The Future of Engineering Education
2024 Annual Conference & Exposition
Oregon Convention Center
Portland, OR . June 23 - 26, 2024
ASEE
Paper ID #42910

# Integrating Model-Based Systems Engineering and Systems Thinking Skills in Engineering Courses

**Prof. Kavitha Chandra, University of Massachusetts, Lowell**

Kavitha Chandra is the Associate Dean for Undergraduate Programs and Professor of Electrical and Computer Engineering in the Francis College of Engineering at the University of Massachusetts Lowell. She directs the Research, Academics and Mentoring Pathways (RAMP) to Success summer bridge program that prepares first year engineering students with research, communication and leadership skills. Her research interests include computational and data-driven modeling of physical systems in acoustics and communications networks, model-based systems engineering, engineering education and user-centric design of emerging technology.

**Dr. Sara Kraemer, Blueprint for Education**

Dr. Sara Kraemer is an evaluator, strategist, and educator in STEM and education at Blueprint for Education. She has pioneered innovative program evaluation methods that lend themselves to programs, courses, and initiatives that focus on systems-level impacts. She is also a Lecturer in the Interdisciplinary Professional Programs in the College of Engineering a the University of Wisconsin-Madison, where she teaches classes on human factors engineering and digital transformation.

**Emi Aoki, University of Massachusetts, Lowell**

Emi Aoki is a PhD student in Electrical Engineering at the University of Massachusetts Lowell. Her past research experience has been in data analysis, computational modeling, systems engineering, and augmented reality application development and design. Her research interests include stochastic and physics-based models.

**Flore Stecie Norceide, University of Massachusetts, Lowell**

Flore Stecie Norceide is a PhD student in Electrical Engineering at the University of Massachusetts Lowell. Her research interests are neuromorphic computing, machine learning and data analytics. Her primary research focuses on benchmarking hardware and software architectures for event driven computation with a focus on its application to object detection, tracking and pose estimation.

**Dr. Ola Batarseh, Dassault Systemes**

Dr. Ola Batarseh is Solution Architect Director in the Digital Transformation team at Dassault Systemes where she conducts internal projects to raise the model-based capability, competency, and capacity. She conducts regular client consulting efforts on enabling improved projects using model-based methods and analysis. She is an adjunct professor at UML where she instructs students from USAF, Raytheon, and other professionals in MBSE.

# Integrating Model-Based Systems Engineering and Systems Thinking Skills in Engineering Courses

## Abstract

The advent of digital engineering practices across industries and defense organizations has created a need for graduating engineers to acquire skills in conceptualizing and creating digital models that capture the lifecycle of the product, system, or service of interest. The transition from traditional document-based models to digital models requires training in modeling languages and modeling methods to apply model-based systems engineering (MBSE) tools and techniques. These pillars are essential to designing digital models and employing a systems-thinking framework. Digital engineering skills focusing on MBSE topics are currently not widely available in undergraduate or graduate programs. We propose a modular approach for integrating these topics in a set of core courses in the second and third years of the engineering curriculum, building incremental skills across this two-year period, and leading to their application in capstone design projects in the final year. The modules are drawn from a recently implemented Digital Engineering graduate certificate for training a civilian Air Force cohort. This twelve-credit certificate included four semester-long courses in: (i) Systems, Models and Simulation for Digital Engineering; (ii) Model-Based Systems Engineering; (iii) Cyber-Physical Systems and Simulation; and (iv) Data-Driven Decision-Making and Risk Management. The modules are hosted on an online learning and course management system. Each module includes an experiential learning project that supports designing use cases with relevant stakeholders, conducting interviews with non-engineering domain-experts and end users of the system or creating digital models using systems modeling languages such as the Unified Architecture Framework (UAF) and the Systems Modeling Language (SysML). We pay particular attention to ensuring engineers learn to incorporate a human-centric approach in systems modeling, fostering a holistic and user-centric design process. This paper presents one example of a case study that introduces students to concepts of stakeholders, use cases and requirements analysis. The example takes the student through a sequence of stages using tools provided by UAF and SysML. These architectural platforms provide structure, but also include several degrees of freedom in design choices, allowing students to exercise and be assessed on the requisite systems thinking competency.

## 1.0 Background

Digital transformation is a process that aims to facilitate the move from document-based approaches in engineering, which may include specialty tools used by a variety of stakeholders, to a digital model of engineered systems that encompasses their entire life cycle. Such models include conceptualized, physical, or operational representations of systems that can be used to capture requirements and conduct analysis of systems design, use and performance. To achieve digital transformation, engineers require skills that meet the core aspects of both a model-based systems' thinking design approach and consideration of the role of humans that are involved in all aspects of the system [1, 2].

The U.S. Department of Defense (DoD) Digital Engineering (DE) Strategy [3], announced five years ago, presented the departments' goals to adopt more integrated digital modeling in their systems acquisition and procurement practices and operations and enable the use of digital artifacts to improve communication across all stakeholders. The goal of transitioning a traditional design-build-test methodology to a model-analyze-build approach extends the role of domain-experts such as systems engineers, to become knowledgeable of the requirements and practice of experts from other domains who engage with the system across its lifecycle. This transition is called digital transformation, and it has begun to be undertaken in industry, federal agencies and in health and medical service organizations [1].

Model-based systems engineering (MBSE) is a methodology that supports the management of requirements, design, analysis, verification, and validation of complex systems models - systems consisting of interoperating subsystems [4]. MBSE is expected to break the siloed responsibilities and functions as it leverages digital tools and technologies to model and simulate systems and represent them at different levels of abstraction to improve communication between stakeholders [5]. Graphical modeling languages have been proposed to support MBSE. The Systems Modeling Language (SysML) is considered the industry standard, general-purpose and "de-facto" modeling language. It is an extension of the Unified Modeling Language (UML), initially developed by the International Council on Systems Engineering (INCOSE) and the Object Management Group (OMG) [6]. SysML is a framework comprising different diagram types allowing for an agile and modular way to map the relationship of different model elements. Several other modeling languages, such as the Unified Architecture Framework (UAF), have evolved to address specific domains and industries.

Systems thinking, skills necessary for MBSE, refers to a set of interrelated skills that aim to understand both individual and holistic aspects of systems, forecast their behaviors, and implement adjustments to achieve desired outcomes [7]. The idea of fostering systems thinking skills in engineering curriculums has been increasing, as these skills are anticipated to facilitate understanding of emerging complex systems that often consist of interoperated and multi-perspective subsystems [4, 8]. In [9] the integration of systems thinking components in the existing educational system is proposed through lectures, labs, case studies, or capstone. In [8] the application of systems thinking is demonstrated across diverse engineering disciplines, including fluid mechanics, heat transfer, and mechanical, electrical, nuclear, and environmental engineering. The authors advocate integrating these systems thinking skills as an adjunct approach to gain more insights, as the complexity of systems driven by technical advancements are increasing.

This paper proposes a modular approach to integrating DE and MBSE practices into engineering courses. The design also supports incremental practice in systems thinking skills across the degree pathway. Modules that describe stakeholders, use cases and system requirements are not only foundational components for MBSE, but they can also instill the systems-thinking practice into students' skill set early in their education. These modules are designed to be independent of a particular course or system so that they can be integrated across multiple courses and presented using case studies of problems that students recognize or have experienced. As students study these online modules, taking on the role of systems engineers, the case studies provide opportunities to contribute to the design and solution space of the problem with authentic interest in the process.

Section 2.0 provides the motivation and prior work on the design of a DE graduate certificate. Section 3.0 presents a high-level use case diagram depicting the various stakeholders who are motivated in developing and scaling DE skills. Section 4.0 discusses the prototyping of modular design for DE skills, the use of architectural tools such as UAF and SysML. This section also presents an example of a few modules for a specific ecosystem and associated system of interest. Section 5.0 summarizes future work in this effort.

## 2.0 Experience from Designing a Digital Engineering Graduate Certificate

The authors developed a twelve-credit graduate certificate in DE responding to requests made by leaders on a local US Air Force Base and state economic council representatives. The request was in response to the DoD initiative [3] that identified the following five areas as necessary for their strategy to survive: (DE.1) Formalize development, integration and use of models; (DE.2) Provide an authoritative source of truth; (DE.3) Incorporate Technological Innovation; (DE.4) Establish Infrastructure and Environments and (DE.5) Transform Culture and Workforce.

Twenty graduate students, all in the civilian workforce at the noted air force base, most with engineering background, enrolled in the graduate certificate in Spring 2021. These students were at various stages of their careers, several in the manager and director role, and most all with the title of systems engineer. They understood the DoD and USAF directives for promoting the use of MBSE in their work on the acquisition of systems but were unclear on how DE would be rolled out or applied in their work. While DE is not a new field for defense contractors, it is a relatively new set of skills for those in the government who typically generate requests for contracts and acquire systems designed by contractors. The digital transformation paradigm is expected to create a common set of digital models and associated data that can serve as a means of communication and exchange of information between vendors and government procurers. To support these goals, the DE Certificate created the following four courses, each one semester in duration. They were all taught in an in-person format: (1) Systems, Modeling and Simulation for Digital Engineering; (2) Model-Based Systems Engineering; (3) Cyber-Physical Systems Modeling and Simulation and (4) Data-Driven Decision Making and Risk Management. These courses crosscut the four initiatives that the DoD Strategy proposed in providing a formal set of architectural platforms for developing digital models, incorporating technical innovations in design of engineered systems and the application of data in the context of systems models and decision-making.

One of the key outcomes from this two-year certificate was the realization that there are multiple directives and visions for digital transformation and engineering based on the organization and its mission. Due to the broad nature of this topic and based on where it is implemented, a modular approach that supports acquiring skills in DE education that are fit for purpose would be better able to address these varied needs. Towards this goal, the 14-week in-person courses are being mapped to 8-week online modules. The goal is to also allow these modules to be applied in the traditional engineering curriculum, thus scaling the training of future DE workforce.

## 3.0 A Systems Model of Digital Engineering Structure and Dependencies

A model of DE education is required to systematically include multiple stakeholders and their requirements and assess how the educational structure and content serve these requirements. A model designed with these objectives will allow a critical review of the resources required, collection of data and creation of a data-thread that can lead to periodic assessment and improvement in response to student experiences and changing requirements. More importantly, it allows a level of transparency to engage new stakeholders, new instructors, and new content. Fig. 1 is a SysML use case diagram that lists key stakeholders in developing the DE Educational Program and their use cases. Stakeholders identified here include individuals or a group of organizations involved in receiving, developing, or offering the educational system, such as students, educational institutions, and various agencies. Bubbles connected to each stakeholder depict functionalities or scope of the considered system associated with each stakeholder. The high-level visual representation as demonstrated here can serve as a starting point to communicate with a broader range of stakeholders to establish a unified understanding of the system's mission.

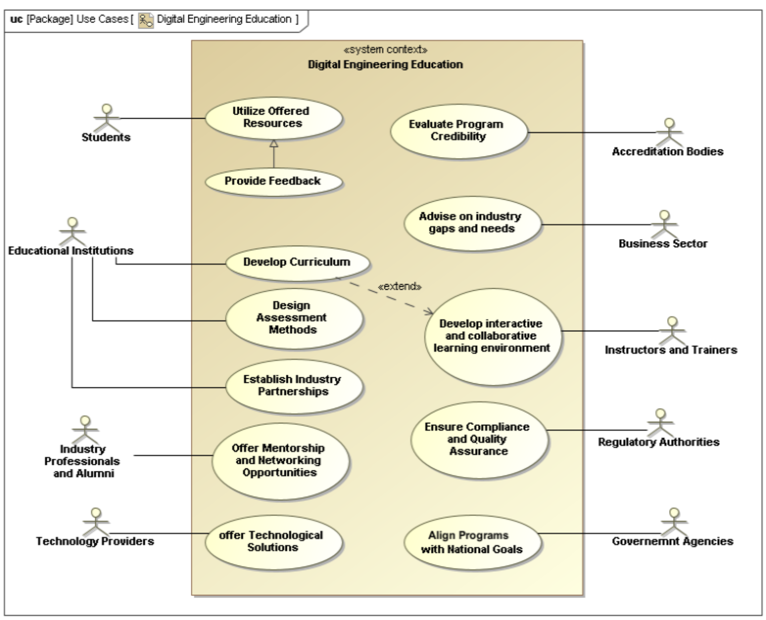


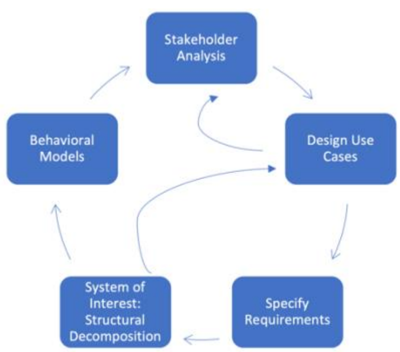

Figure 1: *Use Case Diagram describing stakeholders and use cases in Digital Engineering Educational Program development.*

SysML is designed to model complex systems including structural, behavioral, requirements and parametric aspects. The cross allocation between these aspects is crucial in complex systems. For example, the use case shown in Fig. 1 can be traced to their specifications of how these services will be provided and who is providing these services, ensuring that all requirements are met in a timely manner. The cross allocation between the different system aspects can be traced in digital format to maintain the system with future changes. In rare cases, a change in one aspect requires a simple change in one place but usually includes modifications at least in requirements, structural, and behavioral aspects. SysML is a well-suited language to manage system changes effectively. The traceability views of the cross allocated aspects allow to systematically update the model maintaining a coherent representation of the evolving system throughout its lifecycle phases. Finally, SysML is a versatile language that can be applied in different engineering and non-engineering domains providing a standardized and digitized approach to systems engineering.

## 4.0 Prototyping A Modular Approach to Digital Engineering Education

DE principles will be introduced in engineering courses using several case studies drawn from students and faculty experiences. Students are shown strategies for creating digital artifacts using stages of a MBSE framework. It is expected that students will study these online modules on their own and prepare to take on a course-assigned system design group project that will assess their skills in applying systems thinking and digital engineering principles.

The essential learning objectives, for assessing these skills are: L1: Ability to distinguish between the system to be designed and its broader system context such as its overall mission, the operational environment and users; L2: Ability to identify the sub-systems that support the system, their connectivity and feedback properties; L3: Anticipate human factors that may influence the system and mission goals and objectives; L4: Capability in applying architectural tools and related visualizations to describe their understanding of the system and its context.




The modules are drawn from the design stages depicted in Fig. 2, which include: (i) Stakeholder Analysis; (ii) Use Case Design; (iii) Requirements Specification; (iv) Structural Modeling of the System of Interest and (v) Behavior Modeling. The stages are iterative, requiring frequent stakeholder collaboration to ensure alignment with their use cases and specifications. In the interest of space, this paper presents the modular design approach proposed for the first three stages.

Figure 2: *The key stages of MBSE in system design.*

## 4.1 Architectural Tools for Digital Engineering

A common set of tools that can facilitate the design of the modules that capture the MBSE design is important if we are to scale the content and learning across other systems of interest. Towards this objective, we utilize the Unified Architecture Framework (UAF) and Systems Modeling Language (SysML) architectural languages that are part of the CATIA Magic systems engineering solutions software, licensed by Dassault Systèmes.

The UAF serves as an architectural enterprise framework and language for characterizing the strategy, operations, resources, organization and other aspects involved with the System of Interest (SoI) design. Based on [10], an enterprise consists of much more than an individual system of interest. An enterprise consists of a network of interdependent resources (e.g., people, processes, organizations, supporting technologies, and funding) that:

- interact with each other to coordinate functions,
- share information,
- allocate funding,
- create workflows,
- and make decisions,

to achieve business and operational goals through a complex web of interactions distributed across geography and time.

UAF will be employed to architect and model the enterprise elements while SysML will be employed to model the technical details of the SoI. Designed for enterprise architecture and solution architecture respectively, UAF and SysML can provide students the early experience on model-based system throughout the lifecycle phases of the solutions carried across defense, government and industry organizations.

SysML is a graphical modeling language for specification, analysis, design, verification and validation of systems. There are nine SysML diagrams that are tailored to depict the Structural, Behavioral, Requirements, and Analysis facets of the SoI [11]. In aerospace and defense, SysML serves as a standardized language for modeling complex systems. The UAF evolved from SysML by expanding its scope beyond system modeling to encompass broader aspects of the enterprise architecture. The UAF includes a set of enterprise viewpoints organized in the rows of the grid while the columns categorize the architectural aspects as shown in Fig. 3. The grid helps ensure consistency and completeness in capturing and communicating architectural information across the enterprise. An intersection of a row and a column represents a specific architectural viewpoint within a particular architectural layer helping stakeholders to effectively analyze and communicate architectural information within the framework.


Figure 3: *Unified Architecture Framework (UAF) Grid supporting a design structure.*

### 4.2: Case Study Example & Module Design

The module design approach is presented in this section, considering a scenario and system of interest that may be familiar to students. Only the first three stages of Fig. 2 are discussed in the interest of space. The module template will include sections for (i) System Context; (ii) System of Interest; (iii) Stakeholder Analysis; (iv) Use Cases; and (v) Requirements Specification. Each section will include links to additional resources that students with varied backgrounds may find useful.

**M.1 <u>The System Context:</u>** Faculty directors of an engineering research lab on a university campus are interested in increasing the visibility of their lab and its research and in engaging new student and faculty participants in their lab. They are considering various solutions, including one that is based on voice-activated and directional sound-based acoustic technology. Such technology is also aligned with their research, and they feel it can help make a connection to the lab's activities. Systems engineers have been asked to propose a solution to this problem using digital modeling.

**M.2 System of Interest (SoI):** The students take on the role of systems engineers in evaluating the proposal to use directional sound systems to increase the research lab's visibility. One solution is using a directional loudspeaker near the lab to inform passers-by with information that may generate interest and curiosity about the research facility. This directional-loudspeaker solution is referred to as the SoI.

**M.2.1** *SoI Functionality:* Directional loudspeakers produce sound in a narrow spatial region when that region intersects an object such as a human body. The operation of this system relies on amplitude modulating the message with a higher carrier frequency, causing the sound to be inaudible as it is transmitted in free space. However, when the sound field intersects an object or a human body in its directional zone, the signal is demodulated, rendering the carried signal to become audible to the human ear.

**M.3 Stakeholder Analysis Module:** Stakeholders refer to a group of individuals or entities with particular concerns that need to be fulfilled or those who are influenced by the mission of the system and its behavior [12]. Engineers should understand and integrate stakeholders' needs into their problem and solution spaces [13]. Stakeholder interests can be identified by specific domains, i.e. the rows in the UAF grid shown in Fig. 3. For example, there are distinct stakeholders defining the strategic mission and operational capabilities of the SoI.

**M.3.1** *Strategic Domain Stakeholders:* Stakeholders in this domain include those who have articulated the problem to be addressed, who deeply understand the root causes of the problem and are motivated to find solutions. For the SoI in this case study, they are the directors of the research lab who should be engaged to elicit details of their rationale, motivation, drivers, opportunities and challenges. The UAF view in the Strategic Domain includes the Motivation diagram, shown in Fig. 4, which provides the toolset for engaging these stakeholders on these topics.

Enterprise Architecture constructs such as Drivers, Challenges, and Opportunities are considered strategic elements modeled in a UAF project using CATIA magic. Sets of stakeholders including clients and target groups are modeled using *operational performers*. These entities are abstract mission participants who can perform activities in the scenario. Fig. 4 focuses on motivation for the articulated mission (shown in the diagram) from research lab directors. Given this mission or enterprise vision, drivers are used to define factors or rationale that drive the articulated mission. Each driver can then be mapped to one or more challenges which reflect issues that need to be resolved to address the driver. This dependency is expressed using the *PresentedBy* relationship. The challenges identified are used to motivate a set of opportunities expressed by the *MotivatedBy* dependency. These opportunities can be further traced to the capabilities of the SoI to achieve the proposed mission.

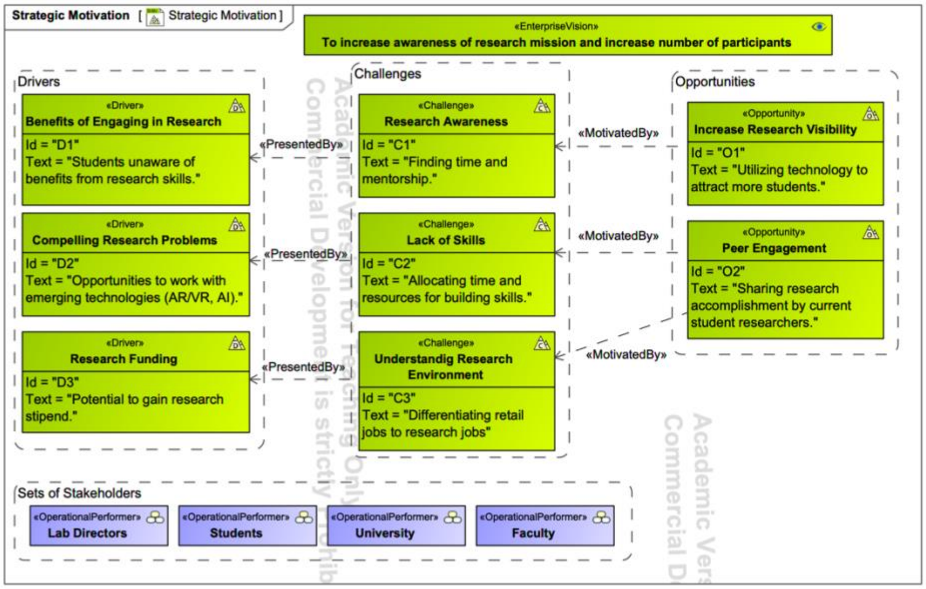



Figure *4: Strategic Motivation view capturing motivation for the mission.*

It is important that the voice of each stakeholder engaged is specifically included in appropriate locations of the diagram and that this can be traced as the system is developed. The specific language used by the stakeholder should also be recorded. This can be achieved by linking the Stakeholder Concern model element to appropriate stakeholders. Fig. 5 utilizes the Relation Map Diagram to organize this information. Connections shown here are configured to display members of the root package, named stakeholders, and the concerns related to each stakeholder. The Relation Map Diagram can be created for motivation or traceability aspects of various viewpoints to realize relationships of any model elements of the enterprise.

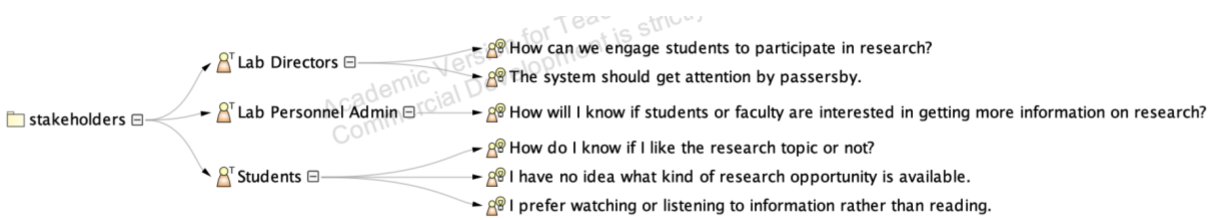



Figure *5: Example* that summarizes *key stakeholders and their concerns using the Relation Map Diagram.*

Opportunities depicted in the strategic motivation view can be used to identify capabilities. Capability in UAF is defined to be an ability that systems can offer to accomplish desired outcomes. Such capabilities and its topology are captured using the Strategic Taxonomy diagram shown in Fig. 6. A link with an open arrow on one end and a filled diamond on the other shows a *composition relationship*, indicating a parent-child relationship. A child capability emphasizes the required abilities to achieve a parent's capability. Closed unfilled arrows show a *generalization relationship*. A child model element specializes a parent’s capability.

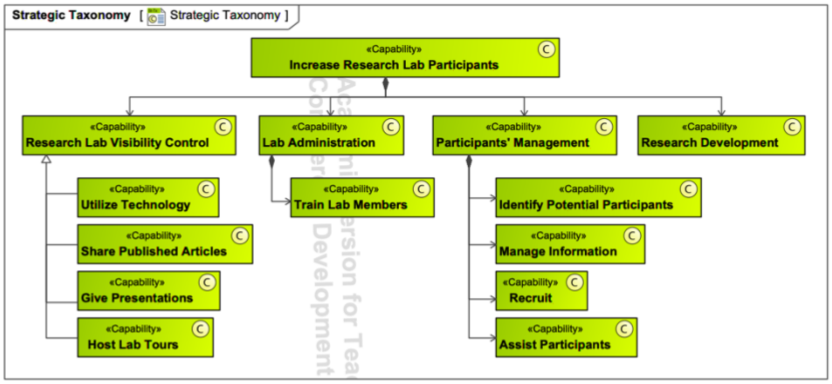



Figure 6: *Strategic Taxonomy view describing the capabilities of enterprise.*

**M.3.2** *<u>Operational Domain Stakeholders:</u>* The operational perspective of the UAF grid is implemented to realize the tasks intended to be achieved in the mission and entities responsible for performing such tasks. The stakeholders contributing to the operational architecture will typically define the requirements that will drive the enterprise's capabilities and help model its structure and behavior. Strategic stakeholders based on their role in the organization can also be involved in the operational requirements. This would be the case for the system context described here. The directors of the research lab who have defined the mission and objective in the strategic domain would likely also contribute to the operational requirements. If the strategic mission is defined by higher levels of the organization, such as by the Provost or Chancellor, their engagement in the operational domain may be more limited. However, end users affected by the SoI should always be engaged as key stakeholders in this domain.

The high-level taxonomy view shown in Fig. 7 classifies these entities and their interactions in the conceptual usage of the SoI. This specifies the context of the intended use of the SoI. Elements within this view are called *concept roles* which are abstract elements that play a role in the mission. A dotted line with an open arrow represents an arbitrary connector, helping visualize the connection among roles.

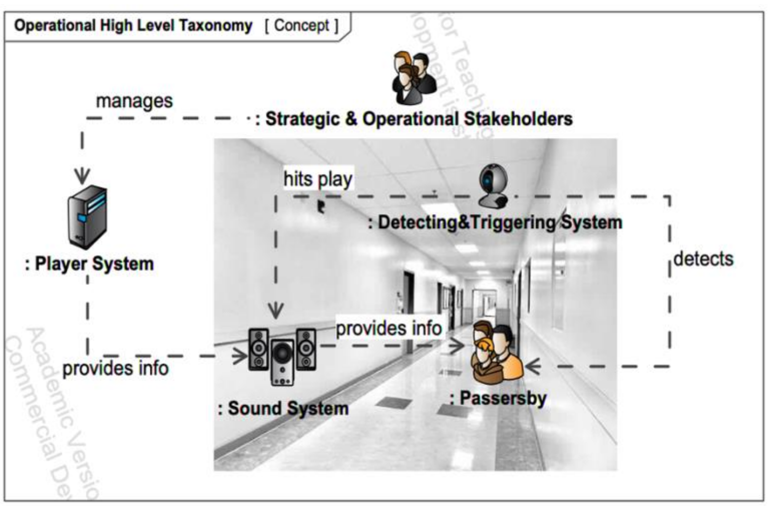



Figure 7: *Operational High-Level Taxonomy view visualizing the main operational concept.*

A further refinement of the operational taxonomy view is shown in Fig. 8 which classifies roles using the concept of *operational performers*. System designers acquire this information from the stakeholder analysis and create a taxonomy diagram to capture all personnel who have roles in interacting with the system.

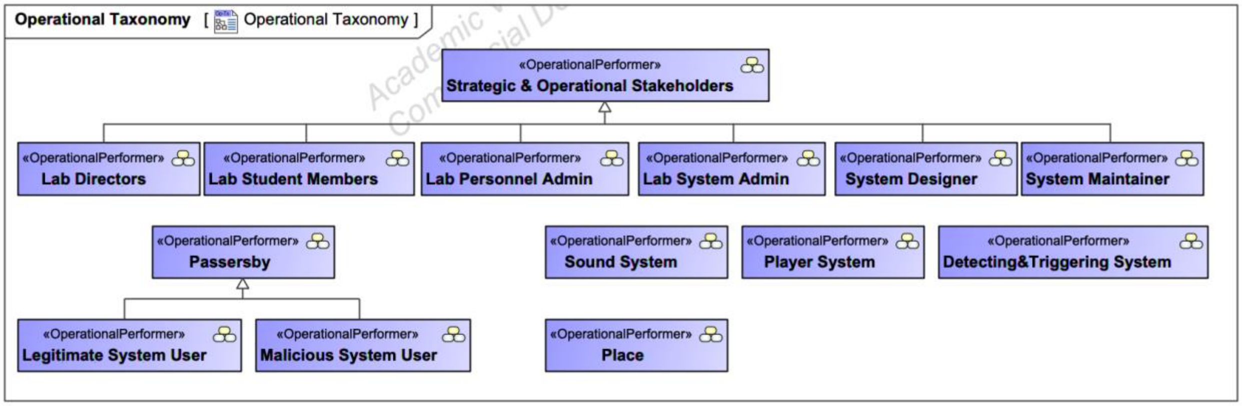


Figure *8: Operational Taxonomy view defining the architecture of operational performers, the main model entities in the operational viewpoints.*

**M.4 <u>Use Case Design:</u>** Using SysML use case diagrams, students can depict their stakeholder analysis in graphical form connecting the mission to specific scenarios of various stakeholders interacting with the systems. Such a representation is expected to provide stakeholders clarity on how engineers are translating their ongoing discussions. By learning to develop these diagrams, students also acquire the skills to present a story that captures the system users, its capabilities and outcomes. Use case diagrams also serve to continually engage stakeholders in the design process and potentially contribute to a refinement of their mission goals and objectives. In this module, students learn to define (i) Actors; (ii) System Boundary or Context; and (iii) Mechanisms of Interaction with and among use cases.

**M.4.1** *<u>Actors:</u>* People, external systems, organizations, the environment, and other entities that have specific roles in interacting with the system boundary. All personnel who have roles in interacting with the system have been captured in Fig. 8. They are used as actors in use case diagrams.

**M.4.2** *<u>System Boundary or Context:</u>* The system boundary compartmentalizes the set of use cases defining the scenario from its actors. The use cases are oval structures that have defined functions and can be linked to more detailed behavior using refined use case, state machine, activity or sequence diagrams.

**M.4.3** *<u>Mechanisms of Interaction:</u>* The relationship between elements in a use case diagram can be described using connectors that depict: (i) Communication or Association; (ii) Generalization; (iii) Includes and (iv) Extends properties. These properties can be visualized in the high-level use case diagram shown in Fig. 9.

The solid line connecting the use case and an actor is the *Association* or *Communication* relationship indicating the exchange of resources. The *Generalization* relationship parses out the required subset from the strategic stakeholder taxonomy engaged in this scenario. A legitimate system user, drawn from the Passersby group of actors interacts with the SoI and their presence is detected, and data recorded on a server. The Lab Personnel Admin is responsible for interfacing with the data server, responding to the passerby and arranging a lab tour. The Includes connector indicates sub-systems required for objects intersecting with the directional loudspeaker, storing related data and interfaces providing access to the server. The Responding System use case addresses the follow up of engaging interested passersby and includes through the *Extends* relationship an optional use case of joining a lab tour.

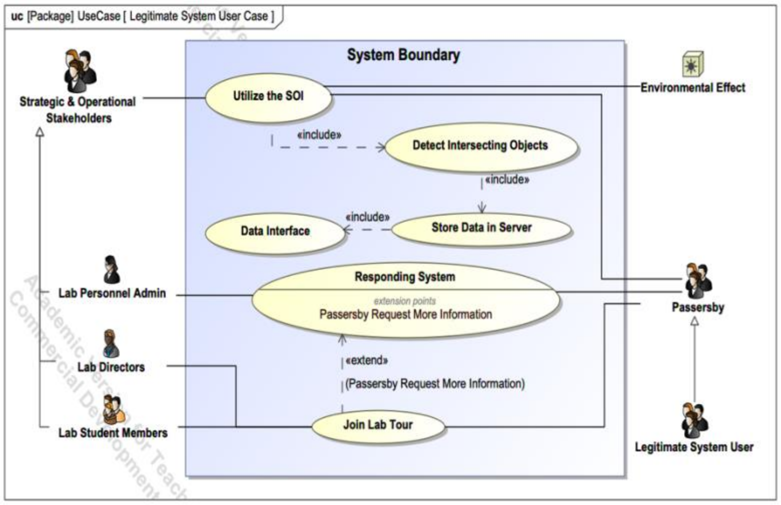



Figure 9*: Use Case Diagram describing the functionalities of the system for legitimate passersby case.*

The associations between the uses cases and stakeholders are created once in the model but can be added in different views and formats such as tables and relationship maps. In addition, these relationships can be traced, analyzed for data-driven decisions, and extractable to different platforms. The digital capture of different system views streamlines the engineering processes and fosters collaboration and communication among team members.

**M.5 <u>Requirements Analysis:</u>** Requirements engineering is a subclass of systems engineering and software engineering and is a crucial step in the design and development process of a system [14]. Requirements are conditions that must be met by the systems to satisfy the conditions and standards provided by the stakeholders.

<u>Definition:</u> A requirement is a statement that identifies a product or process operational, functional, or design characteristic or constraint, which is unambiguous, testable or measurable, and necessary for product or process acceptability [15]. Requirements should also ensure that the system performs its mission in the environment in which it must operate.

Requirement elicitation and specification is a complex effort with skills acquired over many years of systems design. This module provides students a high-level introduction to this process, with the primary learning objectives being: (R.1) Deriving a set of high-level requirements from the capabilities defined in use cases; (R.2) Classifying the requirements in a hierarchy, flowing from the SoI to its connected sub-systems, their components, parts and properties; (R.3) Labeling and organizing the requirements in SysML Requirements Diagram or Table; (R.4) Knowing how to align requirements in related UAF and SysML diagrams.

In the context of MBSE, requirements may be broadly classified at a high level as: (i) Business Requirements; (ii) User Requirements; (iii) System Functional Requirements; (iv) System Non-Functional Requirements [16].

Business requirements specify the mission, goals and objectives motivating the system of interest. User requirements pertain to system features that various stakeholders require to accomplish the system mission. System requirements characterize the functional and non-functional properties the system and its components must adhere to. Functional requirements define the operations of the SoI. Non-functional requirements capture physical attributes such as size, shape and color. They include quantitative specifications of the system functions through performance measures and design constraints. Even though they do not perform a behavioral function, they influence the functional requirements. Operational requirements, introduced in M.3.2, are clear statements of what the system should do and typically include an action on an object.

The UAF grid shows under the Motivation aspect, the Requirements views (Rq-Mv) that can be utilized to record the requirements. This allows the representation of the requirements' properties, their relation to each other and to other UAF architectural elements from different domains [17].

The use case diagram of Fig. 9 guides students to capture high-level requirements. These requirements can be organized in the SysML requirements diagram and requirements table. The mission of the strategic stakeholder serves as the top-level business requirement as shown in Fig. 10. The use of the word *shall* in a requirement indicates that the requirement should be implemented and be verifiable. From business requirements, other types of requirements are derived using a *deriveReqt* dependency, which denotes relationships such as reason-consequence, need-satisfaction, and abstract-concrete [18]. User requirements are derived from the business requirements. Both system functional and non-functional requirements are derived from user requirements. Requirements are hierarchically organized in business-user-functional/nonfunctional order. Alternatively, a tabular form may enhance the presentation of requirements to arrange a large amount of data to prevent creating a cluttered diagram. Fig. 11 demonstrates a list of functional and non-functional requirements in a table.

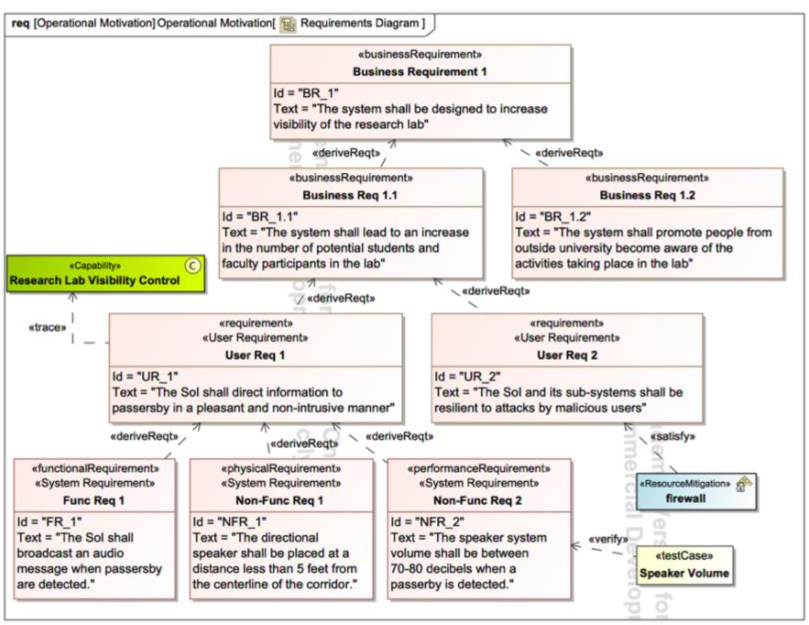



Figure *10:* Requirements Diagram classifying different types of requirements in a hierarchy and illustrating examples of relations with other model elements.

An ID is utilized to distinguish types of requirements. Business, User, System Functional, and System Non-Functional requirements are denoted by BR_X, UR_X, FR_X, and NFR_X, respectively, where X is substituted with a number. A stereotype can be also applied to achieve this distinction: for example, *businessRequirement* and *functionalRequirement*. In the CATIA Magic systems software, non-functional requirement is further categorized into stereotypes such as *usability requirement*, *interface requirement*, *performance requirement*, *physical requirement*, and *design constraint*. User-defined stereotypes, *User Requirement* and *System Requirement*, are utilized for clarification.

Requirements can be related to other model elements to describe a type of dependency. Available relationships include *satisfy*, *verify*, *refine*, and *trace*. As an example, non-functional requirement 2 and functional requirement 2 are connected to test cases *Speaker Volume* and *Detect Passersby* to indicate how these requirements are verified. To satisfy the user requirement 3, a firewall which is modeled using *Resource Mitigation*, is used. This stereotype is utilized to indicate a security measure to address cyber risk associated with systems [19]. A *trace* dependency is used to signify a connection between a requirement and other model kinds [18], which is demonstrated by associating the user requirement 1 and one of capabilities defined in the strategic diagram. Additionally, the functional requirement 3 is clarified by the operational activity, which is depicted by a *refine* dependency in the table. The operational activity model element describes logical processes required to meet the mission.

| # | Name | Text | Derived From | Verified By | Refined By |
|---|---|---|---|---|---|
| 4 | FR_1 Func Req 1 | The SoI shall broadcast an audio message when passersby are detected. | UR_1 | | |
| 5 | FR_2 Func Req 2 | The object detecting system shall detect passersby that appear to be intersected in listening to the sound recording. | UR_3 | Detect Passersby | |
| 6 | FR_3 Func Req 3 | The responding system shall direct engaged-passersby to an interface to record their information. | UR_7 | | Get Passersby Information |
| 7 | FR_4 Func Req 4 | The data server provides an interface to lab personnel admin to retrieve information from engaged users. | UR_5 | | |
| 8 | FR_5 Func Req 5 | The player system shall allow strategic stakeholders to update its content. | UR_4 | | |
| 9 | NFR_1 Non-Func Req 1 | The directional speaker shall be placed at a distance less than 5 feet from the centerline of the corridor. | UR_1 | | |
| 10 | NFR_2 Non-Func Req 2 | The speaker system volume shall be between 70-80 decibels when a passerby is detected. | UR_1 | Speaker Volume | |
| 11 | NFR_3 Non-Func Req 3 | The speaker system shall be turned off after 9 PM. | UR_1 | | |
| 12 | NFR_4 Non-Func Req 4 | The system shall be securely installed inside the research facility setting. | UR_4<br>UR_6 | | |

Figure 11: *Demonstration of organizing requirements in a tabular form containing system functional and non-functional requirements and some of their properties.*

## 5.0 Summary

Integrating systems thinking skills into engineering education has been explored in many studies and programs [20-24]. With competing learning objectives and outcomes across different engineering courses, the integration and assessment of these skills is challenging. Recent initiatives in Digital Transformation of organizational practices have led to the need to reinforce systems thinking and systems engineering skills in engineering education and those who interact with engineers in the design and use of emerging technologies and systems. Architectural tools developed for digital engineering and model-based systems engineering offer a structured approach for introducing systems thinking skills and training the future workforce for competency in digital transformation. Given the complex interdisciplinary nature of these topics, we propose a modular approach for introducing these skills in engineering courses using UAF and SysML platforms. The modules are designed to introduce and reinforce systems thinking using scenarios and systems that students are familiar with in their day-to-day lives. They are designed to be scalable across different courses and help build a foundation that can be applied to course-specific projects.

## References


[1] N. Hutchison and H. Y. S. Tao, "The Digital Engineering Competency Framework (DECF): Critical Skillsets to Support Digital Transformation," *INSIGHT*, vol. 25, no. 3, pp. 35-39, 2022, doi: 10.1002/inst.12396.

[2] L. M. C. Benavides, J. A. Tamayo Arias, M. D. Arango Serna, J. W. Branch Bedoya, and D. Burgos, "Digital Transformation in Higher Education Institutions: A Systematic Literature Review," *Sensors*, vol. 20, no. 11, p. 3291, Jun. 2020, doi: 10.3390/s20113291.

[3] Office of the Deputy Assistant Secretary of Defense for Systems Engineering, *Digital Engineering Strategy*, 2018. Available: https://ac.cto.mil/wp-content/uploads/2019/06/2018-Digital-Engineering-Strategy_Approved_PrintVersion.pdf.

[4] M. J. de C Henshaw, "Systems of Systems, Cyber-Physical Systems, the Internet-of-Things... Whatever Next?," *INSIGHT*, vol. 19, no. 3, pp. 51–54, 2016, doi: 10.1002/inst.12109.

[5] A. L. Ramos, J. V. Ferreira, and J. Barceló, "Model-Based Systems Engineering: An Emerging Approach for Modern Systems," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42, no. 1, pp. 101-111, Jan. 2012, doi: 10.1109/TSMCC.2011.2106495.

[6] M. Amissah, "A Framework for Executable Systems Modeling," Ph.D. dissertation, Eng. Manag. & Syst. Eng., Old Dominion University, Virginia, 2018, doi: 10.25777/f1h6-e712.

[7] R. D. Arnold and J. P. Wade, "A Definition of Systems Thinking: A Systems Approach," *Procedia Computer Science*, vol 44, pp. 669-678, 2015, doi: 10.1016/j.procs.2015.03.050.

[8] J. P. Monat, T. F. Gannon, and M. Amissah, "The Case for Systems Thinking in Undergraduate Engineering Education," *International Journal of Engineering Pedagogy (iJEP)*, vol. 12, no. 3, pp. 50-88, 2022, doi: 10.3991/ijep.v12i3.25035.

[9] E. B. Dano, "Introducing Systems Thinking Techniques into an Undergraduate Engineering Education," *INCOSE International Symposium*, vol. 32, no. 1, pp. 199-209, 2022, doi: 10.1002/iis2.12925.

[10] G. Rebovich and B. E. White, *Enterprise Systems Engineering: Advances in the Theory and Practice*. Boca Raton, FL: CRC Press, Taylor & Francis Group, Auerbach, 2011.

[11] Object Management Group, "OMG Systems Modeling Language." https://www.omgsysml.org/.

[12] S. Friedenthal, A. Moore, and R. Steiner, *A Practical Guide to SysML: The Systems Modeling Language*, 3rd Ed. Waltham, MA: The MK/OMG Press, Elsevier, 2015, doi: 10.1016/C2013-0-14457-1.

[13] E. Zerbe, K. Haas, and A. C. Muscalus, "Board 39: Increasing Students' Understanding of Stakeholder Perspectives: A Value-Sensitive Design Case Study," Paper presented at *2023 ASEE Annual Conference & Expositio*n, Baltimore, Maryland, Jun. 2023. Available: https://peer.asee.org/43100.

[14] I. Sommerville, *Engineering Software Products: An Introduction to Modern Software Engineering*, 1st Ed. United Kingdom: Pearson, 2019.

[15] Object Management Group, "UAF 1.2 - Appendix C: Enterprise Architecture Guide for UAF (Informative)," formal/22-07-10, 2022. [Online]. Available: https://www.omg.org/spec/UAF/1.2.

[16] N. Shevchenko, "Requirements in Model-Based Systems Engineering (MBSE)," *Carnegie Mellon University, Software Engineering Institute's Insights (blog)*, Feb. 22, 2021, doi: 10.1184/R1/14077232.v1.

[17] A. Morkevicius, "Applying Unified Architecture Framework (UAF) for Systems of Systems Architectures," *INCOSE UK*, 2018.

[18] A. Armonas, "Requirements Writing in SysML," No Magic, Inc. 2011. [Online]. Available: https://www.3ds.com/fileadmin/PRODUCTS-SERVICES/CATIA/NoMagic/pdf/cameo-requirements-modeler-plugin-requirements-writing-guide.pdf.

[19] F. Dandashi, "Modeling Security Views with Unified Architecture Framework, Risk Assessment and Analysis Modeling Language, and Systems Modeling Language," *MITRE Corporation*, McLean, VA, Jan. 01, 2022. Accessed Feb. 05, 2024. [Online]. Available: https://apps.dtic.mil/sti/pdfs/AD1173412.pdf.

[20] M. Frank, "Characteristics of engineering systems thinking - a 3D approach for curriculum content," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews),* vol. 32, no. 3, pp. 203-214, Aug. 2002. doi: 10.1109/TSMCC.2002.804450.

[21] H. L. Davidz and D. J. Nightingale, "Enabling Systems Thinking to Accelerate the Development of Senior Systems Engineers," *Syst, Engin.*, vol. 11, no. 1, pp. 1–14, 2008. doi: 10.1002/sys.20081.

[22] L. B. Sweeney and J. D. Sterman, "Bathtub dynamics: initial results of a systems thinking inventory," *Syst. Dyn. Rev.*, vol. 16, pp. 249–286, 2000. [Online]. doi: 10.1002/sdr.198.

[23] M. Frank, "Knowledge, abilities, cognitive characteristics and behavioral competences of engineers with high capacity for engineering systems thinking (CEST)," in *IEEE Engineering Management Review*, vol. 34, pp. 48, 2007. doi: 10.1109/EMR.2006.261381.

[24] T. Ram, O. Ben-Zvi Assaraf, and S. Maman, "System-thinking progress in engineering programs: A case for broadening the roles of students," *Frontiers in Education*, vol. 8, 2023. [Online]. doi: 10.3389/feduc.2023.1138503.