Paper ID #35243

System Architecture, the Missing Piece of Engineering Education

Dr. Eric B. Dano, BAE Systems

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 from Transient Deep-Water Breaking Waves. He has worked at Sanders Associates/BAE Systems for over 23 years, where he has built on his experience as a former USMC Electronic Countermeasures Officer to architect and design a myriad of military systems. Eric currently serves as a Technical Director in the CEMA business area and leads the System Architecture Community of Practice where he is tasked with improving the practice of System Architecture of complex systems in the Electronics Systems business sector. This role involves both the development and teaching of the four course system architecture curriculum to staff engineers. Dr. Dano is a member of Sigma Pi Sigma, ASEE, a Senior Member of IEEE, President of the NE Chapter of INCOSE, and represents BAE Systems on the NDIA MOSA committee.

System Architecture, the Missing Piece of Engineering Education

Eric B. Dano, Ph.D.
BAE Systems, Electronic Solutions
eric.b.dano@baesystems.com

Abstract:

It is unlikely that anyone would allow for the development of a complex system such as a house, ship, aircraft, or even more abstracted systems such as a healthcare or humanitarian aid network, without first painstakingly determining the high level conceptual design and required top level functionality. From there, several holistic system thinking techniques would be applied to define lower level functionality, system constraints, stakeholders, external interfaces and key system properties. The lower level functionality would then be aggregated and partitioned into the defined system elements. This series of key engineering activities constitutes the major elements of system architecture, which is an essential predecessor to any successful engineering effort, especially as the complexity of systems/systems of systems and socio-technical systems continue to grow.

Unfortunately, these architecture-centric activities and system thinking techniques are not typically part of an engineering curriculum. Undergraduate academics are so filled with core courses and humanities that domain learning is primarily limited to the upper class years, leaving little room for system architecture. Noticing the gap in system architecture education, several universities have recently started offering architecture related graduate degrees/certificates. However, with very limited graduates going to any one organization, this alone is insufficient to truly seed the architecture domain knowledge required in an organization. Clearly, university based architecture education alone, is not enough. The architectural education process must be continued into organizations where it can reach all engineers, whether they are serving in an architecture capacity, or in a design capacity where they should know what artifacts to expect from the architect to ensure a successful effort.

This paper will address how both the undergraduate academic curriculum and organizational training can be improved to provide the requisite system architecture and domain knowledge required for complex systems. Rather than expanding the number of undergraduate academic courses, it is recommended that the existing courses implement the use of multiple system thinking techniques to develop the core architecture-centric thinking skills which can be applied to multiple domains. Organizations will continue the architecture-centric education through domain specific training, and the introduction of architecture development processes and tools. With these measures in place, organizations will be much better suited for the development of a diverse set of complex systems well into the future. Examples of the proposed approach as applied to a defense system, and a humanitarian aid system will be used throughout this paper to emphasize the broad applicability.

Complex Systems Drive a Need for Robust Architectures:

From [1], a complex system is, “one made up of a large number of parts that interact in a nonsimple way.” Even in a simple system the combined system functionality is greater than the sum of its individual parts. This truism is significantly magnified in complex systems to the point where it becomes difficult to predict the emergent behavior of the complex system, whether it be desired (good) or undesired (bad). This is especially true in humanitarian socio-technical and socio-economic systems which combine interactions among multiple disparate systems and need to take into account elements such as logistics, supportability, applicable laws, politics, and human-system interactions, as well as the specific technologies used [2]. For this paper, a humanitarian system/project is defined as, "an effort that has as its objective, the improvement of a person or community [3]."

The techniques to reduce complexity to the point where an achievable concept may be defined, are rooted in the practice of system architecture. From [4], architecture is defined as, "an abstract description of the entities of a system and the relationship between those entities." The architecture derivation starts by defining the functionality required to meet the higher level concept across all technical, social and economic domains. The architectural process then derives the physical/logical instantiation of the system from the derived lower level functionality (i.e. Form follows Function [5]) and boundary properties. The system complexity can be reduced, and potentially the emergent properties identified, by using multiple architectural tools and techniques. In [6], system thinking, checklists, modeling of design alternatives, sensitivity analysis, etc. are just a few of the architectural tools available to generate novel options for complex systems. In [7], it is further pointed out how architects must take a transdisciplinary approach to option generation in order to address socio-technical and socio-economic complex systems.

Architecture Tools and methods for Addressing Complex Systems

It is debatable on how to best categorize the various tools and methods used in the system architecture process. For the purposes of this paper, they are categorized as: classical methods, system thinking, modeling and simulation, and formal methods, as shown in Table 1. The list is by no means exhaustive, and it is acknowledged that some listed techniques could be placed in multiple categories.

Table 1. System Architecture (SA) Tools and Methods for Complex Systems

SA Tool	SA Method	Ref.
Classical Methods	• Understanding the scientific method/trades/analyses • Performing sensitivity and error analysis • Performing functional decomposition/system synthesis • Performing architectural artifact generation	[8], [9] [6] [10], [11] [4], [6]
System Thinking	• Option generation/brain storming/conceptual Design • Perform holistic thinking/finding highest level objectives • Incorporating multi-stakeholder viewpoints • Ability to focus on elements providing highest value • Using heuristics/design patterns • Using analogies to create options • Understanding system hierarchical structures, elements, and complex interactions between them	[4], [6] [4], [6] [4], [6] [4], [6] [5], [12] [1], [6] [7]
Modeling & Simulation	• Implementing System/Physics/Environment Models • Using Modeling Languages and Frameworks (e.g. SysML, UML, etc.) • Performing Monte Carlo analysis/Analytics • Creating System Simulations	[13] [13] [13] [13]
Formal Methods	• Using Deterministic methods w/ Invariant Contracts • Using Stochastic methods w/ Flexible Contracts	[14], [15] [14], [15]

The call for training in the system architecture related tools and methods listed in Table 1 is not new. In 1994, Shenhar [16] proposed a systems engineering education curriculum consisting of: basic studies (math, computer science, etc.), cross-disciplinary studies (hardware, software, etc.), engineering systems and technologies (introduction to a wide set of diverse systems), management studies (operations, project management, production, etc.) and system engineering concepts (architecture, holistic and system thinking, case studies, etc.). Similarly, Walther and Radcliff [17], in their study on competency gaps between academia and industry, surmised that university curricula should teach a more holistic view of engineering which will lead to "a central competence which could be applied to solve different problems in different situational contexts." The broadness of applicability of the architectural tools and methods are evidenced by their use in such disparate domains such as innovation and entrepreneurship [18], in the development of complex systems of systems [19], and their criticality in addressing socio-cultural systems to enhance humanity [3], [20], just to name a few examples.

While all the categories in Table 1 are important, they will lend themselves to education at multiple levels. Classical methods, as their name implies, are deep rooted in the scientific method, universally applicable, and are core to all science and engineering curricula. Fundamental elements such as the ability to: 1) observe a problem, 2) research the problem, 3) create a hypothesis, 4) test a hypothesis and 5) validate a hypothesis based on data, should be taught throughout early education, along with basic math , science and humanities. More advanced concepts such as formal trade-offs, analyses and sensitivity studies are more geared for the undergraduate university education. The art and science of synthetically deriving a system architecture, including principals of functional decomposition, allocation, aggregation and partitioning of a system will require some engineering and domain experience [16] and lends itself to organization level training, or graduate level education.

Modeling and simulation have also been around for decades [13] and are well rooted in the sciences and engineering education. Relatively newer elements such as Model Based Engineering (MBE) and digital engineering are rising in popularity and are typically taught in current undergraduate engineering degree programs and refined when engineers go on to apply their modeling and simulation skills in an organization. Formal methods [14], [15] yield greater insight into complex systems and provide the ability to quantify complex system properties (e.g. resiliency [21]) but are arguably beyond the scope of an undergraduate education.

Systems thinking, as defined by [22] 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," covers most of the "art" of systems architecture and is critical in the development of a successful system. The scientific method, math and humanities taught in early education are a prerequisite to the critical holistic systems thinking skills which should be incorporated into the university curricula and then grown though a practitioner's work experience over his/her career. However, currently systems thinking is woefully lacking in undergraduate educations. The INCOSE 2025 SE Vision 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 [23]." In the following section, a proposed implementation of system thinking education into undergraduate courses will be described. This will serve as a major part of the desired system architecture tools and methods learning required by practitioners to architect and develop complex systems.

Incorporating System Thinking into University Curricula

A survey of university curriculums offering Systems and/or Industrial engineering bachelor's degrees [24] shows little signs of the inclusion of system thinking concepts into the curriculum. A question may be asked if this is because system thinking does not lend itself to teaching, or because human practitioners are unable to grasp the complexity of modern systems, and therefore a "natural reductionist" approach is taken [25]. This question was investigated as part of the pioneering work of Kordova and Frank [26] who surveyed multiple engineers to determine their capacity for system thinking. Their findings found that system thinking could be taught but will tend to develop slowly in practitioners. These findings were confirmed in [27] who found that the system thinking development time was more correlated to the diversity of systems a practitioner was exposed to, than time spent on a given system. In [25], it was also found that systems thinking could be taught but the authors pointed out that the current educational system is an inhibitor to systems thinking education because schools are structured for students to find the correct answer over possibility exploration, and for memorization over problem solving. These issues are addressed by the proposed approach.

Before discussing how to best incorporate systems thinking into a university education, the specific competencies of systems thinking [22], and descriptions are summarized from multiple sources in Table 2. The tools and methods listed in Table 1 will be used by the practitioner as they demonstrate the listed system thinking competencies.

Table 2. System Thinking Competencies and Elements

System Thinking Competency [22]	Element Description	Ref.
Recognizing interconnections	• Ability to "identify key connections between parts of a system" • Understanding the interrelationships between elements of a complex System/System of Systems	[22] [26], [27]
Understanding feedback	• Understanding feedback processes in the system (delays, feedback loops, etc.) and how they affect the system behavior	[22], [26], [27]
Differentiating types of stocks, flows, variables	• Understanding the inflows, outflows in the system, and how the outflows effect the system's utility • Understanding on "how systems are capable of producing the desired functions"	[22] [27]
Understanding system structure	• Understanding system structure (elements and interactions between elements) as the cause of system behavior • Understanding problems are treated as part of the system • Understanding systems, the different elements in them, their functionalities, processes, multi-discipline stake holders, etc. • Understanding the view of wholeness about a system (i.e. system is more than the sum of its parts due to emergence)	[22] [25] [20], [27] [28]
Understanding Dynamic Behavior	• Understanding how the system's emergent behavior comes from the complex interactions of its constituent elements (stocks, flows, variables, non-linear relationships) and the ability to predict emergent behavior • Ability to see system patterns of change, vice a static snapshot of the system	[22], [26], [27] [28]

System Thinking Competency [22]	Element Description	Ref.
Understanding Non-Linear Relationships	• Understanding and identifying non-linear elements of the systems and how they affect the system behavior • Improvements to one part of the system may have negative effects on other parts of the system	[22] [25]
Reducing Complexity	• Ability to "conceptually model different parts of a system and view a system in different ways" • Ability to develop mental models for complex systems • Understanding that systems need to be resilient • Understanding that the system hierarchy needs to be able to grow in complexity (typically bottoms up)	[22] [26] [27] [28]
Understanding Systems at Different Scales	• Ability to handle socio-cultural systems with "irreducible complexity" due to diverse interconnected human actors, as well as political and social constraints • Understanding how to define all system stakeholders, their interrelationships, and how to communicate with them • Ability to recognize different scales of a system, and systems of systems (engineered, organizational, human, etc.)	[20] [20] [22], [25]
Analytic Skills and the Scientific Method	• Ability to use mathematical and statistical tools of analysis • Ability to perform research on socio-technical topics • Ability to quantify relationships, perform trades/analysis, make data driven decisions, perform error analysis, etc. • Ability to use a broad range of concepts, principles, models, methods and tools	[16] [20] [26] [28]

From Table 2, the required competencies to cover systems thinking, may seem daunting. However, there are many methods to implement the system thinking learning activities into an undergraduate engineering curriculum. Clearly there needs to be a lecturing/mentoring component [17] [26], [27], however system thinking is best taught by hands-on projects [17], [25], [26], [27], [28], case studies [16], [25], Experimental Systems Engineering (ESE) [25], [29], and exposure to the various multi-discipline system stakeholders [7], [20], [27]. It is worth pointing out that [30] found in their system thinking curriculum that many undergraduate students did not have the prerequisite skills on conducting background research. The same deficiency could be said for trades/analysis, making data driven decisions, performing error analysis, etc. These items must be incorporated early and often into the undergraduate education to allow systems thinking to progress.

The lectures, projects, case studies and ESE should be integrated directly into all science and engineering undergraduate courses [27] so that system thinking on a particular topic/domain can be learned and then assessed against past system thinking domains to build a holistic "system thinking catalog" to draw from in future applications. The students projects should purposely be selected to cover multiple domains, and include both well-defined systems, and ill-defined systems such as complex socio-technical systems (e.g. providing food and clean water to the impoverished communities), and Systems of Systems. By taking on ill-defined systems that may not have a single correct solution, the practitioners will more rapidly shape their system thinking competencies and holistic thinking skills [20].

A good example of an implementation in an undergraduate education approach similar to the proposed is the US Military Academy at West Point. For all non-engineering students they instituted a 3 course approach where system thinking, data analysis, problem modeling and decision making are taught through lecture and by having the students work on various case studies solving wicked problems such as the Syrian refugee problem [30]. One could argue that the military academies have the flexibility to reallocate time devoted to military training and leadership development, and that their approach could not be repeated at civilian universities. I stipulate that it is still possible to implement the system thinking education, without a significant curriculum overhaul, by "tactically" implementing the system thinking learning activities into an undergraduate engineering curriculum. For example, a system thinking module could be added into the lecture portions of all engineering courses. This would likely serve as a catalyst to mastering the other course objectives. Similarly, the labs should be updated to reinforce the classical methods defined in Table 1. Many schools require a senior capstone project to solidify what the undergraduates learned over their previous years. Topics should be chosen to ensure that case studies, ESE and multi-discipline interactions are incorporated to add a system thinking element to each capstone project.

It is worth examining if undergraduate systems engineering and industrial system engineering degree programs include architecture related courses earlier than other degree programs. A survey of curriculums [24] shows that they do not. This is likely because the art of architectural synthesis is often too abstract to teach and too difficult to grade at the undergraduate level. Most undergraduate engineering assignments start with well-defined CONOPs and requirements which obfuscates the importance of the prerequisite system architecture activities. This is why it is critical that the undergraduate curriculum be inclusive of systems thinking so that the foundation for system architecture education is solidly in place prior to engineers graduating. It should also be remembered that system thinking skills apply to all disciplines/domains and would be equally as beneficial to engineers in other discipline programs.

When looking at the entirety of system thinking education, a holistic system thinking approach must be taken. Kordova and Frank [26] found that there was a correlation between personality traits of practitioners and their systems thinking capacity. This diversity in practitioner's innate abilities must be accounted for in undergraduate curriculum to ensure a broad cadre of system thinkers can be produced. Valerdi and Rouse [25], found that there is a difference between those who can perform system thinking, and those who can both do and apply systems thinking. They postulated that an enabling environment needed to be established in an organization so that practitioners are not discouraged from acting on their systems thinking analysis, and that desired system properties were both realized and implemented. Therefore, in addition to just learning the system thinking competencies in Table 2, the learning of "soft skills" (e.g. leadership, motivation, communication, management, etc. [16]), consideration of practitioner's innate capabilities, and establishing an organization wide enabling environment for system architecture are required to get the desired growth in effective systems thinkers.

Incorporating Architecture and System Thinking into Organizations

The systems thinking competencies, if seeded into undergraduate education as described in the previous section, will continue to grow for each practitioner in an organization though experience on assigned projects, mentoring, professional development, and in-house training courses. The challenge in an organization is to introduce the remaining architectural tools and methods from Table 1, in addition to the previously mentioned system thinking techniques.

These critical items include learning the art and science of synthetically deriving and documenting a complex system architecture which meets a defined user concept, and incorporates end user value statements (e.g. resilience, growth, rapid capability insertion, cyber resilience, rapid distribution, low cost, etc.). The system architecture training will typically be incorporated along with the appropriate domain (e.g. aerospace, communications, radar, telecommunications, humanitarian engineering, etc.) and functional training (typically systems engineering for architecture). To support these learning objectives, various modeling, simulation and architecture/engineering tools will be provided to engineers to complement the organizational training, along with any unique set of engineering processes or standards that all systems must adhere to (e.g. IEEE 15288.2 [31], AS9145 [32], etc.).

The methods of teaching system architecture related tools and methods will vary in an organization as the teacher (master)/student (novice) paradigm breaks down. Training may be taken by new hires or senior staff with significant experience. In [33], it was stated that organizational training for the functional competencies (e.g. systems engineering, system architecture/system thinking) are typically comprised of organization approved functional training courses which are assigned to each individual based on their assigned functional roles. Training is typically completed by participating in live classroom sessions. This method of instruction is preferred due to the relatively stable nature of the functional training and the desire for classroom interaction between participants to maximize learning from the participants' collective experience.

System architecture organizational training typically includes Concept of Operations Development, System Architecture, Architectural Modeling Languages/Frameworks, Modular Open System Approach (MOSA), Systems Engineering Process, and Modeling and Simulation. In addition to providing the required system architecture elements from Table 1, these courses carry on the system thinking competencies from Table 2 using a variety of class projects and case studies [e.g. 34]. Due to the socio-technical and socio-economic nature of complex systems, additional courses in reliability, maintainability, supportability, security, networks, logistics, economics, politics, foreign trade, etc. are also required depending on the nature of the complex systems being developed within the organization [20].

Training deficiencies are ideally established through periodic competency assessments of engineers. The original INCOSE Systems Engineering Body of Knowledge (SEBoK) functional competency model serves as the source of most documented competency models [35]. For organizations, [33] suggests an expanded competency model where the "Discipline Oriented Skills" (i.e. domain) competency of the original model be removed from the functional assessment and be addressed in a separate domain technical competency assessment. This approach allows for a more detailed assessment of an individual's technical/domain skill set and allows for the domain competencies to be tailored to match each specific organizational domain area's identified needs to be directly assessed. Once domain training deficiencies are identified, the optimal training approach can be determined as shown in Fig. 1.

As can be seen, the first step is to determine how dynamic the specific domain technology is. Technologies such as radio-wave propagation, for example, are highly mature with considerable training literature available. This literature may be leveraged directly to create either classroom or online offerings and will change very little with time. The stability of such technologies lends itself to online training since the cost to develop the training is amortized over several years, as the training remains relevant [33].

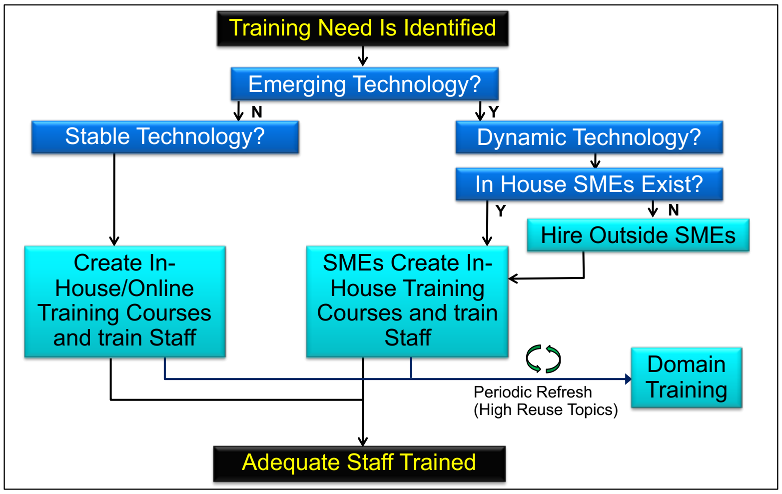


Figure 1. Domain Training Format Decision Tree [33]

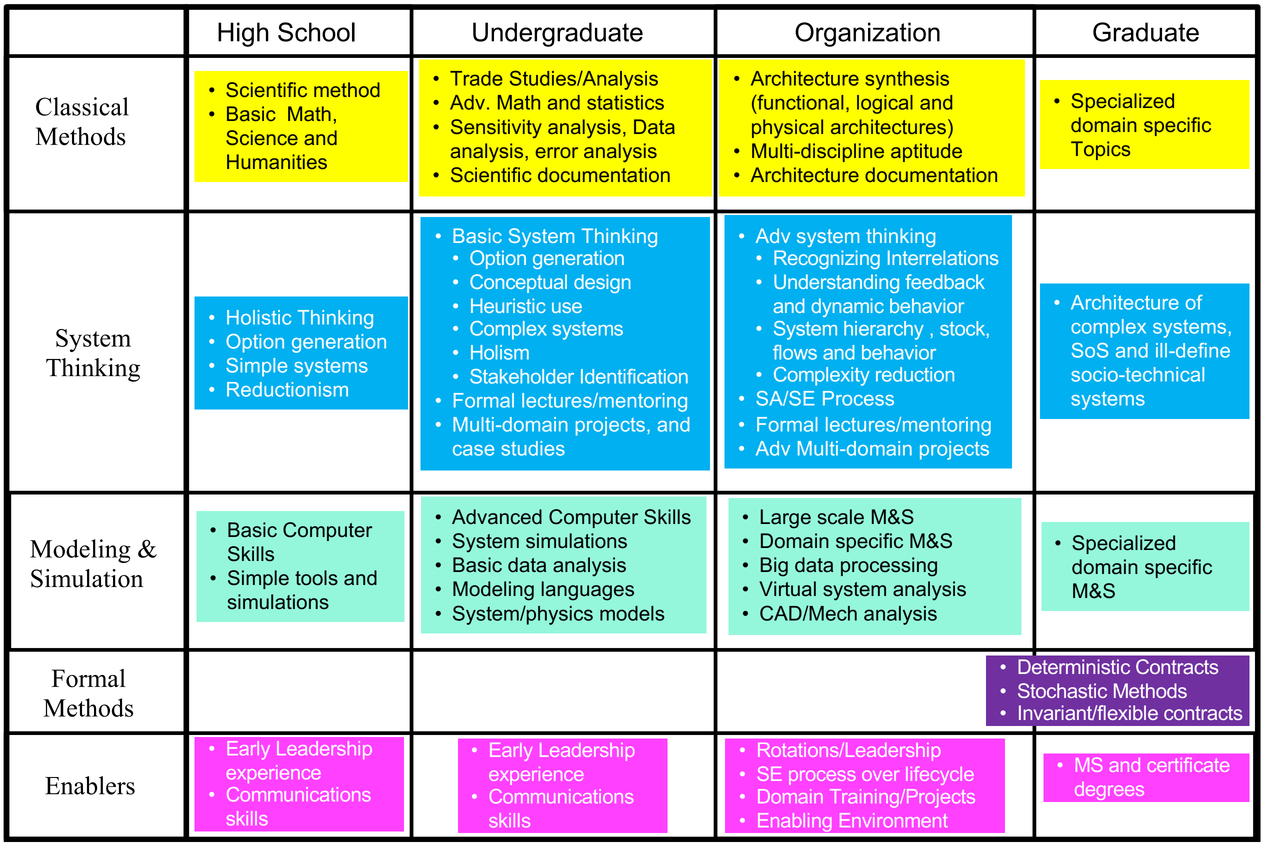
Dynamic or rapidly developing domain technologies that are not adequately covered in open literature must also be addressed. If an organization has Subject Matter Experts (SMEs) in this domain area, then they can have them develop in-house training courses and train staff through a series of classroom sessions. In cases where internal SMEs do not exist, external SMEs should be brought in to develop the training materials and teach the courses [33].

Besides direct domain and functional training, organizations have several other capabilities to enhance system architecture practices and system thinking competencies. Rotations through various projects and domain areas will speed the incorporation of system thinking, and greatly aid the architecting of complex systems [27]. Organizations can also establish Communities of Practice, provide periodic domain training and require professional development to grow practitioners in the areas of system architecture and system thinking. With the increase in the number graduate degrees and certificate programs in system architecture [24], organizations may also choose to fund some of their staff to get advance degrees or certificates in system architecture. These graduates will then return to the organization and act as resources for the existing system architecture/system thinking organizational training.

Organizations also have ample opportunities to put individuals in a leadership positions to hone both their leadership, communication and management skills. This provides the requisite “soft skills” needed to complement system architecture/system thinking effectiveness. All organizations will need to establish an enabling environment for system architecture and system thinking by both supporting training and enforcing a defined process for development of a proper architecture, even under tight constraints of staffing, time and/or funding. The establishment of an organization’s enabling environment [25] is critical to a successful architecture and is the first principal of the Modular Open System Approach (MOSA) [36] for that very reason. Quite simply, if an organization does not establish an enabling environment for both performing and implementing architecture and system thinking, the desired complex systems will not be realized.

Like universities, organizations should also be on alert for practitioners that do not have the prerequisite skills on conducting background research, performing trades/analysis, making data driven decisions, performing error analysis, etc. These prerequisites must be in place to perform proper system architecting and system thinking. Another common shortcoming that must be reconciled is the lack of documentation of systems and complex systems. In [37], it was found that on most programs "only 30% of a simple system's design knowledge is documented, with the remaining 70% as tacit knowledge encapsulated in the experiences of the designers." This is a clear and direct sign that key architecture activities are not being performed properly, and the link between the design and higher level concept of operations will be broken. Organizations should use the holding of both an Architecture Review Board (ARB), to review the top level architectural artifacts, and a System Functional Review (SFR), to review the lower level functional, logical and physical architectural artifacts prior to commencing system design as key metrics for tracking an organization's system architecture maturity [10], [31], [38].

A nominal timeline of the proposed system architecture/system thinking educational approach is shown in Figure 2. As can be seen, the system architecture categories and associated enablers are developed over early education, undergraduate education, within an organization and through advanced degrees or certificate programs. It is imperative that the early fundamentals (e.g. math, scientific method, humanities, etc.) are performed since they are all enablers to both the practice of system architecture, and the application of system thinking.



	High School	Undergraduate	Organization	Graduate
Classical Methods	• Scientific method • Basic Math, Science and Humanities	• Trade Studies/Analysis • Adv. Math and statistics • Sensitivity analysis, Data analysis, error analysis • Scientific documentation	• Architecture synthesis (functional, logical and physical architectures) • Multi-discipline aptitude • Architecture documentation	• Specialized domain specific Topics

System Thinking	• Holistic Thinking
	• Option generation
	• Simple systems
	• Reductionism
		• Basic System Thinking
		  • Option generation
		  • Conceptual design
		  • Heuristic use
		  • Complex systems
		  • Holism
		  • Stakeholder Identification
		• Formal lectures/mentoring
		• Multi-domain projects, and case studies
			• Adv system thinking
			  • Recognizing Interrelations
			  • Understanding feedback and dynamic behavior
			  • System hierarchy , stock, flows and behavior
			  • Complexity reduction
			• SA/SE Process
			• Formal lectures/mentoring
			• Adv Multi-domain projects
				• Architecture of complex systems, SoS and ill-define socio-technical systems

Modeling & Simulation	• Basic Computer Skills • Simple tools and simulations	• Advanced Computer Skills • System simulations • Basic data analysis • Modeling languages • System/physics models	• Large scale M&S • Domain specific M&S • Big data processing • Virtual system analysis • CAD/Mech analysis	• Specialized domain specific M&S
Formal Methods				• Deterministic Contracts • Stochastic Methods • Invariant/flexible contracts
Enablers	• Early Leadership experience • Communications skills	• Early Leadership experience • Communications skills	• Rotations/Leadership • SE process over lifecycle • Domain Training/Projects • Enabling Environment	• MS and certificate degrees

Figure 2. System Architecture Education Timeline

Domain Specific Architecture and System Thinking Learning

Up to this point, the paper has detailed a proposed approach to teaching system architecture/ system thinking that are directly applicable to all domains. However, it is worth a brief example on the differences of architecting across various domains. In this example, three aspects of the architecture development process: stakeholder identification, system environment and standards, will be compared between a defense and a humanitarian system (e.g. nation state disaster response [39]). All system architectures must take into account stakeholders of the system including the government, customers, vendors, developers, end users, maintainers, logisticians, and security staff, as well as constraints of cost, schedule, staffing, system environment, process, regulations, and standards [34]. While the architectural approach and system thinking techniques are the same for both domains, each will have unique elements which will need to be stressed in a practitioner's education and training.

For a defense system the government stakeholders are typically involved prior to system development when negotiating for project funding to starting an acquisition cycle for a new defense system, and then later coordinating with the vendor(s) over the lifecycle of the system. This process is documented in the DoD 5000 acquisition policy [40] which defense system practitioners must be keenly aware of. The stakeholder role of system user is central to defense systems, as they typically both run and status the deployed defense systems. To better integrate the human operators with the system, the system developers must be knowledgeable of advances in Human System Integration (HSI) (e.g. [41]), and automation/decision support systems (e.g. [42]).

Similarly, defense systems are designed to work over a wide range of harsh environments characterized by extreme temperatures, highly dynamic vibration/shock, explosive atmosphere, electromagnetic interference, etc., which requires detailed training in system environmental and electromagnetic design and qualification (e.g. [43], [44]). This training can be significant for systems in exceptionally harsh environments such as space, where engineers must in addition be educated on material properties, radiation effects, low power design, orbits, etc., as well as the complicated certification process for space based system (e.g. [45]).

Standards are significant for defense systems as the Department of Defense looks for both longevity of systems and the ability to perform rapid periodic capability insertion and obsolescence mitigation [36]. Therefore, defense systems typically have standards in the areas of security (e.g. [46]), open architecture (e.g. [47]), and open hardware/software (e.g. [48]). Production and manufacturing of defense systems is just as important as development, and also is bound by several standards (e.g. [32]) . Expertise in these areas will require additional education and "cross pollination" with organizational security, operations, and hardware/ software staff, in addition to monitoring or participating in relevant open architecture standard consortiums.

For humanitarian systems, there is typically only commodity materials used (e.g. food , water, shelter, etc.), so that only local codes directly apply, as opposed to the rigorous development and production standards which defense systems must meet. Similarly, while the humanitarian system's environment may vary significantly due to climate, weather and season, it is almost always a land based application and does not require the level of hardening of defense systems.

Where the humanitarian system differs the most is in the area of stakeholders. The political/ government and aid provider/recipient stakeholder interactions are much more tightly coupled over the lifecycle of the system than for a defense system. The aiding organization must have a deep understand of the U.S. government policies and regulations when providing aid to foreign nations, as well as the policies and regulations of the local government receiving the aid. The aiding organization must also understand the cultures and customs of the aid recipient to most effectively provide aid to them [49], [50]. In most cases, the aid organization will work with a local Non-Government Organization (NGO) and organizations like the United Nations and Red Cross who will coordinate the aid delivery to those in need and can help bridge any knowledge gaps in local customs or culture [51]. Any required administrative or logistics coordination with these groups must be known in advance. These aspects of humanitarian systems requires strong education in the area of system thinking, problem solving, politics, international regulations, humanities, logistics and social sciences.

Exemplars on how to introduce curricula to address the complex socio-technical competencies which make up humanitarian systems are detailed in [30], [52], and [53]. In [30], the US Military Academy established a 3 course Core Engineering Sequence (CES) with 10 of 40 lessons dedicated to systems thinking applied to ill-defined problems such as providing nutritious food to remote island inhabitants and determining a plan for handling the plight of Syrian refuges. Grades were based on the proper use of the provided tool, and ability to analyze the situation and synthesize a solution through system thinking. In [52], Baylor University created a Humanitarian Engineering concentration consisting of: 1) people/cultures, 2) economics, 3) development, 4) energy, 5) food/water, and 6) international service to ensure engineering efforts had a positive long term effect on those receiving aid. This is similar to the Humanitarian Engineering and Social Entrepreneurship (HESE) Program at Penn State University [53] which stresses systems thinking, communication, cultures, ethics, interdisciplinary courses and cocurricular service. The humanitarian courses included in these programs were reflected in a survey of 187 members of the Engineers Without Boarders (EWB) organization. The EWB practitioners identified that along with technical skills and system thinking, that communication, cultural awareness, global awareness, leadership, teamwork, problem solving, project management and innovation/resourcefulness where key competencies that needed to be learned by those performing humanitarian engineering [54].

From these examples you can see that the core learning required to produce system architecture and systems thinking practitioners will also need to ensure completeness of coverage of the “multi-domain projects” line items in Figure 2 based on the system domain, associated stakeholders, environment and standards to be used.

Conclusion

This paper proposed an educational approach to produce practitioners with the appropriate competency level of system architecture to develop complex systems in any domain. The approach centered on developing basic holistic thinking and soft skills in early education as prerequisites to undergraduate curricula where system thinking, arguably the greatest system architecture tool, could be introduced into all science and engineering courses. System thinking elements would be paired with exiting classes and incorporated though lectures, mentoring, hands-on projects, case studies, Experimental Systems Engineering (ESE), and exposure to the various multi-discipline system stakeholders.

Organizations would continue the system thinking education of practitioners, and also provide training for the other architectural tools and methods. This organizational training will take many forms including, mentoring, professional development, in-house training courses, leadership assignments, experience on projects, rotations through various domains and possibly graduate degrees or certificates. Practitioners will be also be introduced to classical analysis, functional decomposition of a system, modeling, simulation, and the architecture/engineering tools to complement the organizational system architecture training, along with any unique set of engineering processes or standards that all systems must adhere to. Lastly, the organization must ensure completeness of domain specific training. As was shown for the example of a defense and humanitarian system, while the architectural approach and system thinking techniques were the same for both domains, each had unique elements which would need to be stressed in a practitioner's education and training.

By implementing the proposed approach, students and practitioners alike will acquire and refine their systems thinking skills which will serve them in all aspects of their professional careers. In addition, by "left shifting" elements of system architecture such as holistic thinking and option generation, along with soft skills such as leadership, communication and management, the gap between what organizations want and what universities produce will be drastically reduced. Finally, with a cadre of skilled holistic thinking students being supplied to the workforce, organizations can more readily complete each practitioner's system architecture training to produce teams of individuals who can successfully address the architecture and design of systems of ever increasing complexity.

References

[1] H. Simon, "The architecture of complexity," in *Proceedings of the American Philosophy Society,* vol. 106, no. 6, pp. 467-482, 1962.

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

[3] 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.*

[4] E. Crawley, B. Cameron and S. Selva, *System Architecture - Strategy and Product Development for Complex Systems.* Hoboken, NJ: Pearson Higher Education Inc, 2016.

[5] E. Rechtin, *Systems Architecting: Creating & Building Complex Systems*. Upper Saddle River, NJ: Prentice Hall PTR, 1991.

[6] A. M. Madni, "Generating novel options during system architecting: Psychological principles, system thinking, and computer-based aiding," *Systems Engineering,* vol. 17, no. 1, pp. 1-9, 2014.

[7] A. M. Madni, "Transdisciplinarity: Reaching beyond disciplines to find connections," *Transactions of the Society for Design and Process Science (SDPS),* vol. 11, no. 1, pp. 1-11, March 2007.

[8] A. M. Madni and A. M. Ross, "Exploring concept trade-offs," in *Trade-off Analytics*, G. S. Parnell, Ed., Hoboken, NJ: John Wiley and Sons Inc., 2016.

[9] E. Ordoukhanian and A. M. Madni, "System Trade-offs in Multi-UAV Networks," in *Proceedings of the AIAA SPACE 2015 Conference and Exposition, Pasadena, CA, USA, 2015*.

[10] F. Kockler, T. Withers, J. Poodiack and M. Gierman, *Systems Engineering Management Guide AD-A223-168*. Ft. Belvoir, VA: Defense Systems Management College Publication, Jan. 1990.

[11] C. Coulston and R. M. Ford, "Teaching functional decomposition for the design of electrical and computer systems," in *Proceedings of the 34th Annual Frontiers in Education Conference, FIE 2004, Savannah, GA, USA, pp. F4G6-11, 2004*.

[12] R. Taylor, N. Medvidovic and E. Dashofy, *Software Architecture - Foundations, Theory and Practice*, Hoboken, NJ: John Wiley and Sons Inc., 2010.

[13] R. Aarenstrup, *Managing Model-Based Design*, Natick, MA: MathWorks Inc. publication, 2015.

[14] M. Sievers and A. M. Madni, “A flexible contracts approach to systems resiliency,” in *Proceedings of the 2014 IEEE International Conference on Systems, Man and Cybernetics, San Diego, CA, USA, Oct 5-8, 2014*.

[15] M. Sievers and A. M. Madni, “Agent-based flexible design contracts for resilient spacecraft swarms,” in *Proceedings of the 54th AIAA Aerospace Sciences Meeting, San Diego, CA, USA, 2016*.

[16] A. Shenhar, "Systems engineering management: a framework for the development of a multidisciplinary discipline," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 24, no. 2, pp. 327-332, Feb. 1994.

[17] J. Walther and D. F. Radcliffe, “The competence dilemma in engineering education: Moving beyond simple graduate attribute mapping,” *Australasian Journal of Engineering Education*, 13:1, pp. 41-51, 2007.

[18] A. Azemi, "Promoting innovation through systems thinking and systems design," in Proceedings of the 2018 Annual IEEE International Systems Conference (SysCon), Vancouver, BC, Canada, pp. 1-3, 2018.

[19] C. B. Keating and A. V. Gheorghe, "Systems thinking: Foundations for enhancing system of systems engineering," in *Proceedings of the 11th System of Systems Engineering Conference (SoSE), Kongsberg, Norway, pp. 1-6, 2016*.

[20] H. Sibo-Ingrid, C. A. Gomez and S. Liou, "Systems thinking: A paradigm for advancing technology to enhance humanity," in *Proceedings of the 2018 International Conference on Orange Technologies (ICOT), Nusa Dua, Bali, Indonesia, pp. 1-6, 2018*.

[21] E. B. Dano, "Resilient system engineering in a multi-UAV system of systems (SoS)," in *Proceedings of the 2019 International Symposium on Systems Engineering (ISSE), Edinburgh, United Kingdom, 2019.*

[22] R. Arnold and J Wade, “A definition of systems thinking: A systems approach,” in *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, 2015, vol. 44, pp. 669-678.

[23] INCOSE, *A World in Motion, Systems Engineering Vision 2025*, San Diego, CA: INCOSE Publication, 2014.

[24] INCOSE and SERC, *Worldwide Directory of Systems Engineering and Industrial Engineering Academic Programs*, San Diego, CA: INCOSE Publication, 2016.

[25] R. Valerdi and W. B. Rouse, "When systems thinking is not a natural act," in *Proceedings of the 2010 IEEE International Systems Conference, San Diego, CA, USA, pp. 184-189, 2010*.

[26] S. K. Kordova and M. Frank, "Systems Thinking - innate or learned? Recent study findings," in *Proceedings of the 2015 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM), Singapore, pp. 1490-1493, 2015*.

[27] D. R. 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.*

[28] S. Koral-Kordova and M. Frank, "Improving capacity for engineering systems thinking (CEST) among industrial engineering students," in *Proceedings of the 2012 IEEE International Conference on Industrial Engineering and Engineering Management, Hong Kong, pp. 1378-1380, 2012*.

[29] A. Golkar, "Experiential systems engineering education concept using stratospheric balloon missions," *IEEE Systems Journal*, vol. 14, no. 2, pp. 1558-1567, June 2020.

[30] T. R. Ryan, J. H. Schreiner and S. E. Gillespie, "Teaching systems thinking: A case study at the United States Military Academy," in *Proceedings of the Annual IEEE International Systems Conference (SysCon), Vancouver, BC, Canada, pp. 1-7, 2018*.

[31] *IEEE Standard for Technical Reviews and Audits on Defense Programs*, IEEE Standard 15288.2^{TM}, 2014.

[32] *AS Standard for Requirements for Advanced Product Quality Planning and Production Part Approval Process,* Aerospace Series Standard AS9145^{TM}, 2016.

[33] E. B. Dano, "A validated systems engineering competency methodology and functional/ domain competency assessment tool," in *Proceedings of the 2019 International Symposium on Systems Engineering (ISSE), Edinburgh, United Kingdom, 2019.*

[34] E. B. Dano, "A Case Study on the Architectural Development of a Transceiver Assembly," in *Proceedings of the 2019 International Symposium on Systems Engineering (ISSE), Edinburgh, United Kingdom, 2019.*

[35] SEBoK Editorial Board. 2020, *The Guide to the Systems Engineering Body of Knowledge (SEBoK), v. 2.3*, R.J. Cloutier (Editor in Chief). Hoboken, NJ: The Trustees of the Stevens Institute of Technology. Accessed Jan 2021. www.sebokwiki.org. BKCASE is managed and maintained by the Stevens Institute of Technology Systems Engineering Research Center, the International Council on Systems Engineering, and the Institute of Electrical and Electronics Engineers Systems Council.

[36] Open Systems Joint Task Force, *A Modular Open Systems Approach (MOSA) to Acquisition – A Program Manager Guide, v2.0*, Washington, DC: OSJTF Publication, September, 2004.

[37] C. T. Lamb and D. H. Rhodes, "Systems thinking as an emergent team property: Ongoing research into the enablers and barriers to team-level systems thinking," in *Proceedings of the* 2nd Annual IEEE Systems Conference, Montreal, Que., Canada, pp. 1-7, 2008.
[38] Defense Acquisition University, *Defense Acquisition Guidebook*, ed. K. Stewart, https://www.dau.mil/tools/t/Defense-Acquisition-Guidebook, Feb., 2017.
[39] K. Ward, A. Newman, J. Gleason, H. Lee, T. Saltysiak and J. Farr, "Complex system architecture analysis of nation state disaster response," Center for Nation Reconstruction and Capacity Publication, West Point, NY, Report 2014-1, July 2014.
[40] Office of the Under Secretary of Defense for Acquisition and Sustainment, *DoD 5000 Series Acquisition Policy Transformation Handbook,* Washington, DC: OUSD(A&S) Publication, January, 2020.
[41] Madni, A. M., "Integrating humans with and within complex systems: Challenges and opportunities," *CrossTalk, The Journal of Defense Software Engineering*, pp. 4-8, May/June 2011.
[42] E. Turban, J. Aronson and T. Liang, *Decision Support Systems and Intelligent Systems*, 7th Edition. Boston, MA: Pearson Publishing, 2004.
[43] *Department of Defense Test Method Standard Environmental Engineering Considerations and Laboratory Tests*, MIL-STD-810G, October, 2008.
[44] *Department of Defense Interface Standard Requirements for the Control of Electromagnetic Interference Characteristics of Subsystems and Equipment*, MIL-STD-461G, December, 2015.
[45] P. Fortescue, G. Swinerd and J. Stark, *Spacecraft Systems Engineering*. Chichester, U.K.: John Wiley and Sons Ltd., 2011.
[46] *Department of Commerce Minimum Security Requirements for Federal Information and Information Systems*, FIPS PUB 200, March, 2006.
[47] Open Systems Architecture Data Rights Team, *DoD Open Systems Architecture – Contract Guidebook for Program Managers, v1.1*, Washington, DC: DoD OSADRT Publication, June, 2013.
[48] *ANSI/VITA OpenVPX™ System Standard*, ANSI/VITA 65.0, 2019.
[49] B. Amadei and W. A. Wallace, "Engineering for humanitarian development," *IEEE Technology and Society Magazine*, vol. 28, no. 4, pp. 6-15, Winter 2009.
[50] P. Winkelman, "Maslow meets the stonecutter," in *Proceedings of the 20th International Conference on Engineering Design (ICED15), Politecnico Di Milano, Italy, pp. 1-15, July 2015.*
[51] A. Monaghan and M. Lycett, "Big data and humanitarian supply networks: Can big data give a voice to the voiceless," in *Proceedings of the 2013 IEEE Global Humanitarian Technology Conference, Silicon Valley, CA, pp. 432-43, Oct. 2013.*
[52] W. Jordan, "Creating a sustainable humanitarian engineering program," in *Proceedings of the* 2015 IEEE Canada International Humanitarian Technology Conference (IHTC2015), Ottawa, ON, Canada, pp. 1-4, 2015.
[53] K. Mehta and I. Gorski, "Preparing engineers for careers in social innovation and sustainable development," in *Proceedings of the 2016 IEEE Frontiers in Education Conference (FIE), Erie, PA, USA, pp. 1-5, 2016.*
[54] M. H. Ba-Aoum, "Comprehensive model for fostering humanitarian engineering education," in *Proceedings of the 2016 IEEE Global Humanitarian Technology Conference (GHTC), Seattle, WA, pp. 304-311*, *2016.*