Introducing Formalized Systems Engineering to the Jacobs School of Engineering

January 2019

Recommendations from the Systems Engineering Visioning Committee
# Table of Contents

Executive Summary.............................................................................................................3

I. What are the commonalities in the systems engineering project case studies, which are best used to select the disciplinary core?...............6

II. How will we distinguish our program?.................................................................8

III. What is the recommended degree program?......................................................11

IV. *What courses should make up the curricula, and what resources are required?*.................................................................12

V. *What research topics will inform the evolution of our curriculum?*..........................14

*Appendix A: Summary of the Dean's Council of Advisors Recommendations*.................................17

*Appendix B: Summary of Jacobs School Corporate Affiliates Program Recommendations*.................................19

*Appendix C: Summary of Systems Engineering programs across selected institutions*.................................20

*Appendix D: Gap Analysis and Proposed ECE Systems Engineering Curriculum*.................................22
Executive Summary

The UC San Diego Jacobs School of Engineering is committed to launching one of the first modern systems engineering programs designed to prepare our graduates to be leaders and innovators in tomorrow's rapidly advancing, and globally connected organizations. This transformation has already begun and will continue to accelerate over the next decade as industry, government and academia take a systems approach to solve the planet's grand challenges. Industry is leading a revolution in complex, massively distributed, data-driven systems that rely on new data streams, analytics, and machine learning and modeling to constantly evolve and improve, during ever-shorter iterations. In our globally-connected socioeconomic ecosystem, the stakes are high, and our engineers must be able to manage known risks, and reduce the impact of unintended consequences.

Industry partners from multiple industry sectors are demanding a new breed of systems engineer and systems engineering education. Traditional systems engineering processes, built around the relatively long and controlled design cycles in aerospace and defense, will need to evolve to satisfy today’s complex and dynamic environment.

The Jacobs School is known for providing strong engineering fundamentals in a collaborative research environment. Our bold vision is to couple the discipline-specific education offered in our engineering departments with a relevant and modern complex systems curricula. Our goal is to teach our students how to think differently: how to ask the right questions; see the big picture as it evolves; and embrace ambiguity as they make decisions in uncertain environments. Systems engineering students will learn how to create cognitive models to visualize, intuit, and innovate complex systems, and how to orchestrate and rapidly integrate new, complex components. They will learn about iterative and agile engineering design. Students in the program will practice business, leadership, ethics, and teaming skills as they work together on practical systems projects both in class and onsite during Co-ops with industry partners.

At the heart of our systems engineering program will be a focus on closed-loop system design built from the ground up to constantly evolve and improve. We intend to practice this paradigm as we design, deliver, and continuously improve the systems engineering education program. Our stakeholders are our students, our industry partners and the academic community. In preparing this White Paper, we have talked to more than 100 industry executives, surveyed hundreds of students and evaluated systems engineering programs at our competing universities. As we launch and grow our systems engineering program, we will continue to seek feedback from our stakeholders, annually evaluate our results, and use these inputs to improve the education we deliver.

The Jacobs School of Engineering's Systems Engineering Visioning Committee was formed to develop a Jacobs School vision for an educational program to educate entry-level systems engineering leaders for the 21st century. The committee consists of faculty representations from
all six departments, Director of the Gordon Center, the Assistant Jacobs School Dean, and Jacobs School Staff:

- Bioengineering: Todd Coleman, Sandrine Javorschi-Miller
- Computer Science & Engineering: Ryan Kastner
- Electrical & Computer Engineering: Truong Nguyen (Chair), Rick Gessner, Todd Hylton, John Sanford, David Whelan
- Mechanical & Aerospace Engineering: Jorge Cortes
- NanoEngineering: Ert Cubukcu
- Structural Engineering: Hyunsun Alicia Kim (Vice Chair)
- Gordon Center: Ebonee Williams
- Jacobs School Dean Office: Denine Hagen, Kate Balderston

The committee solicited input from our industry partners on the case studies representing Systems Engineering and received ten case studies from eight industry partners. These case studies were analyzed by the committee to best develop the curriculum. We also received important recommendations from the Jacobs School of Engineering Deans Council of Advisors and Corporate Affiliates Program executives on critical skills that a “complete” engineer should have. These include:

- Ensure Students Have a Depth Area
- Emphasize Real-World Experience
  - Hands-on project: Customer, Architecture, Integration, Prioritization, Risk Management, Cost benefit tradeoffs
  - Co-Op program
- Teach Modern Systems Engineering Methods and Tools
  - Operational systems test and validation
  - Data-driven systems modeling and optimization
  - Software engineering, including network patterns and interface design
  - Cybersecurity vs performance tradeoff
  - Reliability and Maintainability
- Develop Leadership Competencies
  - Teamwork, communication, design, product, marketing, ethics

Five subcommittees were formed to address several issues ranging from curriculum development to the best option for offering a degree program (undergraduate vs. graduate), to research programs to support the evolving and future of systems engineering and faculty recruitment:

a. **Subcommittee 1** (Cubukcu, Coleman, Javorschi-Miller): What are the commonalities in the case studies, and which are best used to select the disciplinary core?
b. **Subcommittee 2** (Cortes, Kim, Kastner, Sanford): How will we distinguish our program from others? What name best describes the program and helps communicate the uniqueness of what we are offering? What are the focus domains that reflect/serve our regional and statewide industry?

c. **Subcommittee 3** (Hagen, Nguyen): What is the recommended degree program(s)?

d. **Subcommittee 4** (Gessner, Hylton, Whelan, Williams): What are the Courses that should make up the curricula: core systems curricula, the domain courses/electives, leadership & ethics. What are the resources required to develop the new courses?

e. **Subcommittee 5** (Hylton, Sanford, Kim): What research topics in the Jacobs School should inform the evolution of our curriculum, and guide future faculty recruitment?

The following sections (I-V) are reports from the five subcommittees, which will be used to develop a full program proposal.
I. **What are the commonalities in the systems engineering project case studies, which are best used to select the disciplinary core?**

We collected responses and case studies from industrial partners and alumni in order to establish a list of desired systems engineering skills in a UC San Diego-trained engineering student going through the proposed Jacobs School-wide degree program. We have reached out to over 20 people from more than 15 different companies including Qualcomm, HP, Illumina, HRL, General Atomics, Hewlett-Packard, BASF, Pfizer, Raytheon, Viasat, Northrop Grumman, ASML, and Corning. Several skills are identified in addition to the core analytical and engineering skill, for an engineer to succeed in today’s industrial environment. They are summarized below.

a. **Confident individuals:** It is the general belief that engineers, while they are extremely competent at technical challenges, may feel uncomfortable in social environments. The proposed degree program should explore ways to address this important aspect.

b. **Learning from peers:** The complex work environment heavily relies on teamwork, which involves working in harmony with others in the group, while also demanding quick learning skills to benefit from the expertise of the peers. As a part of this, an emphasis should be given to learning how to give and receive constructive engineering design feedback. Team projects should be implemented towards achieving this goal in the proposed degree program.

c. **Great technical and other presentation skills:** While a core engineering program prepares our students for excellence in technical skills, they simultaneously excel in both written and oral presentation skills.

d. **Systems-level thinking vs. systems engineering:** Some of the case studies emphasized the importance of systems-level problem-solving in light of the fact that conventional systems engineering process may be insufficient in overcoming the true barriers to developing the systems in certain cases. A balanced mix of both need to be incorporated into the curriculum in order to best prepare our students. A useful new resource on systems thinking has become available: “The Habit-Forming Guide to Becoming a Systems Thinker,” written by Tracy Benson, Ed.D., and Sheri Marlin, M.Ed.

e. **(Integration) Understanding of how to integrate different components of the project:** To this end, the following skills will be invaluable; learning how to verbally debate system design tradeoffs, being able to conceptualize and clearly articulate multiple alternatives, upfront awareness of end-user experience and requirements in the field.
f. **Agility and adaptation for changing requirements:** Requirements statements are rarely sufficient to arrive at an optimal system concept/design. One needs to have to take the time to understand the context and the environment in which the system development occurs and be prepared to make changes to the design objectives as needed as the project evolves.

g. **Cost requirements:** The students going through the program need to develop an awareness for the cost requirements of the systems and trade-offs as the cost may be the most important driver in most cases. The ability to choose between a perfect system vs one that does the job at the desired cost will be an important skill.
II. How will we distinguish our program?

This summary is structured with answers to the questions originally posted under the "How will we distinguish our program?" theme.

How will we distinguish our program?
To address this question, the subcommittee reviewed 12 undergraduate and graduate programs on systems engineering at top national universities (see Appendix C for a list and a summary description of each one). We also discussed various ongoing efforts from the professional community (AAIA, NASA, Applied Technology Institute) trying to shape what Systems Engineering (SYSE) of the future should look like. Our aim was to learn about what other programs do, and identify how their courses and curriculum structure address critical skills and technical knowledge. Based on their strengths and weaknesses, the subcommittee identified the following distinguishing features of a systems engineering program at UC San Diego:

- Emphasis on emerging technologies and techniques.
- Use of real-time communication tools for program management.
- Project- and team-based learning, leveraging industrial links and real engineering design problems.
- Broad coverage of traditionally non-engineering skills such as communication, team building and management, design for manufacturing, maintenance and sustainability, policy, regulations, ethics, requirement management and communication with customers.

What name best describes the program and helps communicate the uniqueness of what we are offering?
Modern SYSE should account for human in the loop, have a strong team effort component, and is in need of tools to evaluate the unintended consequences of a given design, of quantitative measures of coupling and effect on overall system behavior, and of mechanisms to detect unanticipated interactions among system components. Additional keywords the subcommittee was keen on included “complex systems”, “innovation”, “evolving requirements”, and “input from multiple sources”. The subcommittee did not gravitate towards a single option, and instead identified the following possibilities:

- Multidisciplinary Engineering for Complex Systems
- Engineering Systems
- Evolutionary Engineering
- Strategic Innovation and Development
- Innovation Engineering
- Innovation Strategies
- Complex Systems Engineering
How does our program take advantage of new tools and trends? Traditional SYSE adopts a “divide-and-conquer” approach to deal with complexity, leading to suboptimal designs, rather than an “integrate-and-conquer” viewpoint, and does not place enough emphasis on business models, life-cycle, ethics, communication with users, operation. The subcommittee identified the following tools and trends aligned with our vision for an SYSE program at UC San Diego:

- Multidisciplinary projects that engage the engineering department and business school. Current complex systems have many people and organizations providing input to the process, with requirements morphing over time; this requires the system engineer to integrate real-time inputs from multiple sources into the design process.
- Emphasis on optimization, machine learning, predictive analytics, data science, programming, cloud computing, virtualization, and visualization.
- Broad coverage of subjects such as communication, team building and management, ethics, environment, ethics and policy/law and their consideration on the engineering process and design respond to the new trends and requirements.
- Entrepreneurial activities. Patents and licensing understanding.
- Design Lab.

What are the focus domains that reflect/serve our regional and statewide industry?

- Biomedical, Medical Devices
- ML, AI, Data and Predictive Analytics
- Internet of Things
- Aerospace, Defense, Mechanical
- Smart Cities / Integrated Public Transportation Systems

The subcommittee also discussed the possibility of having two tracks in the SYSE program, each one potentially having different focus domains:

- **Track 1** would be related to cross-disciplinary program management. The focus would be on optimizing model-based systems and defining interfaces between business groups. This track would require some business training and might focus on case studies in system engineering.
- **Track 2** on virtualization and visualization. The student would focus on reducing a complex system to a manageable model and then apply techniques to refine and optimize the model and its applications.

How can we use the Collaboratories for the Digital Future to strengthen the program?
The program is very much aligned with the vision of the collaboratories, which is

1. Inspire New Research Directions
2. Enable Large Collaborative Research Efforts and Success on Center Grants
3. Facilitate Connections Across Collaboratories for Systems-Level Projects
4. Encourage Industry Participation via IP-Protected Spaces
5. Support Commercialization of Research

The program on SYSE needs to be supported by research to educate the next generation of engineers. The research challenges in systems engineering as identified by industry and professional communities are present in crossing disciplines and the unknown internal conflicts between them, indicating new research directions which can only be identified by multidisciplinary collaborations (Vision 1). The effective solutions to these challenges require large collaborative efforts at systems level, which completely align with Visions 2 and 3. It is unlikely that there would be one simple systems abstraction that could address this complex space and a thorough understanding of real systems challenges at scale necessitates a close engagement of the academic research with industry. Thus, Visions 4 and 5 form critical elements of the new systems research. These activities would in turn, be developed into research challenges (Vision 1), inspiring new research topics.

Accompanying research activities are key to the long term success of the program and align well with the vision of the collaboratories and this is discussed further in Section V. For example, we can anticipate a project team, formed by a range of disciplinary engineers (e.g. aerospace, bio, computer, civil, electrical/electronics, materials, mechanical, ...) to conduct a systems level multidisciplinary project supported by the collaboratories research activities, using the new tools in a practical project that has been conceived via the industry engagement, and conduct the research in the IP-protected space, industrialists could be given a space in the collaboratories to conduct design meetings, etc. The multidisciplinary academics can offer relevant classes to the students also in the collaboratories. The students will learn how these new tools can be used to solve a complex systems design problem collaboratively. The outcome of the projects, in turn, can be used to form a feasibility study or research challenges that can inspire new research directions.
III. **What is the recommended degree program?**

The Systems Engineering committee was asked to consider the recommended degree pathway for the systems engineering program. The committee recognizes that the Jacobs School already offers a master’s degree program for experienced systems engineers called the Architecture-based Enterprise Systems Engineering Master of Advanced Study program. The committee recommends that the Jacobs School take a multi-phase approach in launching a comprehensive systems engineering program for entry-level engineers.

a. **Phase I** would be launched quickly as a 3-4 course undergraduate concentration with Co-Op experience, which would be offered within each participating engineering academic department. These core courses would be offered as upper-division technical electives that would be open to master’s students. Courses could be cross-listed between departments as appropriate to the industry domains. This strategy would enable the Jacobs School to quickly infuse “systems thinking” throughout the curricula, ensuring that more students have access to these valuable skills and competencies.

The committee suggests that departments strongly encourage eligible students to elect to take a B.S./Co-Op/M.S. pathway as the ideal preparation. In this scenario, students would take the concentration courses in their junior and senior year, complete a Co-Op between the fourth and fifth year, and then complete the master’s degree.

b. **Phase II** of the Systems Engineering program launch would entail the creation of systems engineering minors in the participating departments. The minor would be built from the 3-course concentration core curricula, with an additional two lower division and two upper division courses added or modified to complete the minor course plan.

In the final phase of launch, the Jacobs School would create a B.S./Co-Op/M.S. pathway in which students complete the systems engineering minor followed by an M.S. degree with a course track that provides a deeper dive into domain-specific systems through technical electives.

Please see the discussion on curriculum development in the curricula section below for more information.
IV. What courses should make up the curricula, and what resources are required?

Curriculum Outcomes - The stated goal of this initiative is to create a new curriculum to augment existing program that meets the following objectives:

a. Student Objectives:
   - Offer student learning outcomes that strongly align with emerging trends in technology and practical methods.
   - Enhance employability and workforce impact.

b. Program Objectives:
   - Apply SYSE themes across all our engineering programs.
   - Integrate and reinforce SYSE concepts into the core curriculum.

c. Institutional Objectives:
   - Increased institutional competitiveness.
   - Develop a continuous sequence of public/provider partnerships.

Stakeholder Guidance - Meeting with industry leaders, business owners, professors, student groups and other stakeholders, we have collected and coalesced feedback and insights into a list of requirements. This guidance is summarized at the end of this document. Essential components include:

a. Emphasize more hands-on skills overall.

b. Increased training in modern engineering methods:
   - Scalable, real-time, distributed software
   - Adaptive, orchestrated software architectures (focus on software)
   - Data-driven decision making and analytics
   - Agile methods: planning, operations, design, cognitive modeling

c. Increased training in requirements, cost, risk management, verification, and validation.

d. Add training in business fundamentals, entrepreneurialism, Intellectual property protection.

e. Add in product and customer lifecycles, UI/UX, information architecture.

f. Increase training in leadership, communication, collaboration.

g. Explore practices in complexity and resilience.

h. Exposure to the computational infrastructure of full-stack systems.

Proposed Content - In response to the requirements guidance provided by stakeholders, educators, and students, the following candidate topics have emerged:

a. Modernize and unify engineering training built upon core math, theory, and computing.

b. Increase software training in general, emphasis on software architecture design, system modeling, assembly, and verification techniques.
c. Increase training regarding integrated, scalable, real-time, orchestrated systems.
d. Ensure graduates understand business fundamentals, requirements management, and analytics that lead to achievable, effective, desirable solutions.
e. Basic training in iterative design/development/release methodology.
f. Basic training in the customer lifecycle and product development lifecycles.
g. Add hands-on training with agile methods, rapid prototyping, testing, and validation.
h. Enhance leadership, collaboration, communication, and emotional intelligence.
i. Add skills related to data analytics, analysis, risk, and validation.
j. Essential training in ethics, social awareness, and sustainability.

For the sake of clarity, we have coalesced the preceding elements into 8 building blocks:

**Gap Analysis and Curriculum Development** - Each department should review its curriculum, prepare gap analysis of current courses and desired outcome, and proposed additional technical electives to cover the gap. Please see Appendix D for the proposed Gap Analysis and Curriculum Development for ECE Dept. We also recommend a program-wide review of existing curriculum, to find opportunities to leverage new material and synchronize new content with existing courses.

**CO-OPs, Incubators and Internships** - As part of the curriculum, hands-on student training will be an essential component. We are expecting each graduate to participate in a two-quarter CO-OP internship (summer, fall). Students will work with industry partners on actual systems. The goal is to give students the chance to gain valuable experience with modern engineering methods, team practices, and real-world system design, development, and support challenges over a two-term sequence. We anticipate this will become a viable recruiting mechanism for industry partners as well.
V. **What research topics will inform the evolution of our curriculum?**

Engineering systems are becoming ever more complex and the limitations of the current state of the art to address the complex systems challenges are recognized in many different disciplines. The current state of systems engineering has been described as an integration of ad-hoc subsystems that becomes a complex system rather than intentionally engineering a complex system. With the rise of sophisticated high fidelity models, model-based engineering and digital engineering, the demands are increasing to consider a wider range of the functional and operational needs as well as its impact and future. It is widely recognized that today’s systems engineering is unable to meet this needs. This section presents the research challenges to support the engineering and design of such complex systems.

A snapshot of a complex system can be schematically summarized as figure 1. An engineering design starts by identifying the needs which may arise from a variety of reasons, e.g. the market, the business model or regulation changes (bottom left-hand side of figure 1). They then need to be articulated as a set of requirements which are often reduced to a set of key parameters. Myriads of challenges arise at this stage as a complex systems design is an ill-defined problem where the internal relationships and potential conflicts of the requirements are not well known at this stage. The decisions that are made at this stage can have the biggest impact on the final design and yet the sensitivity of these decisions to the outcome is not known.

Based on the set of key parameters and requirements, traditional engineering design begins. Engineers typically create an initial systems design based on intuition and low fidelity analyses. The complex system is then broken down to subsystems and components often based on disciplinary and/or functional requirements. As the design process progresses down to component levels, the further details of the designs become available and higher fidelity analyses become possible. However, the primary design parameters defined at the top level often become the design constraints and they are rarely changed due to its complex implications on the other subsystems and components. It is well known that returning to modify the primary design parameters based on high fidelity models and iterating would lead to optimum design, however, the complexity and resource/time demands often prevent many of such iterations. Again, this challenge arises from the unknown internal relationships of various design parameters as well as across fidelities and scale. It is the coupling of various physical and engineering disciplines where traditional Multidisciplinary Design Optimization (MDO) tools have been applied. The current state of the art in MDO also addresses numerical optimization approaches to integrate across scale, model fidelities and uncertainties in design.
Today's engineering considers many more requirements beyond the more “traditional” engineering disciplines (e.g. mathematical and physics based models), as shown in figure 1. The current state of the art methods such as multidisciplinary design optimization (MDO) can couple the traditional engineering disciplines as well as multiscale and multifidelity models (e.g. in the case of aircraft, integrated material-structures represent a multiscale system design). However, the demands for tomorrow’s systems engineering include the impact on society, operations and maintainability, ethics, policy and regulations, lifecycle cost and recyclability. They, in turn, form new systems requirements, both in terms of how to adapt the operation of the existing design dynamically and how to inform the new engineering designs. These are areas that need more research to appropriately characterize and integrate new considerations in the context of engineering and how they should influence the design itself.

The challenges identified in complex systems engineering are substantial and represent a potential need for a revolutionary shift in the way we approach engineering design. We summarize the challenges into three categories and further identify the research needs based on the existing research fields. The three categories are:

**C1.** How to represent and analyze the design requirements from traditionally non-engineering considerations, such as societal impact, human factor, ethics, supply chains, lifecycle cost.

**C2.** How to quantify the influence of these various factors such that the internal relationships of the requirements and their uncertainties and sensitivities to the parameters can inform the engineering design. Without this understanding, unintended consequences cannot be identified.
**C3.** How to integrate the analytical, computational and experimental tools and data to determine the optimum engineering design efficiently.

Broadly speaking, C1 is domain/discipline specific including both traditional engineering and non-engineering subject areas. C2 relates to analyzing the relationships between the domains. C3 is the underpinning mathematics and computational research such as design optimization that enable C1 and C2. We recognize that there are active research activities concerning each of these challenges at various extents. We also identify that the current state of the art is insufficient to address the rising challenges of complex systems engineering and continuing research is critical. Without claiming to be completely comprehensive, the specific research fields are listed below:

- Numerical modeling and simulation of physics and human
- Multi-fidelity/multidisciplinary/multiscale design optimization methods
- Data-driven control and design optimization
- Cybersecurity
- Machine learning and AI in engineering designs
- Model-based systems engineering
- Digital twin and digital threading
- Uncertainty quantification and propagation
- System design at scale
- Verification and validation
- Workflow frameworks
- Visualization
- Human-system interaction / interface
- Humans as a System Element
- Team science in engineering
- Low-latency, high bandwidth, real-time communications
- Requirements management
- Adaptive complex systems
- Resilience in the face of complexity
- Full lifecycle simulation and modeling
- DevOps of Modular Open Systems
- Computational infrastructure
Appendix A: Summary of the Dean’s Council of Advisors
Recommendations

The Jacobs School Dean’s Council of Advisors focused on two meetings (May 4, 2018 and November 2, 2018) on a discussion of the proposed systems engineering program. Twenty-four (24) high-level executives participated in the discussion along with Jacobs School systems engineering committee members. The Council participants represented companies including Chugai Pharmaceuticals, Corning, General Atomics, Genentech, Google, Leidos, Lytx, NEA Ventures, Qualcomm, Resmed, Scopus Ventures, SPAWAR, Viasat, WreThink and XCom.

At the November 2, 2018 meeting, the members received a preview of the curricula outlined in this white paper and generally provided a positive endorsement of the concepts presented.

The following is a synopsis of the input of this Council.

Context

Today’s systems are more complex and are characterized by:

- Networked ecosystem of distributed, often peered suppliers.
- Systems have billions of interactions: data and models are a competitive advantage.
- Software products have rapid, almost continuous updates integrated around rigorous interfaces (APIs).
- Software is central, and often the organizing principle to the company’s system.

Companies face many challenges including:

- We are collecting data but don’t necessarily understand our own data--machine learning will be key.
- The tools are developing rapidly, but the industry talent may not be keeping up.
- Verification and validation of complex systems are difficult.

Key Skills and Capabilities for Systems Engineers

- Domain-specific knowledge, technical depth area
- Good at building and driving cognitive models
- Understand both technical and organizational systems
- Leadership, business, team dynamics, communications, ethics
- Higher level strategic and systems thinking
- Understand product and life cycle management
- Understand requirement management and evolution
Curricula Content

- Fine points on the curricula that should be included:
  - Dynamic, iterative models
  - Test-driven design, modularity, and evolution
  - Operating models when dealing with systems at scale
  - Understand how to modernize existing systems
  - Security-systems are often open and can be vulnerable
  - Tools and skills to include
    - Dynamic Visualization
    - DevOps – Live and adaptive design
    - Understanding power and influence

- Complex Systems Design-information systems, human systems, modeling and reporting.
- Data-driven machine learning.
- First-principles parametric simulation.
- Hardware/software boundary understanding.
- Design thinking and product management-understand the voice of the customer and reconcile what you want to make with what you can make.
- Strong connection to industry.
- Strengthen human connection in the curriculum: human in the loop design, customers at the center of design.
- Include moral and ethical decisions in engineering-think about unintended consequences.

Degree Program

- A 5-Year Bachelor/Masters/Co-Op program would be a differentiated, high-value program.
- Students who have completed the bulk of their coursework could participate in a year-long systems engineering leadership curriculum.
- Consider short-courses on new tools for industry systems engineers.

Research Frontiers Associated with Systems Engineering

- Complexity
- Security
- Robustness
- Explainability
- Distributed computing and control
- Systems modeling
- Cyber resilience
- Big data architectures
- Network dependencies
- Modular/piece-wise “Operational Systems Test and Validation”
Appendix B: Summary of Jacobs School Corporate Affiliates Program Recommendations

The Jacobs School Corporate Affiliates Program is comprised of 70 companies who hire Jacobs School graduates. The systems engineering initiative was discussed at the June 2018 CAP board meeting, and a subcommittee of CAP executives held three meetings in 2018 to articulate their recommendations, including:

- **Ensure Students Have a Depth Area**
  - Graduates should have a core competency in an engineering discipline
  - Even a master's degree should have a domain focus

- **Emphasize Real-World Experience**
  - Hands-on projects that force systems thinking, including; customer requirements elicitation, architecture, integration, prioritization, budgeting, and risk management
  - Co-Op Experience
  - Invited talks and case studies by industry systems architects

- **Develop Systems Thinking**
  - Understand the Company’s Big Picture: mission, business case, the full system, and customer capability needs.
  - Facilitate integration across disciplines (hardware/software/testing) and across functions (engineering/marketing/business).
  - For any given project, understand the problem statement—what are we trying to solve and why. Be flexible, the problem statement may evolve during discovery
  - Elicit customer requirements and what outcomes really matter to the customer.
  - Design Thinking and product management. Design to capabilities.
  - Prioritize and Weigh Tradeoffs to deliver the highest value.

- **Develop Leadership Skills**
  - Effective team dynamics, including distributed and virtual teams
  - Clear, crisp communication
  - Decision making, even when you don’t have all the information with plan b and c in mind
  - Active listening—ask the right questions, listen and learn

- **Teach Modern Systems Engineering Tools**
  - Operational systems test and validation
  - Data-driven systems modeling and optimization
  - Software engineering, including network patterns and interface design
  - Cybersecurity vs performance trade-offs
  - Consider offering technical tools short courses for experienced engineers
Appendix C: Summary of Systems Engineering programs across selected institutions

Georgia Tech: BS, MS, and PhD, all sitting in industrial engineering. Strong on the technical side, ample coverage of fundamentals in industrial engineering and operations research, together with the incorporation of many courses on more “modern” topics, especially at the graduate level: data analytics, computational science, business. There are opportunities for entrepreneurship, but outside the courses/curriculum. Not much emphasis on either leadership or ethics.

Univ. of Michigan: MEng in Systems Engineering and Design (both on campus and online). Emphasis on systems across multiple domains (beyond typical industrial setting) and focus on integration across Core/Electives/Practicum. This MEng seems to be conceived putting SE at the center, not as an afterthought. The technical core is in SE and Design. Electives are less structured, and might not provide enough concentration of courses in any particular area. Missing computational and data analytics courses. Business/entrepreneurial component missing from the curriculum.

Univ. of California, Berkeley: MS in Systems Engineering. A varied number of different areas, flexible to accommodate students from different areas. Must take 4 out of 6 core courses in Scalable Spatial Analytics; Behavioral Modeling for Engineering, Planning and Policy Analysis; Sensors and Signal Interpretation; Civil Systems: Control and Information Management; Control and Optimization of Distributed Parameter Systems; Energy Systems and Control.

Northwestern: Minor in Engineering Management (MS). 2 requires courses, Engineering Management and Decision Tools for Managers, 1 elective, also from management.

Cornell: Interdisciplinary program across 5 engineering departments MEng (on campus and online) and Ph.D. A large number of specializations plus SE core courses allow for flexibility, touches upon many different areas, might be appealing to students -- so long as consistency in SE core can/is maintained across. Management and business side, even leadership, present in the curriculum.

Purdue: Interdisciplinary Master of Science in Engineering. Classical emphasis on modeling, stochastics, and optimization. 12 hours of systems and 18 hours of engineering. Extensive distance courses. Have a separate Engineering Management and Leadership.

MIT: Ph.D. in Social and Engineering Systems, MS in Technology and Policy Program. Engineering Systems Division ended in 2015 replaced by Institute for Data, Systems, and Society as part of the process where Systems Engineering was viewed as an important subfield within Engineering Systems. The latter refers to “integrative holistic view of large-scale, complex, technologically-enabled systems which have significant enterprise level interactions and socio-technical interfaces”. Also Interdisciplinary Doctoral Program in Statistics; Minor in
Statistics and Data Science; MicroMasters in Statistics and Data Science (online courses), and Professional Education.

**Penn State**: Master of Engineering in Systems Engineering. 12 courses, 2 years. Also online. Classical emphasis on modeling and optimization, combined with probability, management, and programming.

**Univ. of Wisconsin**: BS in Industrial and Systems Engineering. Five main areas: Decision Science and Operations Research; Health Systems Engineering; Human Factors and Ergonomics; Manufacturing and Production Systems; Quality Engineering and Management. Classes in Accounting, Human Factors, Economic Analysis, Production planning, modeling, Junior & Senior Design, Technical Writing. Junior Design course focuses on open-ended problem-solving projects or major homework assignments. Senior Design course team-based project experience to address a real-world design challenge posed by an external organization.

**Virginia Tech**: BS in Industrial and Systems Engineering. Courses in Engineering Economics, Data Management, Operations Research, Manufacturing Processes, Operations Research, Human Factors, Lean Manufacturing, Production Planning and Control, Quality Control. Capstone: Full senior year - Capstone design sequence for ISE majors. Students work in teams to apply industrial and systems engineering and project management tools to define and analyze a real-world problem, and communication of solutions to project stakeholders.

**Columbia**: MS, online. Emphasizes modeling tools and optimization. Four core courses on Applied Systems Engineering; Systems Engineering: Tools and Methods; Optimization Models and Methods; and Stochastic Models. Six elective courses. 4-5 from Infrastructure and Sustainability; Mechanical, Electrical and Computer Systems; and Biological Systems; and 1-2 courses from Management.

**UIUC**: BS in Systems Engineering and Design (used to be Engineering Science). Brings together “basic sciences, engineering analysis, and engineering design.” Specific emphasis on “Design experience and project management are emphasized and integrated across the core with a focus on establishing critical problem-solving skills applied across disciplines, strong communication skills, and the ability to work effectively and get results in a team environment.” Beyond the technical core, a huge number of secondary fields to specialize on.

**Additional References from Professional Community**

**Appendix D: Gap Analysis and Proposed ECE Systems Engineering Curriculum**

ECE at UCSD is a nationally ranked, rising program in the field of Electrical and Computer Engineering. ECE academic leadership have maintained a clear focus on the core skills ECE graduates must possess to be effective practitioners in industry and academia. To date, the curriculum has focused on traditional methods, theories, mathematics and physics related to this field of study.

The curriculum development team conducted a gap analysis between the current iteration of the ECE curriculum, and the projected requirements of our proposed ECE + System Engineering concentration.

At present, ECE is principally focused on theory, math, and methods of hardware engineering (shown in light blue in the Gap Analysis Diagram). Based on stakeholder guidance, the ECE+SYSE concentration would require additional training in the disciplines (in dark blue).

In particular, gap analysis indicates four key areas of focus:

a. **Software:** Designing, assembling, maintaining complex solutions in scalable, distributed, secure, data-driven environments controlled by software

b. **Agility:** Experience with agile methods, data-driven analytics, cognitive modeling, visualization, verification, process management, performance/risk/cost analysis

c. **Business:** Awareness of business fundamentals, operations, metrics, to ensure decision-making skills leverage engineering and business factors

d. **Leadership:** Hands-on practicum related to communication, collaboration, ethics, diversity, and social awareness

**Proposed Content** - In response to the requirements guidance provided by stakeholders, educators, and students, and the preceding gap analysis, the following candidate topics have emerged:

k. Modernize and unify engineering training built upon core math, theory, computing
l. Increase software training in general, emphasis on software architecture design, system modeling, assembly, verification techniques
m. Increase training regarding integrated, scalable, real-time, orchestrated systems
n. Ensure graduates understand business fundamentals, requirements management, and analytics that lead to achievable, effective, desirable solutions
o. Basic training in iterative design/development/release methodology
p. Basic training in the customer lifecycle, product development lifecycles
q. Add hands-on training with agile methods, rapid prototyping, testing, validation
r. Enhance leadership, collaboration, communication, emotional intelligence
s. Add skills related to data analytics, analysis, risk, validation
t. Essential training in ethics, social awareness, sustainability

For the sake of clarity, we have coalesced the preceding elements into 8 building blocks:

Proposed New Courses - The following courses comprise the core of the Systems Engineering concentration. ECE140 is a relatively new course at UCSD, while the other two are proposed courses designed specifically for this concentration.

<table>
<thead>
<tr>
<th>ECE-140A</th>
<th>Art of Product Engineering (A)</th>
</tr>
</thead>
<tbody>
<tr>
<td>● Group “IOT” project: design, build “cloud-based” IOT product</td>
<td></td>
</tr>
<tr>
<td>● Agile methods in planning, design, development, management</td>
<td></td>
</tr>
<tr>
<td>● Full-stack development using Javascript, CSS, HTML, Python, SQL, Raspberry Pi</td>
<td></td>
</tr>
<tr>
<td>● Cyber security</td>
<td></td>
</tr>
<tr>
<td>● Business fundamentals, business models</td>
<td></td>
</tr>
<tr>
<td>● Entrepreneurial practices</td>
<td></td>
</tr>
<tr>
<td>● Communication, leadership, ethics</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ECE-DSR</th>
<th>Data science and risk management</th>
</tr>
</thead>
<tbody>
<tr>
<td>● Data-driven systems modeling and optimization</td>
<td></td>
</tr>
</tbody>
</table>
- Trend identification, analysis, tuning
- Machine learning, predictive analytics, probabilistic reasoning
- Risk management

**ECE-OSD**

**Orchestrated system design**
- Systems design patterns (systems thinking and domain models)
- Component discovery patterns
- Connection and assembly patterns, Service-oriented architecture
- Communication patterns [informational law]
- Flow patterns in real-time (events, transactions) [transactional law]
- Validation patterns (confirm normal operations)
- Scaling patterns -- maintaining performance at scale
- Adaptation patterns: versioning, integration, and disintegration

**ECE-COOP1**
**Summer Co-op internship**

**ECE-COOP2**
**Fall Co-op internship**

**ECE-141B**
**Art of Product Engineering (B)**
- Capstone course -- new product startup competition
- Customer lifecycle mgmt., product development lifecycle
- Teams build a full-stack IOT product for chosen customer/market
- Agile planning methods, bi-weekly sprint update presentations

**Unifying Practicum** - The following courses have been newly developed or presently undergoing revision and enhancements. These are modern systems-oriented courses in hardware and software and are well-aligned to meet the goals stated in this initiative.

**Improvements to Existing Software-Oriented Courses**

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ECE-15</td>
<td>Engineering Computation in C</td>
<td></td>
</tr>
<tr>
<td>ECE-16</td>
<td>Rapid Hardware and Software Design</td>
<td></td>
</tr>
<tr>
<td>ECE-17</td>
<td>Object-Oriented Programming in C++</td>
<td>- Language basics</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Core principles of data</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Storage hierarchies</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Software development life-cycle</td>
</tr>
<tr>
<td>ECE-141AB</td>
<td>Software Foundations</td>
<td>- Intermediate software modeling</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Design Patterns</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Software architectures</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Test-driven Design and Validation</td>
</tr>
</tbody>
</table>
**Modern Teaching Methods** - In order to maximize the benefits of the learner outcomes of this curriculum, the team is proposing that the curriculum should be constructed in accordance with the “inverted classroom” model. Teaching and learning experts have conducted studies that show improved student performance, along with improved retention using these methods. Practically speaking, an inverted classroom is one where students watch pre-recorded lectures prior to attending class. Classroom time can then focus on in-class activities, Q&A, and other hands-on learning experiences.

**CO-OPs, Incubators and Internships** - As part of the curriculum, hands-on student training will be an essential component. We are expecting each graduate to participate in a two-quarter CO-OP internship (summer, fall). Students will work with industry partners on actual systems. The goal is to give students the chance to gain valuable experience with modern engineering methods, team practices, and real-world system design, development, and support challenges over a two-term sequence. We anticipate this will become a viable recruiting mechanism for industry partners as well.

**Resources Required** - We developed an estimate of the cost of the new courses based on two sources. First, we used the actual development time associated with the development of ECE-16, ECE-17, ECE-140AB, and ECE-141AB. Second, we used a model from eLearning community (revised in 2017) for teacher-led, web-based learning materials. Generally, this means web-based content (or powerpoint), labs, quizzes, and related training materials. General guidelines for the material of this type (presuming you already have the expertise), is about 30 hours of development per learning hour. Some universities continuously update their curriculum annually, to keep it relevant and aligned with emerging trends.

We propose developing this curriculum using a team of contributors: curriculum developers, media developers, subject matter experts, industry and CTL reviewers.