UTC-IASE
UNIVERSITY OF CONNECTICUT

UCONN
SCHOOL OF ENGINEERING

SE 5001: Model-Based Systems Engineering

Course Instructor: Amy Thompson, Ph.D.

Catalog Description. 3 credits. This course is designed to provide students with the foundations of model-based systems engineering. Students will develop skills in the areas of fundamental logical, behavioral, and physical representations of engineered cyberphysical systems. Topics include software and systems requirements engineering, interface design and modeling, system architecting, system verification and testing, and system simulation. Emphasis is placed on modeling cyberphysical systems using modern MBSE principles, methods, and tools. Examples include a water distiller, a residential security system, an automobile, an elevator, and a geospatial library for the demonstration of the theoretical and practical aspects of systems modeling. The course is designed for all graduate students pursuing graduate certificates and degrees in an engineering discipline.

Pre-Requisites. An undergraduate degree in engineering or science.

Intended Audience. The course is designed for all graduate students in engineering.

Course Delivery Method. The course will be offered online, asynchronously, in small recorded modules according to the course schedule and syllabus. Direct and live communication with the instructor will be available each week, according to the class schedule, for discussion, questions, examples, and quizzes. Attendance at live sessions is required, and you must notify the instructor in advance if you cannot attend. A social networking tool called Slack will be used to communicate with students and the instructor between live sessions.

Course Objectives.

(1) Student uses the knowledge and information gained in the course to expand and improve the application of model-based systems engineering in their field.
(2) Student implements model-based systems engineering practices in their field that result in higher levels of value and satisfaction with engineered systems.
(3) Student pursues further in-depth education and training in systems engineering.

Anticipated Student Outcomes. By the end of SE 5001, a student will be able to:

(1) Describe the processes, methods, and practices of model-based systems engineering.
(2) Apply model-based systems engineering practices and methods to relevant examples.
(3) Develop and relate requirements, architectures, behavior, specifications, verifications, and tests that represent cyberphysical systems using model-based systems engineering methods.
(4) Create effective SysML diagrams that accurately represent views of an engineered system.
(5) Demonstrate analysis of systems using model-based systems engineering approaches that lead to better and increased performance of systems.
(6) Create system simulation models that represent the behavior and performance of cyberphysical systems.
(7) Communicate effectively in teams, via interim and final project reports.

Course Organization. The course is organized into five learning modules:

(1) Solving the Right Problem: Capturing Mission, Needs and Requirements
(2) Understanding and Modeling the System Context and Environment
(3) Modeling and Simulating Cyberphysical System Behavior
(4) Defining the Cyberphysical System and Components
(5) Cyberphysical System Failures, Verification, and Testing

Background information is provided and required on the following subjects: The INCOSE Systems Engineering Technical Processes in the INCOSE Systems Engineering Handbook.

Course Outline. The structuring of these five learning modules into 13 lectures of a one semester course, along with the topics and references, is described in the following:

------ Module 1. Solving the Right Problem: Capturing Mission, Needs and Requirements-----

Week 1: Review of Systems Engineering Technical Processes and Introduction to Model-Based Systems Engineering

- Course Overview, Expectations, and Requirements
- Review of the INCOSE Systems Engineering Lifecycles and Technical Processes
- Understand basic concepts of MBSE
- The Business Case for MBSE
- Introduction to your MBSE Tool

-------Module 2. Understanding and Modeling the System Context and Environment ----------

Week 2: Creating Requirements

- Writing Good Requirements
- Eliciting Requirements
- Capture Originating Requirements
- Requirement Types and Characterization
- Writing Constraints and Performance Requirements
- Creating and Organizing Requirements in the MBSE tool

Week 3: Requirements Modeling, Analysis, and Management

- Requirement Relationships in MBSE
- Requirements Visualization: Requirements Diagram, Traceability Hierarchy Diagram
- Using your MBSE Diagram Tool
- Creating and Handling Requirement Risks and Concerns
- Requirements Management

Week 4: Define the System Context and Boundary

- Define the Physical Context
- Environmental Characteristics as Components
- Use of Diagrams and SysML: Structure Block Diagram, Component Hierarchy Diagram

Week 5: Define Interfaces and External Interface Elements

- Describe Interactions
- Capturing the Interfaces
- Use joins, connects, transfers, and comprises relationships in MBSE tool. Creation of links, interfaces, and items in MBSE tool.
- Changing Interfaces
- Use of Diagrams and SysML: Interface Block Diagram, Physical Block Diagram, Interface Hierarchy Diagram, Flow Internal Block Diagram

----------Module 3: Modeling and Simulating Cyberphysical System Behavior -----------------

Week 6: Define the System Behavior

- Behavior Constructs

- Functions and Functional Hierarchy
- States/Modes
- Use Cases and Threads
- Other Types of Behavior Diagrams: Activity, FFBD/EFFBD, N2 Diagrams, Sequence
- Use of Folders to Organize Work

Week 7: Advanced System Behavior Modeling

- Integrated Systems Behavior
- Functional Traceability and Model Relationships: Functions to Use Cases to Requirements
- Performing Functional Decomposition
- Capture Functional and Performance Concerns and Risks
- Capturing Behavior with SysML
- Simulating Behavior: Introduction/Getting Started

Week 8: Introduction to Simulating Cyberphysical Systems

- Handling Arrivals to the System
- Handling Timing of Events and Queues
- Dealing with Logic and Constructs
- Handling Exits to the System
- Execution and Analyzing Results
- Simulation: Options and Preferences
- Simulation: Use of Probabilities and Distributions
- Streams for Repeatability

----------------Module 4: Defining the Cyberphysical System and Components ----------------

Week 9: Allocate the Behavior to Physical Components

- Relating Functional and Physical Models
- Tool Automatic Allocation
- Determine Partitioning Strategies
- Determining When to Stop Allocating
- Define Functional Boundaries and Context: Activity Diagram and N2 Diagram
- Use of Images in Model

Week 10: Defining Physical Components

- Allocate Functions to Next Level Components
- Refine External Interface Definitions
- Derive or Refine Internal Interfaces
- Assign/Derive Constraints for Components
- Capture Physical Architecture Concerns and Risks
- Capturing Physical Model using SysML

Week 11: Failure Modes and Effect Analysis (FMEA)

- Basics of FMEA
- Application of FMEA in Model-Based Systems Engineering

-----------Module 5: : Cyberphysical System Failures, Verification, and Testing -----------------

Week 12: Verification Requirements and Test Plans

- Relationships between Requirements, Documents, Verification, Tests, and Components
- Writing a Verification Requirement
- Creating a Verification Hierarchy
- Creating Verification Properties
- Create a Verification Event
- Creating Testing Plans and Procedures
- System and Software Testing Methods and Approaches
- Creating Test Threads
- System Testing: Sequence and Analysis

Week 13: Integrating and Deploying SysML and MBSE into a Systems Development Environment

- Understanding the Systems Model in a Broader Context
- Tool Roles
- Information Flows Between Tools
- Data Exchange Mechanisms

USEFUL READING.

Texts are available through a local or online bookstore. The UConn Co-op carries many materials that can be shipped via its online Textbooks To Go service. For more information, see Textbooks and Materials on our Enrolled Students page.

Required Texts

(1) Buede, Dennis and William D. Miller. The Engineering Design of Systems: Models and Methods, 3rd Edition. Wiley. ISBN: 978-1-119-02790-4.
(2) Berenbach, Paulish, Kazmeier, Rudorfer. Software and Systems Requirements Engineering in Practice. 2009. ISBN: 978-0-07-160547-2.
(3) Friedenthal, Moore, and Steiner. A Practical Guide to SysML, 3rd Edition: The Systems Modeling Language. The MK/OMG Press. 2012. ISBN-13: 978-0128002025.

References for Vitech CORE Modeling

(1) Architecture Definition Guide
(2) System Definition Guide
(3) COREsim User Guide
(4) COREscript Reference

Background and Supporting Texts from INCOSE*

(1) INCOSE, Systems Engineering Handbook Version 4.
(2) M. Ryan and L. Wheatcraft, Guide for Writing Requirements, INCOSE Technical Product INCOSE-TP-2010-006-02, 1 July 2015.

*Full-time students can gain access to all INCOSE materials in .pdf format by joining INCOSE as a student member. Employees of an INCOSE CAB Organization who are not INCOSE individual members are able to sign up for a CAB Limited access account to the materials at no-charge. If you are not already an individual member, you can join as a CAB employee or Student to access all materials and participate in limited INCOSE benefits.

Other Useful Reading and Materials

(1) INCOSE Materials
- INCOSE-TP-2003-002-04, 2015.
- BKCASE Editorial Board. 2015. The Guide to the Systems Engineering Body of Knowledge (SEBoK), v. 1.5., R.D. Adcock (EIC). Hoboken, NJ: The Trustees of the Stevens Institute of Technology. www.sebokwiki.org.
- INCOSE, Journal of the International Council on Systems Engineering, Seattle, W.A.: International Council on Systems Engineering.
- See also the INCOSE web site: http://www.incose.org/ for other useful products and resources.

(2) Relevant Standards

- ANSI/EIA-632-1998, EIA Standard—Processes for Engineering a System, Arlington, V.A.: Electronic Industries Association, 1999.
- IEEE-STD-1220-2005, IEEE Standard for Application and Management of the Systems Engineering Process, New York: IEEE Computer Society, 2005.
- MIL-STD-498, Military Standard: Software Development and Documentation, Washington D.C.: United States of America Department of Defense, 1994.
- MIL-STD-499B, Military Standard—Systems Engineering—Draft, Washington D.C.: United States of America Department of Defense, 1994.
- ISO/IEC 15288-2015, Systems and Software Engineering—System Life Cycle Processes, 2015.
- ISO/IEC, ISO/IEC 29148, FDIS, Systems and Software Engineering—Life Cycle Processes—Requirements Engineering, 2011.
- International Institute for Business Analysis, A Guide to the Business Analysis Body of Knowledge® (BABOK® Guide), Version 2, 2009.
- ANSI/AIAA G-043A-2012, Guide for the Preparation of Operational Concept Descriptions, American National Standards Institute, American Institute of Aeronautics and Astronautics (sponsor), 2012.

Copyright. Copyrighted materials within the course are only for the use of students enrolled in the course for purposes associated with this course and may not be retained or further disseminated.

Grading. Grading of the course will be exclusively based on the Course Project described below. Optional homework and quizzes will be assigned to students during the semester for bonus points towards grade.

Grade	Letter Grade	GPA
97-100	A+	4.3
93-96	A	4.0
90-92	A-	3.7
87-89	B+	3.3
83-86	B	3.0
80-82	B-	2.7
77-79	C+	2.3

Grade	Letter Grade	GPA
73-76	C	2.0
70-72	C-	1.7
67-69	D+	1.3
63-66	D	1.0
60-62	D-	0.7
<60	F	0.0

Due Dates and Late Policy. All course due dates are identified in the Course Schedule. Deadlines are based on Eastern Standard Time; if you are in a different time zone, please adjust your submittal times accordingly. The instructor reserves the right to change dates accordingly as the semester progresses. All changes will be communicated in an appropriate manner.

Course Project. A project is to be developed by each student, which is expected to evolve during the entirety of the semester. The project will entail creating a systems engineering model of an engineered product using the Vitech CORE tool by applying MBSE methods. The final deliverable (presentation file) shall include descriptions of the model, SysML views that describe the model, and summary conclusions about the performance of the model and engineered product. A separate rubric with the details of the project will be provided to the students on HuskyCT. A mid-term and final report are the main deliverables of this project, and are the basis of which each student will be graded.

Student Conduct. http://www.dosa.uconn.edu/student_code.html. Students are responsible for adherence to the University of Connecticut student code of conduct. Pay attention to the section on Student Academic Misconduct, "Academic misconduct is dishonest or unethical academic behavior that includes, but is not limited, to misrepresenting mastery in an academic area (e.g., cheating), intentionally or knowingly failing to properly credit information, research or ideas to their rightful originators or representing such information, research or ideas as your own (e.g., plagiarism)." Examples of academic misconduct in this class include, but are not limited to: copying solutions from the solutions manual, using solutions from students who have taken this course in previous years, copying your friend's homework, looking at another student's paper during an exam, lying to the professor or TA and incorrectly filling out the student workbook.

Attendance. Students should make every effort to attend the live sessions and to talk with students in the Slack chat forum to get help and assistance from others. It is practically impossible to follow the class if classes are missed.

Absences. Make-up of missed exams requires permission from the Dean of Students, see "Academic Regulations." Midterm-exams are treated the same as Final Examinations. Students involved in official University activities that conflict with class time must inform the instructor in writing prior to the anticipated absence and take the initiative to make up missed work in a timely fashion. In addition, students who will miss class for a religious observance must "inform their instructor in writing within the first three weeks of the semester, and prior to the anticipated absence, and should take the initiative to work out with the instructor a schedule for making up missed work."

Adding or Dropping a Course. If you should decide to add or drop a course, there are official procedures to follow:

- Matriculated students should add or drop a course through the Student Administration System.
- Non-degree students should refer to Non-Degree Add/Drop Information located on the registrar's website.

You must officially drop a course to avoid receiving an "F" on your permanent transcript. Simply discontinuing class or informing the instructor you want to drop does not constitute an official drop of the course. For more information, refer to the online Graduate Catalog,

Academic Calendar. The University's Academic Calendar contains important semester dates.

Students with Disabilities. Students needing special accommodations should work with the University's Center for Students with Disabilities (CSD). You may contact CSD by calling (860) 486-2020 or by emailing csd@uconn.edu. If your request for accommodation is approved, CSD will send an accommodation letter directly to your instructor(s) so that special arrangements can be made. (Note: Student requests for accommodation must be filed each semester.)

Course Schedule*

Date[1]	Topic	Module No	Details
Aug 28-Sept 1	*Week 1: Systems Engineering Technical Process Review and Introduction to Model-Based Systems Engineering*	1	Live Meeting August 28th 5:00pm Live Makeup: Tuesday 29th 5:00pm
Sept 4-8	*Week 2: Creating Requirements*	1	No Live Meeting on Sept 4th Holiday
Sept 11-15	*Week 3: Requirements Modeling, Analysis, and Management. Present and Discuss Course Project*	1	Live Meeting Sept 11th
Sept 18-22	*Week 4: Define the System Context and Boundary*	2	Live Meeting Sept 18th **Project Proposal Due**
Sept 25-29	*Week 5: Define Interfaces and External Interface Elements*	2	Live Meeting Sept 25th
Oct 2-6	*Week 6: Define the System Behavior*	3	Live Meeting Oct 2nd
Oct 9-13	*Week 7: Advanced System Behavior Modeling*	3	Live Meeting Oct 9th
Oct 16-20	*Week 8: Introduction to Simulating Cyberphysical Systems*	3	Live Meeting Oct 16th **Project Mid-Term Report Due**
Oct 23-27	*Week 9: Allocate the Behavior to Physical Components*	4	Live Meeting Oct 23rd
Oct 30 – Nov 3	*Week 10: Defining Physical Components*	4	Live Meeting Oct 30th
Nov 6-10	*Week 11: Failure Modes and Effect Analysis (FMEA)*	5	Live Meeting Nov 6th
Nov 13-17	*Week 12: Verification Requirements and Test Plans*	5	Live Meeting Nov 13th
Nov 20-24	*Thanksgiving Recess*		No Live Meeting on Nov. 20th Holiday
Nov 27-Dec 1	*Week 13 Integrating and Deploying SysML and MBSE into a Systems Development Environment*	5	Live Meeting Nov 27th **Project Final Report Due Dec. 1**

* Schedule is tentative and may change

[1] First Date indicates release of lecture modules

Instructor's Contact Information:

- Amy Thompson: amy.2.thompson@uconn.edu Phone: (860)486-8462
- Office Hours: Tuesday 1:00 – 3:00pm and Wednesday 10:00am – Noon

Helpful Links:

- Virtual Computer Lab at UConn: http://skybox.uconn.edu/
- Course Material: https://lms.uconn.edu
- Institute for Advanced Systems Engineering: http://www.utc-iase.uconn.edu/