2023 Annual Conference & Exposition
Baltimore Convention Center, MD | June 25 - 28, 2023
The Harbor of Engineering Education for 130 Years
Paper ID #38545

A Novel Approach for Teaching System Architecture at the Undergraduate Engineering Level

Dr. Eric B. Dano, George Washington University

Dr. Eric Dano received a B.S. in Physics from the U.S. Naval Academy, and a Ph.D. in Electrical Engineering from the University of Michigan, where his research focused on radar design and the Experimental Study of the Microwave Radar Backscatter. Upon graduation, Eric worked 25 years at Sanders/BAE Systems serving as a technical director and system architect on multiple diverse defense systems. He is currently an Associate Professor of Practice in George Washington University's Engineering Management and Systems Engineering Department in Washington DC.

Dr. Joost R. Santos, The George Washington University

A Novel Approach for Teaching System Architecture at the Undergraduate Engineering Level

Eric Dano, Ph.D. and Joost Santos, Ph.D.
George Washington University, Engineering Management and Systems Engineering (EMSE) Department
ericdano@gwu.edu and joost@gwu.edu

Abstract:
Research has shown that system architecture, along with the concurrent decision analyses and requirements definition, has the greatest effect on an engineered system's cost and capabilities. The question then becomes, given that system architecture is so critically important to the success of a system, why is system architecture not taught in any detail in undergraduate education? Some would posit that system architecture does not lend itself to teaching because of its abstract nature requiring synthesis to derive a proper architecture would be difficult for undergraduates. However, there have been several successful initiatives which have introduced system thinking, a core element of system architecting, into undergraduate systems/industrial engineering programs. These undergraduate system thinking skills provide the foundation to expand upon and provide undergraduate engineers with an understanding of system architecture. The proposed learning objectives should focus on the key architectural learning points of 1) the importance of system architecture in creating effective systems; 2) how outputs of system architecture seed system development; and 3) how architecture is transdisciplinary and considers the customer's needs and system lifecycle. This approach was implemented into George Washington University's undergraduate Systems Engineering program by implementing these architecture learning objectives directly into existing undergraduate systems engineering course material. Using this approach, a significant increase in knowledge in all three architecture learning objectives was quantified by querying the recent sophomore Fundamentals of System Engineering class in a modified Likert scale based pre-/post-class survey. Findings showed that most students went from a pre-class level (1.26) equating to most students never having heard of system architecture, to a post class level (2.86) equating to the understanding and ability to explain the importance of system architecture. In addition, the importance of decision analysis courses to compliment systems architecture knowledge will be described herein.

Introduction:
With the increase in the complexity and interconnectivity of modern systems, systems engineering practitioners must have the ability to abstract their level of thinking to holistically look at a system through its elements, stakeholders, interactions between elements, and required quality factors (aka "ilities") so that a robust transdisciplinary architecture can be developed [1]. In [2], the methods and tools defined to support the synthesis of complex system architecture were defined as classical methods, modeling and simulation, formal methods and systems thinking. The first 3 items can be categorized under decision analysis methods, which are clearly required in each step of an architecture development process to determine optimal solutions and to select between architectural alternatives.

Systems thinking is defined by [3] as "A set of synergistic analytic skills used to improve the capability of identifying and understanding systems, predicting their behaviors, and devising modifications to them in order to produce desired effects." In [4], several basic systems thinking competencies were defined, including option generation, stakeholder identification, holistic thinking, finding highest level objectives, using analogies to create options, understanding system structure/scalability, and recognizing interconnections. In [5], several advanced systems thinking competencies were defined such as: the iceberg model, causal loop diagrams, behavior over time plots, stock and flow diagrams, system dynamics modeling, archetype identification, root cause analysis, systemigrams and interpretive structural models, along with a defined 10-step methodology to implement systems thinking. The actual competencies and methodology required for a given system will vary based upon the system and application type.

The defined systems thinking competencies, align completely with those required to architect a complex system [6], and are included to some degree in virtually all systems engineering programs [7]. The systems thinking competencies are also applicable to all domains, as evidenced by their incorporation into the core curriculum for various domains such as mechanical engineering [8], sustainability engineering [9], engineering management [10], and in addressing socio-cultural systems to enhance humanity [11], just to name a few. This also aligns with the INCOSE 2025 SE Vision, which states that systems engineering education must be advanced such that, "Systems thinking is formally introduced in early education. Systems engineering is a part of every engineer's curriculum and systems engineering at the university level is grounded in the theoretical foundations that spans the hard sciences, engineering, mathematics, and human and social sciences" [12]. Thus, to fully incorporate the teaching of system architecture to undergraduates requires the inclusion of systems thinking competencies, hands-on experience with various decision analysis techniques (informal and formal methods) and an introduction to an Architecture Development Method/Model Based Systems Architecture synthesis tool [1]. It should be noted that industrial engineering programs, due to their roots of creating/improving systems [13], are usually strong in teaching the undergraduates decision analysis methods, which is a critical skill for performing system architecture.

Teaching System Architecture at the Undergraduate Engineering Level:
This paper will describe the implementation of systems architecture/systems thinking elements into George Washington University's undergraduate Systems Engineering curriculum. This approach was incorporated directly into the existing ABET accredited courses, except for a Model Based Systems Architecting/Systems Engineering (MBSA/SE) course in development. The courses in Table 1 originally contained some system architecture content. This content was expanded upon, both in emphasis and content to support the following system architecture learning objectives:

1) The importance of system architecture in creating effective systems
2) How outputs of system architecture seed system development
3) How architecture is transdisciplinary and considers the customer's needs and system lifecycle

The expanded system architecture content is listed in Table 1 for each course containing architecture content. The Bloom's updated taxonomy levels [14] are also indicated to define the level of desired competency targeted for the students to achieve on the enhanced systems architecture content.

Table 1. Undergraduate Courses Which Have Implemented System Architecture Learning Objectives

Course Number	Name	Nominal Term	Content to support System Architecture Learning Objectives	Bloom's Taxonomy Level [14]
EMSE 1001	Introduction to Systems Engineering	Freshman-Fall	• Importance of system architecture in the systems engineering lifecycle • System architecture definition and introduction to an Architecture Development Method (ADM)	Remembering, Understanding
EMSE 2801	Fundamentals of System Engineering	Sophomore-Fall	• Importance of system architecture in the systems engineering lifecycle • System architecture definition and introduction to an Architecture Development Method (ADM) • Recursiveness of requirements, decision analysis methods and functional decomposition • Hands-on elementary systems architecture, design and characterization of Vex Robots based on a series of robot challenges	Remembering, Understanding
EMSE 3815	Requirements Analysis and Elicitation	Sophomore-Spring	• System Architectures key role in performing decision analysis and elicitation to define a realizable system concept and its associated requirements • Recursiveness of requirements, decision analysis methods and functional decomposition • Hands-on use of various elicitation methods to gather requirements • Hands-on requirements analysis using various analytical methods	Applying, Analyzing
EMSE 4190/4191	Senior Capstone Project I & II	Senior-Fall/Spring	• Teams apply systems thinking to holistically examine a selected problem. System modeling, dynamics, literature reviews, methodology development and multiple decision analysis techniques are employed to solve each problem. • Hands-on application of the earlier systems architecture/systems thinking skills, and selected decision analysis techniques (see Table 2) to solve a complex problem of their choice.	Applying, Analyzing (4190) Evaluating, Creating (4191)
EMSE 6801	Systems Engineering 1 – With Model Based System Architecting/ System Engineering (MBSA/SE) Modules	Undergraduate elective/ required for SE master's program	• Advanced systems engineering concepts • Importance of MBSA/SE • Creating architectures of engineered and socio-technical systems using ARCADIA Capella. • Model based tracking of requirements and parametric analysis • Introduction to SysML as used for system design, analysis and requirements.	Evaluating, Creating

Mapping of Student Outcomes in the Systems Engineering Curriculum

The EMSE faculty adopted the performance levels listed in Figure 1 when mapping the student outcomes to the SE curriculum. The course learning objectives are categorized and assessed according to three performance levels (Introduce, Reinforce, and Emphasize), as described below:

- Performance Level **Introduce** – indicates students are introduced to course learning objectives such that students should be capable of remembering and understating the concepts.
- Performance Level **Reinforce** – indicates the outcome is reinforced and students given opportunities to further apply and analyze the course learning objectives.
- Performance Level **Emphasize** – indicates that students have had sufficient practice and are capable of evaluating and creating the concepts learned in class.

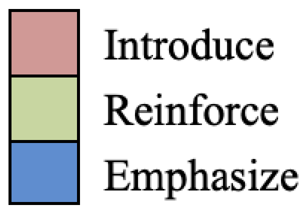


Figure 1. Performance Levels for Student Outcomes in Systems Engineering Curriculum

These performance levels are linked to the Bloom's taxonomy classification (see Fig. 2) and are subsequently used by the EMSE faculty in assessing the course learning objectives. The implementation of these performance levels has also been mapped to the appropriate levels according to a Bloom's taxonomy as illustrated by the different levels shown in Figure 2.

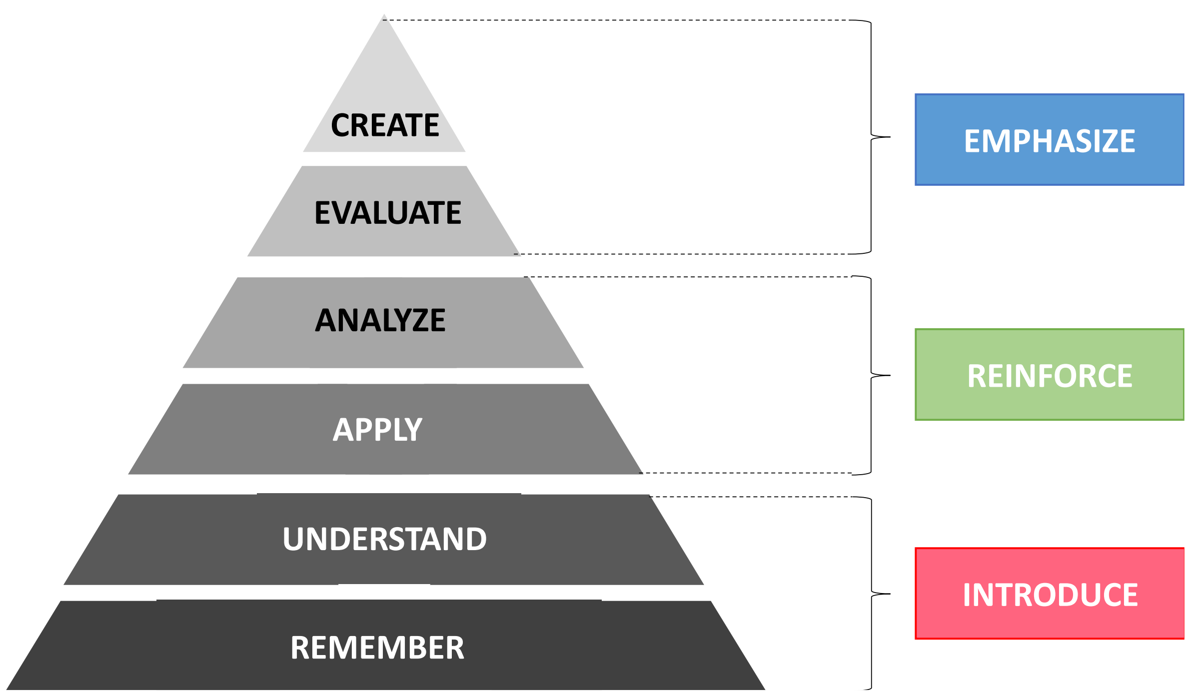


Figure 2. Relating Systems Engineering Performance Levels with Bloom's Taxonomy

Core Elements of Systems Architecture

The core elements of systems architecture, described in this section, were first taught in the EMSE 1001 course, and then reemphasized in all the later systems engineering courses along with additional in-depth systems architecture activities, culminating in the EMSE 4190/4191 Senior capstone project. This approach integrates the insertion of systems architecture knowledge in a manner consistent with the systems engineering and Bloom's taxonomy performance levels (see Figure 2).

The systems lifecycle was presented to the students as an "agile version" of the systems Vee, as shown in Figure 3.

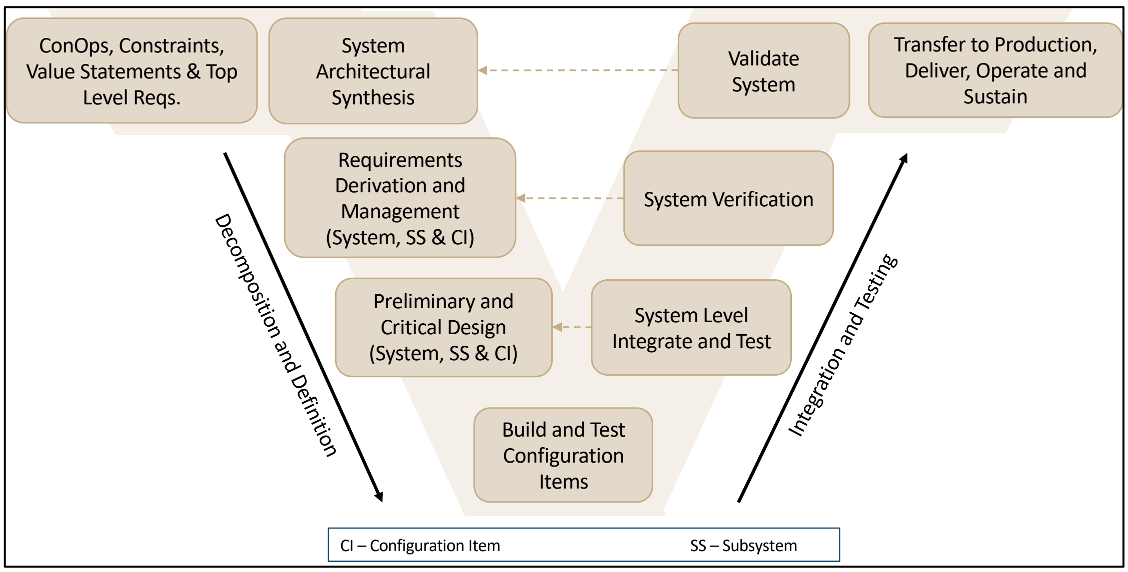


Figure 3. Agile Systems Engineering Process Vee. Note: dashed feedback lines represent early prototyping/integration, verification, and validation

The following definition of systems architecture was provided to the students:

> **System Architecture** ≡ Fundamental organization of a system embodied in its components, their relationships to each other, and to the environment, and the principles guiding its design and evolution [15].

In addition, a modified 4 step Architecture Development Method (ADM) based on [16] was presented and described to the students as the steps to take the customer's inputs, which are necessary but not sufficient, and synthesize them into a set of complete, transdisciplinary [17] concept and associated set of requirements that the system can be developed to:

- **Conceptual Architecture –** Defines elements, actors, high level capabilities/activities and exchanges between activities to meet the customer's Concept of Operation (ConOps), based on decision analysis.
- **Functional Architecture –** Performs further functional decomposition from activities to lower-level system functions, identifies exchanges between functions,

defines system boundaries, external interfaces, and external data packages, based on decision analysis.

- **Logical Architecture –** Defines internal logical elements/subsystem boundaries and allocates/aggregates functionality to logical elements. Defines internal interfaces, and data flows, based on decision analysis. Allows for definition/assessment of system use cases/threads. Note: The defined system, threads/use cases are a critical artifact used at a System Functional Review (SFR) for all US defense systems [18].
- **Physical Architecture –** Allocates/aggregates all functions to type (e.g., hardware, software, firmware, etc.) and places in logical sub-elements within each logical element. Partitions logical elements/sub-elements to physical system elements/sub-elements and performs further aggregations by common type if required (based on decision analysis). Final data and interface architectures are defined.

These ADM steps align very closely with the ARCADIA method and Capella tool [19] which will be the primary tool used in the planned MBSA/SE course to synthesize system architectures, and students will benefit from having early familiarity with the ADM at the introductory level.

Findings from [20] showing that system architecture has the greatest effect on the success of an engineered system, with 70% of a system's cost and capabilities being directly attributed to how the system was architected (including concurrent decision analysis and requirements derivation), were also presented to, and discussed with the students to emphasize the importance of performing a proper system architecture and simultaneous requirements derivation.

EMSE 1001 Introduction to Systems Analysis

In the EMSE 1001 course, the systems engineering process was a single lecture based on the NASA Systems Engineering Handbook [21]. Each step in the SE process was discussed at a high level and mapped to the simultaneous class hands-on activity of creating a paper helicopter. To achieve the systems architecture learning objectives, the course materials were first updated to also include an "agile version" of the systems Vee (see Figure 3) to directly show the key systems architecture process step and how it bridges the customer's conceptual problem space to the vendor's solution space. The 4-step ADM was also described at a high level with its importance to developing robust systems discussed with the students.

EMSE 2801 Fundamentals of Systems Engineering

The EMSE 2801 course consists of nine lectures which cover the Agile Systems Vee in a step-by-step manner, and a lean manufacturing and production simulation which takes 3 class sessions. Like the EMSE 1001 approach, the nine EMSE 2801 lectures are paired with corresponding challenges that must be performed using a VEX Robot [22]. The challenges match each lecture topic (ConOps, architecture, requirements, design, integration & test and verification/validation). The students write up their findings for each prosses step with their robot, and then submit a final report with overall findings and lessons learned at the end of the course. The LEGO® based lean manufacturing and production simulation [23] is run in a similar fashion with student teams writing up recommended improvements after each simulation (6 total) in addition to taking a lean process quiz.

Because system architecture was afforded a full lecture, the topics described under EMSE 1001 were first repeated to reinforce the student's high-level understanding of system architecture. The EMSE 2801 course built on these fundamentals and went into deeper detail on system architecture. To put system architecture in perspective, it was first described that the customer's top-level requirements, ConOps, constraints and value statements are necessary, but not sufficient to begin a system development with. It was then described how architecture synthesis is needed to take in the customer's inputs and expand upon them by working with a set of multi-disciplinary subject matter experts [17], developing alternative architectures, performing various decision analysis methods/modeling and simulation, and generating a complete set of requirements to allow the requirements derivation and development phases to begin. Further, it was also discussed how there is a set amount of system conceptual definition that needs to be performed prior to development to ensure a system that meets the customer's needs is produced. Customers with strong systems staff may produce complete ConOps and requirements, which drastically reduces the workload of the architect. Conversely, customers without strong system staff typically produce incomplete ConOps (technical and support) and requirements, which puts a burden on the architecture to "fill in the gaps."

The students were shown the architectural synthesis process using the previously described 4 step ADM, on a simple example of a new house, with details below:

- **Conceptual Architecture**
  - Identification of stakeholders/entities/organizations involved in the home building process. This emphasized the holistic thinking required to realize that beyond the actual construction staff, there are utilities, building codes, government tax offices, etc., which effect the new home.
  - Identification of the ConOps and top-level capabilities performed by the home, and decomposing the capabilities to progressively lower level activities. This emphasized how the architecture process will continue to decompose functionality and naturally generate functional requirements.
  - Allocation of the activities to the Actors/Entities with their interactions defined. This emphasized understanding the completeness of the high-level functions (activities) before defining the system boundaries in the next step.
- **Functional Architecture**
  - Definition of the system boundary. This emphasized the black box view of the system, defined the external interfaces and high-level aggregation of functionality to "in", "out" of the system. The importance of interface coordination with external elements was also discussed.
  - One of the high-level activities (distribute electricity into all rooms) was decomposed into lower-level system functions, which included "ilities." In the class example, safety/maintainability related requirements were defined

including: Provide GFE outlets near sinks, provide accessible circuit breaker panel, utilize standard circuit breakers, etc. The point was made that if the functions are not included for both operational and support (aka "ilities") elements, they will not end up in the architecture/requirements, and therefore not end up in the system design.

- **Logical Architecture**
  - Definition of the internal subsystems and their boundaries. This emphasized the glass box [16] view of the system, defined the internal interfaces and mid-level aggregation of functionality into the various subsystems.
  - The importance of identifying the system threads/use cases, mapping them into the logical architecture to show the flows through subsystems, and vetting these with the customer as an early validation activity was discussed.
- **Physical Architecture**
  - The process of allocating functions into: hardware, software, firmware, etc. as a function of heuristics, required functional performance, state of technology, decision analysis, and leverage was discussed. Similarly, the aggregation (grouping) of similar functions to provide high cohesion of the co-located functions was shown for a bathroom sink unit. Lastly, the partitioning of the aggregated functions to specific configuration items was discussed. This emphasized the white box view of the system, with defined internal/external interfaces and exchanges defined.
  - Further, it was discussed how system attributes such as modularity/growth, compliance to open standards, etc. must occur in both the system architecture partitioning step, as well as in the system design phase to be incorporated properly into the system.

In the EMSE 2801 course, system architecture was also emphasized during the hands-on lean process simulation [23]. After each of the 6 simulation segments, the teams made process and plant improvements to improve their production outputs. This facilitated a discussion on how production would be severely delayed if a project waited until production time to fine tune their production process, and how producibility (and all other ilities) need to be addressed and incorporated into the system architecture. Additionally, it was discussed how any production improvements need to be developed concurrently with the system to support optimized system production runs, and reduced system fielding times.

The system architecture knowledge gained in the 2022 offering of the EMSE 2801 course was assessed in a survey, which will be described in a later section.

EMSE 4190/4191 Senior Capstone I & II

The senior capstone combines all the undergraduate curriculum learning points into a single project to both identify and solve a real-world problem. While a full architecture development is typically not performed as part of this effort, the conceptual architecture/system thinking skills are critical in assessing their team's selected system. Most students go through a detailed stakeholder analysis, and also assess their selected system from a system dynamics point of view. The capstone project requires the use of multiple decision analysis tools and methods to process the data required to solve their chosen problem and to select between alternatives. Typical decision analysis tools include Matlab®, Minitab®, R-Studio, Microsoft Excel®, QGIS, Simio®, etc. Decision analysis methods vary project to project, but typically include methods such as system dynamics (e.g., causal loops, stock and flow diagrams, root cause analysis) [5], Principal Component Analysis (PCA) [24], Failure Mode Effects and Criticality Analysis (FMECA) [25], Input-Output analysis techniques [26], modeling and simulation [27], surveys (structured, semi-structured, expert elicitation) [28], Analytic Hierarchy Process (AHP) [29], etc. The tools and methods (see Table 2) are directly applicable to assessing the many required decision analysis items encountered during the architecture synthesis process.

EMSE 6801 Systems Engineering I with Model Based System Architecting/System Engineering

The MBSA/SE course will provide the students with the ability to synthesize and evaluate system architectures using multiple MBSA/SE tools (e.g., ARCADIA Capella [19] and a SysML based tool), and the ADM introduced in EMSE 2801. Students will work on 2 projects over the semester to both build their proficiency with the tools and their understanding of the elements of the architectural development method. The students will model actual raspberry pi-based sensor systems which they will architect and develop as part of the course. This is similar to the approach used in EMSE 2801 with the VEX robots used to provide active learning elements but will utilize a significantly more sophisticated sensors and system. Students will also have several requirements development/tracking, and modeling & simulation aspects incorporated into their projects. Although MBSA/SE is the initial focus of the EMSE 6801 course, a goal is to add in models from other domains to provide a multi-domain Model Based Engineering (MBE) environment for the students to develop in.

EMSE Decision Analysis Courses that Supplement Systems Architecture

The GWU Systems Engineering program evolved from an Industrial Engineering program and still maintains a strong decision analysis core as part of its undergraduate systems engineering degree. Table 2 shows the various decision analysis tools learned, which will greatly compliment the system architecture learning objectives and build the students system thinking skills, which will aid them in addressing the novel systems they will face once they graduate [30].

Table 2. Undergraduate Decision Analysis Courses Which Compliment System Architecture Skills

Course Number	Name	SA Learning Objectives/Methods	Analysis Tool(s) Used
EMSE 2705	Mathematics of Operations Research	• Linear equations/models • Matrix algebra • Least squared methods	Excel®
EMSE 3740	System Thinking and Policy Making	• Introduction to systems thinking • Introduction to systems dynamics (Causal Loops, Stocks and Flows, Root cause analysis, etc.) • Policy analysis	Vensim®

EMSE 3760	Discrete Systems Simulation	• Modeling of systems using discrete event simulation	Vensim®, Simio®
EMSE 3850	Quantitative Models in Systems Engineering	• Introduction to analytical models • Solving relevant system optimization problems	Excel®
EMSE 3855	Critical Infrastructure Systems	• Topics in critical infrastructure: asset management, environmental impact, input-output lifecycle analysis, reliability analysis, resilience, sustainability metrics, etc.	Excel®
EMSE 4710	Applied Optimization Modeling	• Formulation and analysis of linear, network, and integer optimization models • Linear optimization theory and algorithms	AMPL®, Excel®
EMSE 4755	Quality Control and Acceptance Sampling	• Acceptance sampling • Capacity analysis • Control charts • Design of experiments	Minitab®, Excel®
EMSE 4765	Data Analysis for Engineers and Scientists	• Estimation, confidence intervals, hypothesis testing, goodness of fit • T-squared test, multiple linear regression, Principal Component Analysis	Minitab®, Excel®

All of the previously mentioned systems engineering and decision analysis courses emphasize various system architecture/systems thinking learning objectives and are bolstered by the use of hands-on learning activities and the concurrent incorporation of multiple decision analysis methods and associated tools. The system architecture enhancements to the currently ABET accredited curriculum aligns it even stronger with the specified ABET Criterion 3 Student Outcomes, 2022-2023 shown below [31]:

1. Ability to identify, formulate, and solve complex engineering problems by applying principles of engineering, science, and mathematics.
2. Ability to apply engineering design to produce solutions that meet specified needs with consideration of public health, safety, and welfare, as well as global, cultural, social, environmental, and economic factors.
3. Ability to communicate effectively with a range of audiences.
4. Ability to recognize ethical and professional responsibilities in engineering situations and make informed judgments, which must consider the impact of engineering solutions in global, economic, environmental, and societal contexts.
5. Ability to function effectively on a team whose members together provide leadership, create a collaborative and inclusive environment, establish goals, plan tasks, and meet objectives.
6. Ability to develop and conduct appropriate experimentation, analyze and interpret data, and use engineering judgment to draw conclusions.
7. Ability to acquire and apply new knowledge as needed, using appropriate learning strategies.

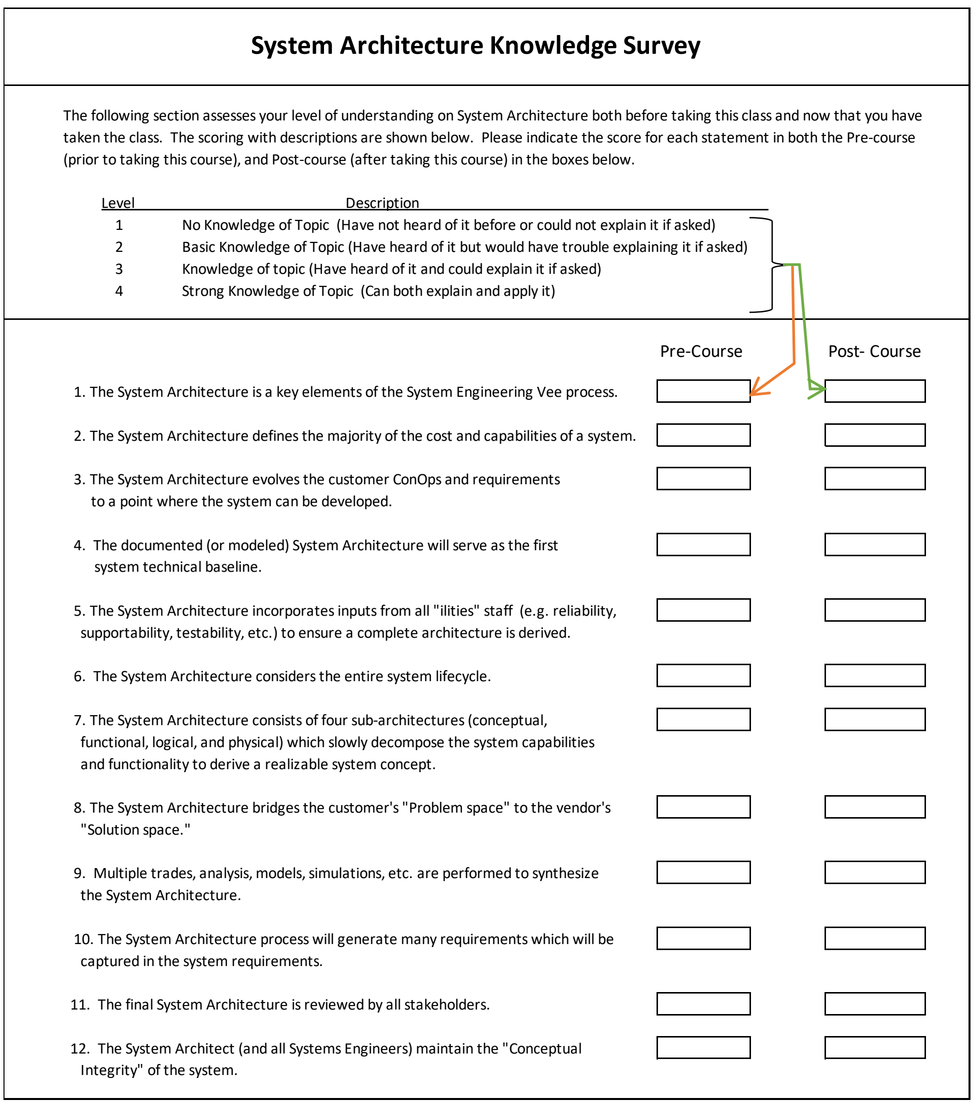
System Architecture Knowledge Growth Survey:

To test the premise that system architecture can be incrementally taught and understood by undergraduate students, a structured survey was given to the 2022 EMSE 2801 Fundamentals of Systems Engineering class. The survey was given at the end of the semester, and it was explained to the students that this was to assess knowledge growth in the topic of system architecture for the purpose of future course offerings. It was also explained that this did not affect a student’s grade, that there was no wrong answer, and to put the rating that first comes to mind. As can be seen in

Figure 4, the students just entered their pre-course and post-course assessment of their knowledge level based on the following modified Likert scale:

Level	Description
1	No Knowledge of Topic (Have not heard of it before or could not explain it if asked)
2	Basic Knowledge of Topic (Have heard of it, but would have trouble explaining it if asked)
3	Knowledge of topic (Have heard of it and could explain it if asked)
4	Strong Knowledge of Topic (Can both explain and apply it)

The survey, shown in Figure 4, covers the three stated system architecture learning objectives, with each learning objective decomposed into four related questions placed into the survey. Each student used the modified Likert scale above to score their pre-course assessed competency for all 12 questions (orange arrow), and then for the post-course assessed competency (green arrow) in figure 4. The 2022 class consisted of 11 students who all completed the survey.


System Architecture Knowledge Survey

The following section assesses your level of understanding on System Architecture both before taking this class and now that you have taken the class. The scoring with descriptions are shown below. Please indicate the score for each statement in both the Pre-course (prior to taking this course), and Post-course (after taking this course) in the boxes below.

Level	Description
1	No Knowledge of Topic (Have not heard of it before or could not explain it if asked)
2	Basic Knowledge of Topic (Have heard of it but would have trouble explaining it if asked)
3	Knowledge of topic (Have heard of it and could explain it if asked)
4	Strong Knowledge of Topic (Can both explain and apply it)

	Pre-Course	Post- Course
1. The System Architecture is a key elements of the System Engineering Vee process.		
2. The System Architecture defines the majority of the cost and capabilities of a system.		
3. The System Architecture evolves the customer ConOps and requirements to a point where the system can be developed.		
4. The documented (or modeled) System Architecture will serve as the first system technical baseline.		
5. The System Architecture incorporates inputs from all "ilities" staff (e.g. reliability, supportability, testability, etc.) to ensure a complete architecture is derived.		
6. The System Architecture considers the entire system lifecycle.		
7. The System Architecture consists of four sub-architectures (conceptual, functional, logical, and physical) which slowly decompose the system capabilities and functionality to derive a realizable system concept.		
8. The System Architecture bridges the customer's "Problem space" to the vendor's "Solution space."		
9. Multiple trades, analysis, models, simulations, etc. are performed to synthesize the System Architecture.		
10. The System Architecture process will generate many requirements which will be captured in the system requirements.		
11. The final System Architecture is reviewed by all stakeholders.		
12. The System Architect (and all Systems Engineers) maintain the "Conceptual Integrity" of the system.		

Figure 4. System Architecture Knowledge Survey given to the Sophomore Fundamentals of System Engineering Class to Assess Pre– And Post-Course System Architecture Knowledge Level.

The mapping of the lower-level questions to the system architecture learning objectives and survey results are shown in Table 3. Average scores for each question (yellow columns), each learning objective (blue columns) and average score over all questions are shown for both pre- and post-course assessed competencies using the described modified Likert scale. As could be expected at the Sophomore level, the students pre-course average score over all architecture questions of 1.26 indicates that most students had never heard of system architecture prior to the class with some having heard of it before, but not the ability to explain it (note: the EMSE 1001 course was not updated for these students prior to their freshman year).

After enhancing the system architecture content in the EMSE 2801 course as previously described, the students assessed their post-course knowledge at an average score over all architecture questions of 2.83. This indicates that the student's understood the importance of system architecture and could mostly explain why if asked but did not have the ability to apply their knowledge and perform the architectural synthesis of a system. This is the motivation for the EMSE 6801 MBSA/SE course to further reinforce student's systems architecture knowledge and to have them use MBSA/SE tools to architect systems and advance their systems architecture competency.

Table 3. Summary of Findings for the System Architecture Knowledge Level Survey

System Architecture Learning Objectives	Survey Question	Avg. Score Prior to Taking Course	Avg. Score After Taking Course	Avg Category Score Prior to Course	Avg Category Score Post Course
1) Importance of system architecture in creating effective systems	1. The System Architecture is a key elements of the System Engineering Vee process.	1.27	3.00	1.16	2.91
	2. The System Architecture defines the majority of the cost and capabilities of a system.	1.09	2.91		
	3. The System Architecture evolves the customer ConOps and requirements to a point where the system can be developed.	1.09	3.09		
	8. The System Architecture bridges the customer's "Problem space" to the vendor's "Solution space."	1.18	2.64		
2) How outputs of system architecture seed system development	4. The documented (or modeled) System Architecture will serve as the first system technical baseline.	1.18	2.55	1.30	2.73
	7. The System Architecture consists of four sub-architectures (conceptual, functional, logical, and physical) which slowly decompose the system capabilities and functionality to derive a realizable system concept.	1.09	2.82		
	9. Multiple trades, analysis, models, simulations, etc. are performed to synthesize the System Architecture.	1.55	2.64		
	10. The System Architecture process will generate many requirements which will be captured in the system requirements.	1.36	2.91		
3) How architecture is transdisciplinary and considers the customer's needs and system lifecycle	5. The System Architecture incorporates inputs from all "ilities" staff (e.g. reliability, testability, etc.) to ensure a complete architecture is derived.	1.27	2.73	1.32	2.84
	6. The System Architecture considers the entire system lifecycle.	1.18	3.18		
	11. The final System Architecture is reviewed, verified and validated by all stakeholders.	1.36	2.82		
	12. The System Architect (and all Systems Engineers) maintain the "Conceptual Integrity" of the system	1.45	2.64		
	Avg. Score all Categories:	1.26	2.83		

Conclusion:

This paper detailed an approach of teaching the important topic of system architecture at the undergraduate level. The approach was centered on expanding the system architecture content in existing classes, combined with active learning elements, to rigorously present the key system architecture learning objectives of: 1) the importance of system architecture in creating effective systems, 2) how outputs of system architecture seed system development, and 3) how architecture is transdisciplinary and considers the customer's needs and system lifecycle. Details of the implementation into 5 current undergraduate courses in George Washington University's Systems Engineering degree program were described, along with a description of a new Model Based Systems Architecture/Systems Engineering (MBSA/SE) course which will give students hands on architecture experience using MBSA/SE tools and methodologies to architect a complex system. These skills, when combined with the multiple data analytics/decision analysis courses mentioned, will produce systems engineers with a thorough understand of what system architecture is and its importance in complex systems. This approach was validated by a survey performed on the Sophomore EMSE 2801 Fundamentals of System Engineering course which showed knowledge growth from little to no knowledge of the 3 system architecture learning objectives prior to the course, to strong basic knowledge of the learning objectives after the course. The survey will be given in the future to the other undergraduate courses discussed to define the student's total knowledge growth over time, and alignment with the stated objectives. Although the novel element presented herein is the introduction of system architecture into an undergraduate education, the methodology described can be extended into other domains, as it has been for general systems thinking, another key element of system architecture.

References:

[1] E. Dano, "Augmented ARCADIA Capella based approach for the system architecture of engineered systems," Presented to INCOSE Canada, Feb. 13, 2023, website: (https://incose-org.zoom.us/rec/play/qK0_yMxsAgwT5GZEEVwggAltKpDmQJk5NBM0AhCMzrE6-KOLNozMAIvUxqLVWkn5BKTY3iair6zxiZFx.mZjttg633nJRGoTs?continueMode=true&_x_zm_rtaid=liVYNxq9T3aSYjYJFK5FOg.1677558111981.d3e6fc0a4ca18c2e12583257a0238e66&_x_zm_rhtaid=581).

[2] E. Dano. "System architecture, the missing piece of engineering education," *Proceedings of the Mid-Atlantic Section (MAS) ASEE Fall Conference,* Philadelphia, PA., Sept. 2021.

[3] R. Arnold and J Wade, "A definition of systems thinking: A systems approach," *Proceedings of the 2015 Conference on Systems Engineering Research, Procedia Computer Science*, Hoboken, NJ, USA, Mar. 17-19, 2015, J. Wade et. al. Eds. Elsevier Procedia Publishing, vol. 44, pp. 669-678, 2015.

[4] A. Madni, 'Generating novel options during system architecting: psychological principles, systems thinking, and computer-based aiding', *Systems Engineering*, vol. 17, no. 1, pp. 1-9, 2014.

[5] J. Monat, J & T. Gannon, *Systems volume 8: using systems thinking to solve real-world problems*, Lawson H, Wade, J & Hofkirchner, W (eds.), College publications, U.K., 2017.

[6] M. Maier and E. Rechtin, *The art of systems architecting, 3rd ed.*, CRC Press, New York, NY, 2009.

[7] M. Brannon, Z. Szajnfarber, and T. Mazucchi, "Introducing the fundamentals of systems engineering to freshmen through various interactive group activities," *Proceedings of the 121st ASEE Annual Conference and Exposition*, Indianapolis, IN., June 2014.

[8] K. Muci-Kuchler, C. Birrenkott, M. Bedillion, M. Lovett and C. Whitcomb, "Integrating systems thinking and system engineering concepts in a Freshmen-level mechanical engineering course," *Proceedings of the 127th ASEE Annual Conference*, (Virtual), June 2020.

[9] K. Hiller, S. Remington and C. Armstrong, "Assessing system thinking skills in two undergraduate sustainability courses: A comparison of two strategies," *Journal of Sustainability Education*, Vol. 3, March 2012.

[10] S. Furterer, "Applying systems engineering tools to teach systems engineering in an engineering management program," in *Proceedings of the 125th ASEE Annual Conference and Exposition*, Salt Lake City, UT., June 2018.

[11] G. K. Conkol, "Humanitarian engineering - emerging technologies and humanitarian efforts," in *Proceedings of the 2012 IEEE Global Humanitarian Technology Conference, HTC 2012, Seattle, WA, USA, 2012, pp. 253-258*.

[12] B. Beihoff, C. Oster, S. Friedenthal, C. Paredis, D. Kemp, H. Stoewer, D. Nichols and J. Wade, *A World in Motion – Systems Engineering Vision 2025*, INCOSE, San Diego, CA, (US), p. 45, 2014.

[13] J. Fraser and A. Gosavi, "What is systems engineering," in *Proceedings of the 2010 ASEE Annual Conference and Exposition,* Louisville, KY, June 2010.

[14] D. Krathwohl, B. Bloom and B. Masia, *Taxonomy of educational objectives: The classification of educational goals. Handbook II: The affective domain*. New York: McKay, 1964.

[15] IEEE, *Systems and software engineering — System life cycle processes*, IEEE Standard 15288™, 2008.

[16] H. Sillitto, *Architecting systems – Concepts, Principles and Practice*, College Publications, ISBN 978-1-84890-154-4, 2014.

[17] A. M. Madni, *Transdisciplinary Systems Engineering - Exploiting Convergence in a Hyper-Connected World*. Cham, Switzerland: Springer International Publishing, 2018.

[18] IEEE, *IEEE standard for technical reviews and audits on defense programs*, IEEE Standard 15288.2™, 2014.

[19] J. Voirin, *Model-based System and Architecture Engineering with the Arcadia Method*, Elsevier Publishing, ISBN 978-0-0810-1794-4, 2018.

[20] D. Walden (ed.) et. al. 2015, *Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities*, Version 4.0., INCOSE, San Diego, CA (US).

[21] NASA, *Systems Engineering Handbook*, NASA publication SP-2007-6105 Rev 1, Dec 2007.

[22] T. Robinson and G. Stewardson, "Exciting students through VEX robot competitions," in Technology and Engineering Teacher, 72(2), pp. 15-21, 2012.

[23] H. McManus, E. Rebentisch, E. Murman, and A. Stanke, "Teaching lean thinking principles through hands-on simulations," in *Proceedings of the 3rd International CDIO Conference*, Cambridge, Massachusetts, June 11-14, 2007.

[24] H. Abdi and L. Williams. "Principal component analysis," Wiley interdisciplinary reviews: computational statistics, Vol. 2.4, pp. 433-459, 2010

[25] H. Hwang, K. Lansey, and R. Quintanar, "Resilience-Based Failure Mode Effects and Criticality Analysis for Regional Water Supply System," Journal of Hydrodynamics, IWA Publishing, Vol. 17, Issue 2, March 2015, pp193-210.

[26] Y. Okuyama and J. Santos, "Disaster impact and input-output analysis," in *Economics Systems Research*, Vol. 26, Issue 1, 2014.

[27] R. Burkhart, P. Coleman, R. Dreisbach, F. Popielas and H Tummeschiet, *What is systems modeling and simulation?*, NAFEMS/INCOSE publication (https://www.nafems.org/publications/resource_center/wt10/), March 2019.

[28] F. Fowler Jr., *Survey Research Methods (Applied Social Research Methods), Fifth Edition*, Sage Publishing, Thousand Oaks, CA., 2014.

[29] R. Saaty, "The analytic hierarchy process – What it is and how it is used," in *Mathematical Modeling*, Vol. 9, No. 3-5, pp 161-176, 1987.

[30] D. Padhi, P. Chavan and R. Mitra, "Understanding systems thinking from the perspectives of experience and diversity," in *Proceedings of the IEEE Tenth International Conference on Technology for Education (T4E)*, Chennai, India, pp. 122-125, 2018.

[31] ABET, "Criteria for Accrediting Engineering Programs, 2022 – 2023," https://www.abet.org/accreditation/accreditation-criteria/criteria-for-accrediting-engineering-programs-2022-2023/, accessed Jan 22, 2023.