signals controlling the timing of when outputs become available. Students will require a better understanding of simultaneous events as the introduction of a clock event to numerous gates results in many state variables changing at the same simulation time, but all requiring their output to be a function of the state prior to that simulation time. A simple D flip-flop event graph model is shown in Figure B. Again, by constructing models of these gates, the students can construct sequential circuits by interconnecting a combination of combinational and sequential gates. However, when implementing in the classical discrete event software tools such as Arena and Simio, the students will observe the limitations of those tools applied to digital circuits. A simple ring counter will not behaver correctly because the underlying model for simultaneous events in the tools (generally to execute the events in the order they were scheduled) results in erroneous behavior. Thus, a discussion is necessary addressing the problem through either more detailed modeling of the gates (necessary for the general DES software tools) or through modifying the behavior of the simulation executive.

DES is commonly used in manufacturing. To tailor an example to computer engineering, *semiconductor fabrication* will be discussed. The process is characteristically comprised of several operations, such as oxidation/diffusion, photolithography, ion implantation, and so on, where a DES approach can be leveraged to alleviate bottlenecks in the process and optimize decision-making in a mass-production setting. This use-case mirrors myriad industrial production applications that exploit DES as the foundation of a simulation-based approach to logistical improvements. The inclusion of semiconductor wafer fabrication scenarios in DES instruction can potentiate the adoption of this course as a useful complement to an EE or CpE background that emphasizes microelectronics design.

Finally, another use-case that will make the course especially relevant to CpE students is *computer networking*, whose performance is characterized by internal states coinciding with distributed hardware components or software layers. This area is extremely rich in possibilities, ranging from optimizing network protocols and identifying bottlenecks, to cybersecurity simulation designed to anticipate and thwart malevolent intruders while enabling real network performance to degrade gracefully rather than collapsing completely. Many networked resources behave as, and are readily modeled with, queues. Packets of data can be thought of as being sent from one node to another, where concepts of inter-arrival time and service time are still relevant to such as discussion. What-if questions will be explored as well, motivating students to apply a DES framework to deploy computing resources optimally.

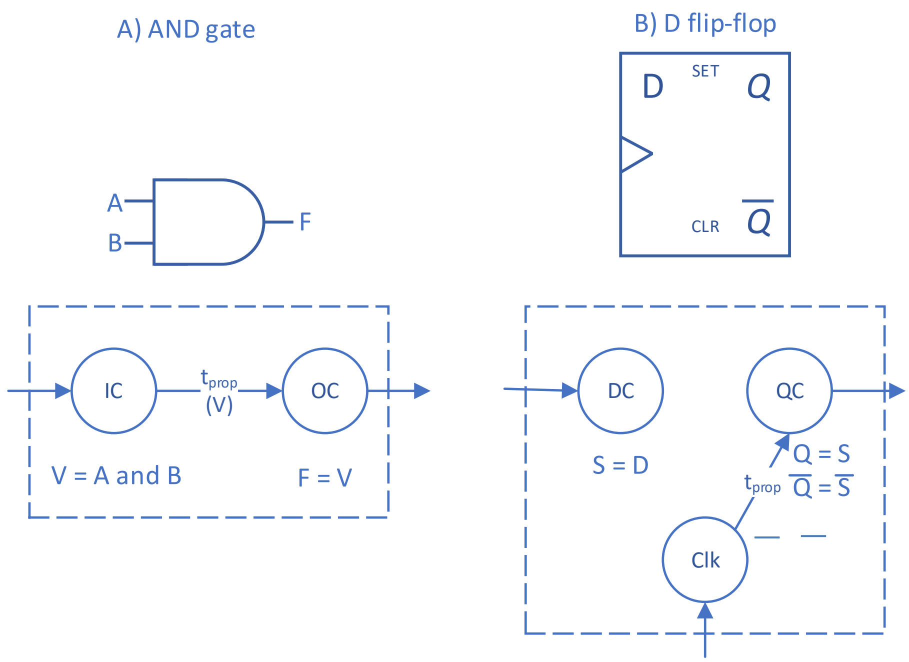

Figure 2. Example simulation blocks implementing digital circuit gates. a) is a combinational gate while b) is a sequential gate. b) is simplified to ignore setup and hold times.

Continuous Systems M&S

The content of this course is fundamental to the entire modeling and simulation field as continuous-type systems arise in all the engineering disciplines and in many pivotal areas of society, such as economics and sociology. Students majoring in modeling and simulation engineering must be able to model systems in different areas, to apply appropriate simulation methods that solve these models on different time scales, and to analyze the obtained results.

Presently, these course outcomes are achieved by first introducing students to the real-world problems that give rise to continuous systems followed by teaching them how to model these systems using the knowledge of physics and mathematics along with the notion of the feedback mechanism. The course teaches students to simulate the obtained models using mathematical software environments, such as MATLAB and Simulink (Mathworks, 2022). A set of laboratory exercises has been designed towards achieving their proficiency in these software packages. The analysis part of the course relies heavily on the mathematical underpinnings of a variety of numerical integration methods and on computer arithmetic and representation of real numbers.

Implemented as a new course within computer engineering, the Continuous Systems M&S course will incorporate the general outline and structure of the present course with the notable changes in its two major parts, model representation and solution analysis. To expose students to a broader set of model representation methods, such as bond graphs and stock-flow diagrams---presently covered in a separate modeling course---the new course will show how to represent the models in a variety of disciplines using tools most widely used in that discipline. For example, continuous models in economics will employ stock-flow diagrams, electrical circuit models will start with circuit-specific modeling in Spice, while bond graphs may be applied to represent more general systems from which the definition of the state-space models follows. Since the CpE students have a knowledge of vectors and linear systems, the presentation of some analysis topics, which currently start with the explanation of vector manipulations, will be accelerated as will be evidenced, e.g., in the presentation of stiff continuous systems. The freed lecture time will be spent on teaching students how to conduct continuous simulations using Simulink and MATLAB to make up for the absence of laboratory hours, and a laboratory-like experience will be integrated into assignments.

Student projects will remain the highlight of the course organized in their present structure: The topics are given to the students individually or in small groups leaving them about a month to research and implement an assigned project. The project culminates in student presentation during the last lecture and in project reports. In addition to the contents and course-specific requirements, both the presentations and reports are graded based on the format and style consistent with general University scientific and engineering guidelines. A major change in the project course activity will come from the type of project domains. In the new ECE course, these domains will be mainly from the fields of computer and electrical engineering to leverage the knowledge that students obtain as the part of their ECE core courses. As a consequence, it will take them less time to get familiar with the project domain at hand, hence the emphasis will be shifted towards learning more of Simulink for the modeling and simulation of their projects.

Simulation Software Design

A common theme in this paper is clearly more to teach, less time to do so, less opportunity to experience. That is very evident in this course. The course highlights the importance of understanding simulation software, not just the tools (Schriber, et al., 2016). As the lynchpin of the software experience, a necessity to both satisfy the computer engineering curriculum and to support Virginia's Tech Initiative, this course is taking on even more software content. There was already a push to have this course be the primary exposure to object-oriented design due to dissatisfaction with other object-oriented courses and the natural presentation of object-oriented concepts in simulation (simulation did spawn the object-oriented paradigm (Black, 2013)). It was also identified that insufficient effort was spent on the process interaction worldview resulting in the need to introduce either threads or coroutines. Initial efforts have introduced implementing the process interaction worldview with threads since they are standard in C++, but with the inclusion of coroutines as standard in the language, they are being introduced as well, in fact as the focus being a much better choice for performance reasons. This is also the only place in the curriculum where threads and coroutines are guaranteed to be covered (threads are covered in more detail in the distributed simulation elective) and current trends are making these concept crucial to graduates. For example, coroutines are considered by many to be the appropriate approach to event programming. Many of these concepts are discussed in (Leathrum, et al., 2017).

A planned time saver is that previously all application discussion beyond the basic single server/single queue model required an introduction to the concept. However, computer engineering students will have a rich background in digital circuits and networks allowing examples to be introduced with no explanation. While the discrete and continuous courses will still require demonstrating the breadth of application beyond the digital and analog circuit examples, the software course can rely on circuit examples. But that is not anticipated to be enough. Students will be expected to spend a significant amount of time outside of structured time to get the necessary hands-on experience to become proficient. Support will be provided by giving unstructured "lab" time where students can collaborate and have access to a TA. The external requirements placed on the student are now discussed.

Homework in the original course focused on the development of a simulation executive based on the event scheduling worldview. Normally six assignments, most with a high level of complexity, building on each other until they have a working simulation executive. A first assignment gives them a home-grown simulation executive library where the construct a simulation application, both using the event scheduling worldview and the process interaction worldview to introduce them to the different software models. They then begin developing their own library to replace the provided one. Initial design and implementation involve working on an interface to the simulation executive, studying and implementing various approaches, including functional and object-oriented (inherited, static classes and singletons) allowing discussion of their relative ease of use and implementation. The storage of events presents the opportunity to explore a rich set of data structures and discover their relative merits. They implement the internal event list using linked lists, heaps, calendar queues, and binary sort trees allowing them to observe their relative performance. The new course retains this basic structure but reduces the breadth of the student's experience. They still will experience building an application in both the event scheduling and process interaction worldviews. However, the following assignments will be reduced. They will experience implementing a single software interface. The data structure experience will remain, but due to time constraints, they will be limited to just linked lists and calendar queue. The calendar queue is used to demonstrate the reuse of the previous linked list and to show the marked performance improvement.

Labs in the original course provided students the opportunity to experience the development of a simulation library based on the process interaction worldview. They developed a tool for simulation development at the application level, given a simulation executive library based on the event scheduling worldview. They created a set of blocks similar to those in Arena (i.e., Create, Process, Decide, Separate, etc.). Internal to the blocks they utilize the event scheduling world view for implementation purposes, but then interconnected (in software, not visually) the blocks to create complex process interaction simulations, such as simulating an airport. This experience is lost as an end-to-end effort with the removal of the lab. However, through a couple of homework assignments the students will build blocks representing digital gates, both combinational and sequential. In this manner, students can then construct digital circuit simulations in software and get appropriate timing diagrams.

Figure demonstrates simple digital circuit gates modeled as event graphs. One homework will now involve developing a set of combinational gate blocks (AND, OR, NOT, XOR) in software and another homework for a set of sequential gate blocks (DFF, JKFF). Each homework will involve constructing a circuit and simulation. Digital circuit simulation was attempted in an early version of the lab course but changed to an Arena like experience due to the discomfort students had with digital circuits. Now with a very strong background in digital circuits, students will be able to create simple gate models and discrete event simulations for each gate to be interconnected into a complex circuit, to include counters, ALUs, etc. Software is currently in development to read a simple netlist circuit representation to instantiate and interconnect the appropriate gate simulations to create a circuit simulation. In this manner, students can experience a reusable application tool development (minus the graphical component) with as little time commitment as possible in just two homework assignments.

A final note is that course development is struggling with the inclusion of the University Requirement to analyze the broader impacts of simulation in a global, economic and societal context. This was previously accomplished in a senior capstone project where the discussions could be drawn from the actual project the students were working on. While covering the topic in the broad M&S domain is not difficult where the implications of the model and resulting simulation results can be considered, doing so in the context of simulation software design is proving more difficult where the model is already assumed and results are not considered except software verification.

Computer Graphics and Visualization

Major revisions of the Computer Graphics and Visualization course are required to address the following curriculum requirements.

1. Key concepts and theories covered in the course Discrete Time Signal Processing, which is a prerequisite for the Electrical and Computer Engineering Senior Design Project course, but not included in the M&SE major curriculum. It was decided that these key concepts and theories will be covered in Computer Graphics and Visualization, such as sampling and discrete Fourier transform.
2. Coverage of content on communication and information literacy. The Computer Graphics and Visualization course already has a heavy component on written communications in the form of software development reports. Content needs to be added to cover more information literacy.

The current offering of Computer Graphics and Visualization provides a practical treatment of computer graphics and visualization with emphasis on the usage of industry standard application programming interface (API) libraries for modeling and simulation applications. It introduces computer graphics fundamentals, including mathematical foundations, rendering pipeline, geometrical transformations, 3D viewing and projections, lighting and shading, texture mapping, etc. It teaches OpenGL programming for developing interactive visualizations for modeling and simulation applications. The Unity game engine is utilized to illustrate advanced concepts and techniques. Software architectures for modeling and simulation and visualization principles based on perception are covered in depth with case studies.

Fortunately, computer graphics and discrete time signal processing (or digital signal processing as it is more commonly referred to as) are highly related. The following list only highlights a few aspects of the close relationship between computer graphics and digital image processing.

- The overall objective of a computer graphics pipeline is to transform a mathematical representation of 3D worlds into 2D digital images that can be shown on display devices. The outputs of computer graphics pipelines are digital images, which are one of the most important types of digital signals.
- Digital images are also heavily used as inputs to the computer graphics pipeline as part of the modeling process, e.g., digital images are used as textures to enrich the visual appearance of 3D models with only a moderate increase of computational complexity. Similarly, digital images are used as textures for other purposes, such as lighting and shadows.

- Digital signal processing techniques such as filtering are utilized as part of the computer graphics pipeline to increase the rendering quality and efficiency.

To incorporate key concepts of digital signal processing, e.g., sampling and Discrete Fourier transform, some existing content in Computer Graphics and Visualization need to be removed or curtailed. The following two course modules are being considered to undergo major revision.

- **Color Models and Application of Color in Visualization**. This module covers various aspects of color models and principles for using proper colors in various applications. Topics include trichromacy theory (cone sensitivity functions, luminance sensitivity function, primary colors, color blindness), color measurement (color primaries, color spaces, conversion between spaces (CIE, XYZ, RGB), chromaticity diagram), opponent process theory (luminance channel, red-green channel, yellow-blue channel), properties of color channels (spatial sensitivity, motion sensitivity, forms), color appearance (color contrast, constancy, saturation, monitor surrounds), and application of color in visualization (color specification interfaces, color for labelling, color sequences for data maps, color reproduction). While these topics are very interesting, some of the topics focus more on principles of visualization, using psychology and neurology, e.g., the opponent process theory, and they are not part of the graphics pipeline, which is the core of the course. A majority of these topics will be eliminated, and the remaining topics will be condensed.
- **Discrete Techniques**. This module covers various discrete techniques utilized in the graphics pipeline with emphasis on texture mapping. Topics include digital buffers, digital images, bit and pixel operations, mapping methods, texture mapping, texture mapping in OpenGL, texture generation, environment maps, and compositing techniques. This module covers texture mapping in great detail, e.g., various coordinate spaces used by texture mapping (object space, screen space, and texture space), and projections for parametric surfaces and arbitrary surfaces. This module also contains a major programming assignment in which students are asked to manually generate 3D objects and their corresponding texture coordinates in code. In most advanced computer graphics and visualization applications, such as games and modeling and simulation applications, modelers (e.g., artists) are responsible for 3D modeling, e.g., creation of models of advanced weapons and vehicles using 3D modeling software (e.g., Maya). Texture mapping is part of 3D modeling process carried out by the modelers. Application software developers have much less need to determine the texture coordinate manually in their code; they usually just use the texture coordinates directly, which are generated by the modeler and stored in a 3D model file. Therefore, the plan is to reduce significantly lecture content and programming requirement on texture coordinate calculation to make resources available to cover the new content.

The new content to be introduced includes the following three components.

1. **Sampling**. Sampling is a key concept and technique of digital signal processing to convert and represent continuous signals (e.g., sound) in digital form to be used by digital computers. The fundamental principle for sampling is the Nyquist-Shannon sampling theorem, which states that a continuous signal with a bandwidth B can be fully reconstructed from discrete signals sampled from the said continuous signal using a sampling frequency f_s if $f_s > B$.
2. **Discrete Fourier Transform**. The Fourier transform is a mathematical transform that converts a function in time or space domain into a function in temporal or spatial frequency domain. For instance, a unit pulse function in time domain will be transformed into a *sinc* function in temporal frequency domain. The physical meaning of Fourier transform is that the original signal in temporal or spatial domain can be restructured by superimposing a series of sine functions whose frequency and magnitude are specified by its Fourier transform in frequency domain. The discrete Fourier transform (DFT) converts equally spaced samples of temporal or spatial signals into equally spaced samples in the frequency domain. It is often said to be a frequency domain representation of the original signal. The DFT of the convolution of two functions (e.g., an image and a filter) is the product of their corresponding DFT. This property makes DFT very useful in signal sampling and reconstruction.
3. **Communication and Information Literacy.** The current offering of the course already has an emphasis on written communications in the form of project reports for programming assignments. The project report must describe software design and generate software documentation using industry standard tools, e.g., Doxygen (Doxygen, 2022). When considered on combination with the oral and written component of Continuous Systems M&S, the content is properly covered. More coverage on information literacy will focus on information related to the content covered by this course, e.g., strategies for finding and evaluating information, various Internet sources, and how to properly utilize the information.

Sampling and discrete Fourier transform will be integrated to the Discrete Techniques module as both are integral parts of discrete techniques, utilizing the resources (lecture time and assignments) made available by elimination of major topics in the module Color Models and Application of color in Visualization and the detailed coverage of texture coordinates in the Module Discrete Techniques. The increased coverage on communication and information literacy will be distributed across the entire semester to focus on different types of information, e.g., theoretical questions, software tools, and software development issues.

FUTURE WORK

There are two primary areas of future work to complete the curriculum and consider its external impact, development of M&SE elective courses and consideration of the impact on the graduate program.

The desire is to transition most of the existing elective courses into the new program. While general M&S courses such as distributed simulation and gaming, are broad enough to transition, many elective courses are application domain based, requiring an assessment of their applicability. Cyber security courses will transition easily, but courses such as those in transportation need to consider how they are tailored to CpE students to ensure their viability. In addition, versions of the existing modeling course and analysis course will be developed as electives. The analysis course in particular will be highly recommended to students as an elective based on industrial board feedback, a hard sell as other elective options are seen as more attractive.

The impact of the new courses on the graduate program will be huge. The graduate program is not targeted to computer engineers, supporting a broad background of applicants. Introductory coursework gets students up to speed with the graduates from the BS in Modeling & Simulation Engineering in terms of discrete event and continuous simulation and software design. Previously, the modeling course and analysis course taught at the undergraduate level were made available to graduate students not graduating from the undergraduate program. With those courses potentially taking on a more computer engineering slant, partly to attract computer engineers to take the classes, this approach must be reconsidered. In addition, the program has always supported a BS/MS option to qualified students. The transition of students from the new undergraduate program to be graduate students must be reviewed.

CONCLUSIONS

Do more with less. While this can be a negative takeaway from this discussion, it can also be seen as an opportunity. The difficulty is making the hard choices as to what should remain from the original curriculum. This paper laid out the lost content from prior courses no longer required and new required content. Approaches to address these problems were presented with resulting changes in course content. While this has been a difficult transition, the authors are confident that the resulting coursework will still produce highly qualified M&S engineers, and potentially even stronger from a technical viewpoint.

Issues remain, though not within the course content. The primary issue is still recruitment, a question asked when the curriculum was presented at the Winter Simulation Conference 2021 (Leathrum, et al., 2021). The number of engineering students is anticipated to drop nationwide (Kline, 2019), making it more competitive to attract students. Now a student interested in M&S must also be interested in computer engineering. The prior program was twice as attractive to female engineers at ODU than other engineering programs. But now those female students must also want to do computer engineering and may opt for an alternative degree such as computer science.

Of particular benefit to the program is the ability to attract students from the community colleges. Virginia community colleges have long been considered as a cost-effective quality option for higher education. Many of the best students in the original program came from the Virginia community college system. However, those students required three years to complete rather than the minimum of two years for other engineering programs. This was a result of the community colleges not teaching the sophomore level courses for the program, either from lack of instructors or students. Now those students will be able to complete the M&SE major in two years as all four courses discussed here are at the junior or senior level. Many community college students chose not to do our program for that reason, so the hope is an increase in recruitment success from the community colleges, offsetting potential losses discussed above.

REFERENCES

Black, A.P.. (2013). Object-Oriented Programming: Some History, and Challenges for the Next Fifty Years. *Information and Computation*, *vol. 231*, pp. 3-20.

Buss, A. (1996). Modeling with Event Graphs. In *Proceedings of the 1996 Winter Simulation Conference*.

Doxygen. (2022). *Generate documentation from source code*. https://www.doxygen.nl/index.html.

Kline, M. (2019). The Looming Higher Ed Enrollment Cliff. *Higher Ed HR Magazine* Fall 2019.

Leathrum, J.F., R.R. Mielke, A.J. Collins, and M.A. Audette. (2017). Proposed Unified Discrete Event Simulation Content Roadmap for M&S Curricula. In *Proceedings of the 2017 Winter Simulation Conference*.

Leathrum, J.F., Y. Shen, O. Gonzales. (2021). A New M&S Engineering Program with a Base in Computer Engineering. In *Proceedings of the 2021 Winter Simulation Conference*.

Mathworks. 2022. *Simulink*. https://www.mathworks.com/products/simulink.html.

News @ ODU. 2019. ODU Signs Tech Talent Investment Agreement. https://www.odu.edu/news/2019/11/tech_talent_investme#.YfBcUPhOm00.

OpenGL. 2022. *The Industry's Foundation for High Performance Graphics: From Games to Virtual Reality, Mobile Phones to Supercomputers*. https://www.opengl.org//.

Rockwell Automation. 2022. *Arena Simulations Software*. https://www.rockwellautomation.com/en-us/products/software/arena-simulation.html.

Schriber, T., D. Brunner, and J. Smith. (2016). Inside Discrete-Event Simulation Software: How it Works and Why it Matters. In *Proceedings of the 2016 Winter Simulation Conference*.

Schruben, L. (1983). Simulation modeling with event graphs. *Communications of the ACM*, vol. 26, no. 11, pp. 957–963, 1983.

Simio. 2022. *Simio Forward Thinking*. https://www.simio.com/.

Unity. 2022. *The Platform of Choice for Multiplayer Hits*. https://unity.com/.